A novel feed rate scheduling method with acc-jerk-Continuity and round-off error elimination for NURBS interpolation

Yifei Hu\(^1,2\), Xin Jiang\(^{1,4,5,*}\), Guanying Huo\(^{1,4,6}\), Cheng Su\(^{1,4,6}\), Shiwei Zhou\(^3\), Bolun Wang\(^7\), Hexiong Li\(^{1,2}\), Zhiming Zheng\(^{1,4,5,6}\)

\(^1\) LMIB & NLSDE & Institute of Artificial Intelligence, Beihang University, Beijing, China

\(^2\) School of Mathematics Science, Shenyuan Honors College, Beihang University, Beijing, China

\(^3\) GSK CNC equipment Co., Ltd., Guangzhou, Guangdong China

\(^4\) Zhongguancun Laboratory, Beijing, China

\(^5\) Peng Cheng Laboratory, Guangdong, Shenzhen, China

\(^6\) Zhengzhou Aerotropolis Institute of Artificial Intelligence, Henan, Zhengzhou, China

\(^7\) Visual Computing Center, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia

Corresponding author: Xin Jiang (e-mail: jiangxin@buaa.edu.cn)

Abstract

Feed rate scheduling is a critical step in computer numerical control (CNC) machining, as it has a close relationship with machining time and surface quality. It has now become a hot issue in both industry and academia. In this article, we present a novel and complete S-shape-based feed rate scheduling method for three-axis NURBS tool paths, which can reduce high chord errors and round-off errors, and generate continuous velocity, acceleration, and jerk profile. The proposed feed rate scheduling method consists of three modules: a bidirectional scanning module, a velocity scheduling module, and a round-off error elimination module. The bidirectional scanning module aims to guarantee the continuity of the feed rate at the junctions between successive NURBS blocks, where the chord error, tangential acceleration, and tangential jerk limitations are considered. After the NURBS blocks have been classified into two cases by the previous module, the velocity scheduling module first calculates the actual maximum feed rate. It then generates the feed rate profiles of all NURBS blocks according to the proposed velocity profile. Later, the round-off error elimination module is applied to adjust the actual maximum feed rate so that the total interpolation time becomes an integer multiple of the interpolation period, which leads to the elimination of round-off errors. Finally, benchmarks are conducted to verify the applicability of the proposed method. Compared with the traditional method, the proposed method can save the interpolation time by 4.67% to 14.26%.
Graphical Abstract

A novel feed rate scheduling method with acc-jerk-Continuity and round-off error elimination for NURBS interpolation

Key words: NURBS interpolation; Jerk continuous; Bidirectional scanning; Round-off error elimination
Highlights

* A novel feed rate scheduling method with acc-jerk-Continuity and round-off error elimination for NURBS interpolation

- A complete S-shape feed rate scheduling approach for NURBS interpolator is given.
- The acceleration and jerk are continuous and strictly constrained.
- The round-off error is eliminated.
- The machining efficiency is improved.

1. Introduction

In traditional multi-axis computer numerical control (CNC) machining, piecewise linear (G01-based) tool paths are widely used. (Shanshan et al., 2015). Especially when machining complex surface parts such as blades, dies and automobile components, numerous segments will be needed to guarantee accuracy. When the G01 tool path is directly used to guide the movement of machine axes (Zhong et al., 2019), the inevitable frequent acceleration and deceleration processes (Hui et al., 2018) will reduce the machining efficiency and affect the machining quality (B. Xu et al., 2021).

As a result, parametric interpolations are increasingly developed in CNC (Sang et al., 2020). Due to the superior properties such as global smoothness and local support (Piegl & Tiller, 1997), NURBS curves are widely used in parametric interpolation.

The first problem in NURBS interpolation is the nonlinear relationship between the arc length displacement and the parameter value (M. Liu et al., 2014). To couple with this problem, some works such as Taylor’s expansion method (Yang & Kong, 1994), predictor-corrector interpolator (PCI) algorithm (Mi-Ching et al., 2003), quintic spline interpolation method (Erkorkmaz & Altintas, 2005), and inverse length functions (ILF)
method (Lei et al., 2007) were presented. When the interpolation period is constant, the interpolation step lengths, which are utilized to compute the interpolation points, are determined by the feed rate, so the feed rate scheduling has been one of the most critical processes in NURBS interpolation (H. Li et al., 2022). Though a higher feed rate value can reduce machining time, when the limitation is exceeded, significant geometric deviations such as contour error (Schmitz et al., 2008) and chord error (Yeh & Hsu, 2002) are unavoidable, which seriously worsen the machining quality. Therefore, feed rate scheduling is one of the most essential works in CNC (Hong-Seok et al., 2017), how to achieve the trade-off between machining efficiency and machining quality is a critical problem to be solved in feed rate scheduling.

In recent years, extensive feed rate scheduling methods have been proposed by the experts and scholars, which can be roughly classified as two approaches: time-optimal approaches (Erkorkmaz & Heng, 2008; Dong et al., 2005; Liu et al., 2017; Lu et al., 2020; Qiang et al., 2012; Sencer et al., 2008; Sun et al., 2014; SUN et al., 2019; Timar & Farouki, 2007; Timar et al., 2005; Yuan et al., 2013; K. Zhang et al., 2012) and acceleration/deceleration (ACC/DEC) approaches (Du et al., 2015; Erkorkmaz & Altintas, 2001; Fan et al., 2012; Huang & Zhu, 2017; Jahanpour & Alizadeh, 2014; Jeon & Ha, 2000; Jia et al., 2016; Lai et al., 2008; Lee et al., 2011; D. Li et al., 2016; Y. Li et al., 2009; M. Liu et al., 2014; X. Liu et al., 2005; Luo et al., 2007; Nam & Yang, 2004; Ni et al., 2018; Sang et al., 2020; Tajima & Sencer, 2016; Wang et al., 2015; Xinhua et al., 2016; B. Xu et al., 2021; R. Z. Xu et al., 2008; Ye et al., 2008; Yeh & Hsu, 2002; L. Zhang & Du, 2018).

The time-optimal approaches usually construct an optimization problem with the shortest machining time as the optimization goal, while the geometric deviation and kinematic characteristics are constraints. Then the shape of the feed rate profile can be obtained by solving differential equations, where the solution method can be an analytic or a numeric method. Timar et al. (2005) first showed that the square of the time-optimal feed rate can be determined as a piecewise rational function of the curve parameter. The velocity curve met the “bang-bang” control principle was obtained. In (Timar & Farouki, 2007), two differential equations, which have closed-form solutions, were derived and solved. The time-optimal feed rate function under constant or speed-dependent acceleration limits was obtained. Since only the acceleration was involved in these two works, the time-optimal feed rate profiles can be analytically
obtained. However, it is difficult to integrate multiple constraints and high-order constraints. More numeric methods were developed to solve this problem. Dong et al. (2005) generalized the two-pass feed rate optimization algorithm, and various state-dependent constraints, such as jerk constraint, were incorporated. This generalized algorithm could generate a globally optimal solution under various constraints. Sencer et al. (2008) expressed the variation of the feed rate on the toolpath by a cubic B-spline curve. The velocity, acceleration, and jerk limits of the five axes limitations were all considered, and then the time-optimal feed rate profile was obtained by iteratively modulating the control points. Erkorkmaz & Heng (2008) proposed a heuristic search method to generate the feed rate profile, where the axis velocity, acceleration, torque, and jerk were constrained. In (K. Zhang et al., 2012), the same purpose was achieved under a greedy rule. In (Lu et al., 2020), Pontryagin’s maximum principle was applied. Qiang et al. (2012) applied a control vector parameterization (CVP) method to convert the optimal control problem into nonlinear programming (NLP), which was solved by the Sequential Quadratic Programming (SQP) method. Yuan et al. (2013) proposed a chord-error-velocity-limit curve. Then found a velocity curve governed by the acceleration or jerk bounds ‘under’ the chord-error-velocity-limit curve to plan the velocity. Liu et al. (2017) derived a linear objective function for feed rate optimization using a discrete format of the primitive continuous objective function. Then, the preset multi-constraints were approximated as linear constraint conditions on the decision variables in the linear objective function. The linear model for feed rate optimization could be solved by well-developed linear programming algorithms. The optimal solution was fitted to the smooth spline curve as the ultimate feed rate profile. Sun et al. (2014) first constructed the initial feed rate profile with confined chord error, angular velocity, and axis velocities. Then, the proportional adjustment of feed rate-sensitive regions was applied to simultaneously reduce the magnitudes of constraints such as angular acceleration, linear acceleration, axis accelerations, and jerks. Since the iterative adjustment was involved, this scheme cannot be implemented in real-time. Later, Sun et al. (2019) developed a piecewise linear programming scheme with $C^2$ continuity assurance for adjacent B-spline feed profiles to solve the feed rate scheduling optimization model on time. However, the tool path curves were parameterized with the normalized arc-length parameters in the above method, which was a relatively strong condition.

Though time-optimal approaches can generate minimum time or near minimum time feed rate profile, the computational cost of solving an optimization problem and fitting a feed rate profile is exceptionally high. Moreover, some problems related to the machining quality, such as
round-off errors (Ni et al., 2019), are still unresolved. Therefore, researchers have developed a significant number of ACC/DEC approaches.

The ACC/DEC approaches usually design the acceleration and deceleration process in advance, where the geometric deviation and kinematic characteristics are considered. Due to the complex geometric shape of the parts, the tool paths usually contain multiple accelerations and decelerations processes. Thus, critical points, such as high-curvature points and \( C^0 \) continue points, are usually utilized to divide the whole toolpath into several blocks. The designed ACC/DEC model is applied to plan the acceleration and deceleration process. Some early works (Jeon & Ha, 2000; Yeh & Hsu, 2002) only considered the acceleration limitation. When the curvature of the toolpath changes abruptly, the jerk may exceed the limit capacity of the machine, which will cause the machine’s vibration and reduce the machining accuracy. Several adaptive interpolation methods (Du et al., 2015; Erkorkmaz & Altintas, 2001; Jahanpour & Alizadeh, 2014; Jia et al., 2016; Lai et al., 2008; D. Li et al., 2016; X. Liu et al., 2005; Nam & Yang, 2004; Tajima & Sencer, 2016; R. Z. Xu et al., 2008; Ye et al., 2008) considering the jerk limit have been proposed to cope with this problem. Erkorkmaz et al. (2001) first introduced an S-shape feed rate scheming to generate jerk-limited trajectory. In (Jahanpour & Alizadeh, 2014), an optimized S-shaped \( C^2 \) quintic feed rate planning scheme was proposed to generate an acc-jerk-limited feed rate profile. Du et al. (2015) classified the NURBS blocks into three types, then generated feed rate profiles with a confined jerk, acceleration, and command feed rate for each block. D. Li et al. (2016) proposed a novel S-curve ACC/DEC control model and investigated the discretization of the Acc/Dec process. Jia et al. (2016) introduced the concept of feed rate-sensitive regions. The feed rate was kept constant at the feed rate-sensitive regions and varied smoothly within parts of the transition regions to reduce the frequency of feed rate changes, which benefited the machining quality. Although the jerk of these methods will not exceed the limit, the machining accuracy was affected due to the drastic change of jerk. To further generate continuous and smooth acceleration transition profiles, two acceleration smoothing algorithms based on the jounce limit were proposed by (L. Zhang & Du, 2018) and (Fan et al., 2012). Since the calculation processes were complicated, it is unsuitable for parametric curve interpolation. In (Lee et al., 2011), a sine-curve velocity profile was proposed to generate the feed rate profile, which constructed a jerk-limited and continuous feed rate profile. Wang et al. (2015) presented a trigonometric velocity scheduling algorithm based on two-time look-ahead interpolation. In (Xinhua et al., 2016), a novel approach for non-uniform rational B-spline (NURBS) interpolation through the integration of an
acc-jerk-continuous-based control method and a look-ahead algorithm was proposed. Since most of the above methods apply the trigonometric profile, only several points can reach the maximum acceleration or jerk (Huang & Zhu, 2017), and the machining process becomes less efficient. Moreover, since the interpolation time can’t be guaranteed to be an integer multiple of the interpolation period, the round-off errors are inevitable, whose existence may affect the machining accuracy and motion smoothness. Therefore, how to compensate for or eliminate round-off errors should not be ignored in feed rate scheduling.

To compensate for the round-off error, Du et al. (2015) proposed a variable-jerk compensation and an arc-length-error-compensation strategy to make the feed rate more continuous. The duration of each phase was changed into the integer multiples of the interpolation period by appropriately adjusting the jerk value. The round-off error was split into several sections, and the arc length increment during every interpolation was modified according to the corresponding section. In (Luo et al., 2007) and (Y. Li et al., 2009), similar time rounding and error compensation methods were proposed. However, the acceleration profiles were discontinuous. In (Ni et al., 2018), an optimized feed rate scheduling consisting of initial feed rate scheduling and parameters calculation of round-off error compensation based on an improved S-shaped ACC/DEC algorithm was introduced. They generated the initial feed rate profile according to the typical S-shaped ACC/DEC algorithm and then adjusted the maximum feed rate to guarantee the existence of the constant feed rate phase. The round-off error was split according to an improved S-shaped ACC/DEC algorithm based on trigonometric function. The feed length was updated by increasing the compensation length obtained by the improved algorithm. Although the interpolation time can become an integer time of the interpolation period, some kinematic characteristics might exceed the limit. Ni et al. (2019) proposed a time-rounding-up scheme to make the total interpolation time of each curve segment an integer multiple of the interpolation period, in which four feed rate scheduling methods were designed for different situations. Although the round-off error could be eliminated for most situations, when these methods were not applicable, the method by (Ni et al., 2018) was adopted, which resulted in the same problem (Ni et al., 2018). However, since the motion time was compressed in these methods, there is no guarantee that the motion parameters will not exceed the constraint range.

Considering chord error, round-off error, and kinematic characteristics, we propose a novel feed rate scheduling method with acc-jerk-continuity and round-off error elimination for three-axis NURBS tool paths. Firstly, in the preprocessing stage, the maximum allowable feed rate at sample points limited by the chord error, normal acceleration, normal jerk, and command feed rate is obtained. The NURBS curve is divided into several blocks by the critical points (the point whose maximum allowable
feed rate is the local minimum). Our method consists of three modules: a bidirectional scanning module, a velocity scheduling module, and a round-off error elimination module. For short NURBS blocks, the bidirectional scanning module calculates the actual maximum velocity and adjusts the start or end velocity accordingly if necessary. After that, the NURBS blocks may be in two situations: (1) the case of acceleration or deceleration (ACC-or-DEC case); (2) the case of acceleration-uniform-deceleration or acceleration-deceleration (ACC-and-DEC case). The velocity scheduling module then uses two strategies to generate the corresponding velocity profile for these two kinds of NURBS blocks. In the round-off error elimination module, the actual maximum velocity of the first NURBS block with ACC-and-DEC case is reduced appropriately so that the total interpolation time becomes an integral multiple of the interpolation period without changing the start and end velocity of this NURBS block.

The rest of this paper is as follows: Section 2 introduces the basic knowledge and the division of the NURBS curve. Section 3 introduces the improved S-shape velocity profile based on the trigonometric function and the three modules. In Section 4, simulations and experimental results are implemented to prove the feasibility and applicability of the proposed method. Finally, the conclusions and future works are given in Section 5.

2. The NURBS curve and its division

In this section, the definition and properties of the NURBS curve are reviewed first, then the process of generating the NURBS blocks is given.

2.1. The NURBS curve

A p-degree NURBS curve equation \( C(u) \) can be expressed as follows (Ni et al., 2019):

\[
C(u) = \frac{A(u)}{B(u)} = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i P_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \quad #(1)
\]

where \( \{P_i, i = 0, 1, ..., n\} \) are the control points, \( \{w_i, i = 0, 1, ..., n\} \) are the weights of the control points, \( n + 1 \) is the number of control points and \( p \) is the degree of the NURBS curve, \( \{N_{i,p}(u), i = 0, 1, ..., n\} \) are the \( p \)-degree B-spline basis functions defined on the non-uniform knot vector \( U = [u_0, u_1, ..., u_{n+p+1}] \). When the knot vector is known, \( \{N_{i,p}, i = 0, 1, ..., n\} \) are recursively defined as follows:

\[
N_{i,0}(u) = \begin{cases} 
1 & \text{if } u_i \leq u < u_{i+1} \\
0 & \text{else} 
\end{cases} \quad #(2)
\]

\[
N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad i = 0, 1, ..., n \quad #(3)
\]

where \( \frac{0}{0} = 0 \) is stipulated. Usually, the knot vector is unitized, i.e., \( u_0 = u_1 = \cdots = u_p = 0 \) and \( u_n = u_{n+1} = \cdots = u_{n+p+1} = 1 \).
To calculate the derivatives of a NURBS curve, the recursive formulas for calculating the k-order derivative of p-degree B-spline basic functions are given as follows:

\[ N_{i,p}^{(k)}(u) = \frac{p}{u_{i+p} - u_i} N_{i,p-1}^{(k-1)}(u) - \frac{p}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}^{(k-1)}(u) \]  

then the first and second derivatives can be calculated as:

\[ C^{(1)}(u) = \frac{A^{(1)}(u) - B^{(1)}(u)C(u)}{B(u)} \]  
\[ C^{(2)}(u) = \frac{A^{(2)}(u) - 2C^{(1)}(u)B^{(1)}(u) - C(u)B^{(2)}(u)}{B(u)} \]

where \( A^{(k)}(u) = \sum_{i=0}^{n-k} N_{i,p}^{(k)}(u)w_i \), \( B^{(k)}(u) = \sum_{i=0}^{n-k} N_{i,p}^{(k)}(u)w_i \) and \( k = 1,2 \). The curvature radius of a NURBS curve is defined as:

\[ \rho(u) = \frac{\|C^{(1)}(u)\|^3}{\|C^{(1)}(u) \times C^{(2)}(u)\|} \]

The arc length function \( s(u) \) of a NURBS curve \( C(u) \) over parameter interval \([a, b]\) is defined as:

\[ s(u) = \int_a^b \|C^{(1)}(u)\| \, du \]

Since there is no analytic expression between \( s(u) \) and \( u \), the adaptive quadrature method based on Simpson's rule (Lei et al., 2007) is applied to calculate the arc length, which is omitted here.

2.2. The division of the NURBS curve

In order to simplify the process of feed rate scheduling, we divide the NURBS curve into several blocks. Unlike the reference (Du et al., 2015), which defined the local maximum curvature point whose curvature is greater than the curvature threshold as the critical point, we define the critical point according to the maximum allowable feed rate directly. Here, the maximum allowable feed rate is limited by the chord error, normal acceleration, and normal jerk (Lee et al., 2011).

For a parameter \( u_i \) of the NURBS curve, the chord-error limited feed rate \( v_{i,1} \) is expressed as:

\[ v_{i,1} = \frac{2}{T_s} \sqrt{2\rho_i \delta - \delta^2} \]

where \( T_s \) is the interpolation period, \( \rho_i \) is the curvature radius at point \( C(u_i) \), \( \delta \) is the maximum allowable chord error. The normal acceleration limited feed rate \( v_{i,2} \) and normal jerk limited feed rate \( v_{i,3} \) are as follows:
\[ v_{i,2} = \sqrt{A_n \rho_i} \]  
\[ v_{i,3} = \sqrt{J_n \rho_i} \]

where \( A_n \) and \( J_n \) are the maximum allowable normal acceleration and normal jerk. Then the maximum allowable feed rate \( v_{i,\text{limit}} \) at point \( C(u_i) \) is expressed as:

\[ v_{i,\text{limit}} = \min(F, v_{i,1}, v_{i,2}, v_{i,3}) \]

where \( F \) is the command feed rate.

In this article, the point whose maximum allowable feed rate is local minimal is called a critical point. When the critical points are detected, the NURBS curve is divided into several blocks by these critical points.

3. Proposed method

In this section, the proposed method is detailed. Firstly, the improved S-shape velocity profile is introduced in Section 3.1. The complete velocity profile includes three stages: (1) acceleration stage: the feed rate accelerates from the start velocity \( v_s \) to the command velocity \( F \); (2) uniform stage: the feed rate keeps at the command velocity \( F \); (3) deceleration stage: the feed rate decelerates from the command velocity \( F \) to the end velocity \( v_e \). However, in practice, if the arc length of the block is not long enough, the feed rate may not reach the command velocity before decelerating. Another situation is that when the arc length is too short, even if the feed rate keeps increasing or decreasing, it can’t reach the end velocity before reaching the endpoint of the block. The bidirectional scanning module and the velocity scheduling module are introduced in Section 3.2 and Section 3.3 to solve these problems. In the bidirectional scanning module, the start and the end velocity will be adjusted based on the improved S-shape velocity profile if necessary. In the velocity scheduling module, the actual maximum velocity is calculated first. Then the feed rate profile can be obtained easily. Since the interpolation time is not an integer multiple of the interpolation period in most cases, the round-off error will be introduced. In Section 3.4, the round-off error is eliminated by reducing the actual maximum velocity of a specific block. Then the total interpolation time becomes an integer multiple of the interpolation period. Meanwhile, the interpolation time is extended by no more than one interpolation period. In Section 3.5, the proposed method is summarized.

3.1. Improved S-shape velocity profile

In (Xinhua et al., 2016), five phases S-shape velocity profile based on trigonometric function was applied. The continuous jerk profile reduced the chord errors, and the vibration of the machine tool was suppressed to a certain extent. However, since the acceleration could not maintain at the maximum for a while, the feed rate changed slowly. Therefore, we introduce an improved seven phases S-shape
velocity scheduling function, shown in Fig. 1. When the arc length \( L \) is long enough, the feed rate can accelerate from the start velocity \( v_s \) to the command velocity \( F \), maintain the command velocity \( F \) for a while, and finally decelerates to the end velocity \( v_e \).

Fig. 1 The feed rate, acceleration, and jerk profile.

It is obvious that the proposed velocity profile is more continuous compared with the traditional S-shaped velocity profile, and the acceleration can maintain at the maximum for a while. Its acceleration equation can be given as formula (13):

\[
A(t) = \begin{cases} 
A_1 (1 - \cos(\pi t/T_1))/2 & 0 < t < t_1 \\
A_1 & t_1 < t < t_2 \\
A_1 (1 + \cos(\pi (t - t_2)/T_3))/2 & t_2 < t < t_3 \\
0 & t_3 < t < t_4 \\
-A_2 (1 - \cos(\pi (t - t_4)/T_5))/2 & t_4 < t < t_5 \\
-A_2 & t_5 < t < t_6 \\
-A_2 (1 + \cos(\pi (t - t_6)/T_7))/2 & t_6 < t < t_7 
\end{cases} \tag{13}
\]
where $T_1 = T_3, T_5 = T_7$ and $t_i = \sum_{j=1}^{i} T_j$, $A_1$ and $A_2$ are the actual maximum acceleration in the acceleration stage and the deceleration stage, respectively.

Differentiating equation (13), the jerk equation can be obtained as formula (14):

$$J(t) = \begin{cases} 
A_1 \pi \sin(\pi t/T_1)/(2T_1) & 0 < t < t_1 \\
-A_1 \pi \sin(\pi(t - t_2)/T_3)/(2T_3) & t_1 < t < t_2 \\
-A_2 \pi \sin(\pi(t - t_4)/T_5)/(2T_5) & t_2 < t < t_3 \\
A_2 \pi \sin(\pi(t - t_6)/T_7)/(2T_7) & t_6 < t < t_7 \\
0 & t_3 < t < t_4 \\
0 & t_4 < t < t_5 \\
0 & t_5 < t < t_6 \\
0 & t_6 < t < t_7 
\end{cases} \quad \#(14)$$

Integrating equation (13) gives the velocity equation as formula (15):

$$v(t) = \begin{cases} 
v_o + A_1 t/2 - A_1 T_1 \sin(\pi t/T_1)/(2\pi) & 0 < t < t_1 \\
v_o + A_1 T_1/2 + A_1 (t - t_1) - A_2 T_5 \sin(\pi(t - t_4)/T_5)/(2\pi) & t_1 < t < t_2 \\
v_o + A_1 T_1/2 + A_1 T_2 + A_2 (t - t_2)/2 + A_1 T_3 \sin(\pi(t - t_2)/T_3)/(2\pi) & t_2 < t < t_3 \\
F - A_2 T_5/2 - A_2 (t - t_3) & t_3 < t < t_4 \\
F - A_2 T_5/2 - A_2 T_6 - A_2 (t - t_5)/2 - A_2 T_7 \sin(\pi(t - t_5)/T_7)/(2\pi) & t_4 < t < t_5 \\
F - A_2 T_5/2 - A_2 T_6 - A_2 (t - t_6)/2 - A_2 T_7 \sin(\pi(t - t_6)/T_7)/(2\pi) & t_5 < t < t_6 \\
F - A_2 T_5/2 - A_2 T_6 - A_2 (t - t_7)/2 - A_2 T_7 \sin(\pi(t - t_7)/T_7)/(2\pi) & t_6 < t < t_7 
\end{cases} \quad \#(15)$$

Take the acceleration stage as an example. According to equation (15), we can obtain the following equation:

$$A_1 (T_1 + T_2) = F - v_o \quad \#(16)$$

This formula can be written as:

$$A_1 = \frac{F - v_o}{T_1 + T_2} \quad \#(17)$$

According to equation (14), the actual maximum jerk in the accelerating process is:

$$J_1 = \frac{A_1 \pi}{2T_1} \quad \#(18)$$

When equation (17) is substituted into equation (18), the actual maximum jerk can also be expressed as:

$$J_1 = \frac{\pi(F - v_o)}{2T_1(T_1 + T_2)} \quad \#(19)$$

Since the actual maximum acceleration and jerk can’t exceed the tangential acceleration limitation $A_t$ and the tangential jerk limitation $J_t$ respectively, the following inequality should be satisfied:

$$\begin{cases} 
F - v_o \leq A_t \\
\frac{F - v_o}{T_1 + T_2} \leq J_t \\
\frac{\pi(F - v_o)}{2T_1(T_1 + T_2)} \leq J_t 
\end{cases} \quad \#(20)$$

The acceleration time $T_1 + T_2 + T_3$ should be minimized to maintain high efficiency. Then the selection of the value of $T_1$, $T_2$ and $T_3$ can be transformed into a simple optimization problem as follows:
This problem can be solved with an illustration. As shown in Fig. 2, the X-axis represents $T_1$, and the Y-axis represents $T_1 + T_2$. Since $T_2 \geq 0$, then $T_1 + T_2 \geq T_1$, which means the optimal point $Q = (x, y)$ must be above the line $L_1$. The first constraint in equation (21) can be represented by a line parallel to X-axis, which is noted by $L_2$. The second constraint in equation (21) can be represented by a parabola $c_1$, then point $Q$ must be above the line $L_2$ and the parabola $c_1$. Denote the intersection point of $L_1$ and $c_1$ as point $B$. If $L_2$ below the point $B$, the optimal point $Q$ coincides with point $B$; otherwise, the optimal point $Q$ coincides with point $A$, which is the intersection point of $L_2$ and $c_1$. The optimal solution can be represented by the formula below:

\[
\begin{align*}
T_1 &= \frac{\pi(F - v_s)}{2J_t}, T_2 = 0 \quad \text{if } F - v_s \leq \frac{A_t^2}{2J_t} \\
T_1 &= \frac{\pi A_t}{2J_t}, T_2 = \frac{F - v_s}{A_t} - T_1 \quad \text{else}
\end{align*}
\]

Fig. 2 The schematic diagram of linear programming

To calculate the displacement of the acceleration stage, integrating equation (15) gives the displacement equation as formula (23):
Table 1, which are also the parameters in Section 0 segment with a length of acceleration, velocity, and displacement at any time So far, all the coefficients in equations (obtained. Then, the time of the uniform stage Similarly, \( S_3 = FT_4 \), \( S_5 = s_4 + FT_5 - \frac{(1 - \frac{1}{\pi^2})A_2T_5^2}{4} \), \( S_6 = s_5 + \left( F - \frac{A_2T_5^2}{2} \right)T_6 - \frac{A_2T_5^2}{2} \) and \( S_7 = s_4 + \left( \frac{v_x + F}{2T_1 + T_2} \right)T_3 \). According to equation (23), the displacement of the acceleration stage can be given as:

\[
S_A = \frac{(v_x + F)(2T_1 + T_2)}{2}
\]

Similarly, \( T_5 \), \( T_6 \), \( T_7 \) and the displacement of the deceleration stage \( S_D \) can be obtained. Then, the time of the uniform stage \( T_4 \) can be given as:

\[
T_4 = \frac{L - S_A - S_D}{F}
\]

So far, all the coefficients in equations (13, 14, 15, 23) are known, so the jerk, acceleration, velocity, and displacement at any time \( t \) can be calculated easily.

To demonstrate the advantages of our improved S-shape velocity profile, our velocity profile and the velocity profile in (Xinhua et al., 2016) are applied to a linear segment with a length of 100mm. The start velocity and the end velocity are set as 0 mm/s and 20 mm/s respectively. The other interpolation parameters are listed in Table 1, which are also the parameters in Section 4.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Symbols</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command feed rate</td>
<td>( F )</td>
<td>200 mm/s</td>
</tr>
<tr>
<td>Maximum tangential acceleration</td>
<td>( A_t )</td>
<td>1000 mm/s²</td>
</tr>
<tr>
<td>Maximum centripetal acceleration</td>
<td>( A_n )</td>
<td>1000 mm/s²</td>
</tr>
<tr>
<td>Maximum tangential jerk</td>
<td>( J_t )</td>
<td>30000 mm/s³</td>
</tr>
<tr>
<td>Maximum centripetal jerk</td>
<td>( J_n )</td>
<td>30000 mm/s³</td>
</tr>
<tr>
<td>Maximum chord error</td>
<td>( \delta )</td>
<td>0.001 mm</td>
</tr>
<tr>
<td>Interpolation period</td>
<td>( T_s )</td>
<td>1 ms</td>
</tr>
</tbody>
</table>
As shown in Fig. 3, the blue profiles are the results of our velocity profile, and the black profiles are the results of the velocity profile (Xinhua et al., 2016). It is noted that the feed rate, acceleration, and jerk value are both controlled within the set limitations. When our velocity profile is applied, the acceleration can maintain the maximum for a while, and the jerk can reach the maximum, which implies that the velocity can be accelerated/decelerated faster from the start/command velocity to the command/end velocity. It can also be proved from the interpolation time. When the velocity profile in (Xinhua et al., 2016) is applied, the interpolation time of this linear segment is 0.863s, and the interpolation time is 0.731s when our velocity profile is proposed, which is reduced by 15.30%.
Fig. 3 Simulation results of a linear segment by our velocity scheduling function and velocity scheduling function in (Xinhua et al., 2016). (a) Scheduled feed rate profiles. (b) Acceleration profiles. (c) Jerk profiles.

3.2. Bidirectional scanning module

As mentioned above, when the arc length of a NURBS block is too short, even if the feed rate keeps accelerating or decelerating, it can’t reach the end velocity before reaching the endpoint of the block. To solve this problem, we propose a bidirectional scanning module that consists of forward and backward scanning.

Supposing that there are \( n \) critical points detected in Section 2, the NURBS curve is divided into \( n + 1 \) blocks. The maximum allowable feed rate at each critical point is denoted as \( v_i \) (\( i = 1, 2, \ldots, n \)). Since the feed rate should be zero at the start and end points of the NURBS curve, \( v_0 = 0 \) and \( v_{n+1} = 0 \) are added into the maximum allowable feed rate set \( \{v_i, i = 1, 2, \ldots, n\} \). Then, the maximum allowable feed rate set \( \{v_i, i = 0, 1, 2, \ldots, n, n+1\} \) is adjusted based on the velocity profile in Section 3.1. Finally, \( v_i \) and \( v_{i+1} \) are assigned to the start and end velocity of block \( C_i \) (\( i = 1, 2, \ldots, n, n + 1 \)).

‘Bang-Bang’ control is usually adopted in the traditional bidirectional scanning algorithms, the primary concern of which is to reach the acceleration or torque limit of one axis at any time (Zhao, Zhu, & Ding, 2013). However, the resulting jerk profile is discontinuous. In this article, jerk continuity and acceleration continuity are both considered in the bidirectional scanning module. Both the forward and backward scanning applies the three-phase S-shape acceleration in Section 3.1. The flowchart of STEP I (forward scanning) is shown in Fig. 4, and the details are summarized as follows.
STEP I

Forward scanning is performed from the first to the last NURBS block. Since \( v_0 = 0 \), the tool is guaranteed to start moving from rest at the start point of the NURBS curve. As the chord error and the command feed rate are considered in Section 2, only tangential acceleration and tangential jerk are considered in this step. It should be judged whether the feed rate can reach \( v_{i+1} \) before the tool reaches the end point of \( C_i \). If not, an actual maximum achievable velocity \( v \) needs to be calculated, and \( v_{i+1} \) is updated to \( v \). This process is performed recursively until the last block is reached. Denote \( l_i \) as the arc length of the block \( C_i \). The detailed algorithm is described as follows:

(1) Let \( i = 0 \).

(2) If \( v_{i+1} > v_i \), go to (3); otherwise, go to (6).

(3) If \( v_{i+1} - v_i \leq \frac{\pi A_i^2}{2f_t} \), go to (4); otherwise, go to (5).

(4) Let \( T_1 = T_3 = \frac{\pi (v_{i+1} - v_i)}{2f_t} \) and \( T_2 = 0 \), calculate

\[
s = \frac{(v_{i+1} + v_i)(2T_1 + T_2)}{2} = (v_{i+1} + v_i)\frac{\pi (v_{i+1} - v_i)}{2f_t} \quad \text{#(26)}
\]

If \( s < l_i \), \( v_{i+1} \) does not need to be adjusted; otherwise, the actual maximum velocity \( v \) has the following relationship with \( v_i \) and \( l_i \):

\[
(v + v_i)\sqrt{\frac{\pi (v - v_i)}{2f_t}} = l_i \quad \text{#(27)}
\]

Equation (27) can be transformed to a univariate cubic equation:

\[
v^3 + v_i v^2 - v_i^2 v - v_i^3 - \frac{2l_i A_i}{\pi} = 0 \quad \text{#(28)}
\]

It can be proved that equation (28) has one and only one root \( v_{\text{update}} \) satisfying:

\( v_i < v_{\text{update}} < v_{i+1} \), then \( v_{i+1} \) is updated to \( v_{\text{update}} \), go to (6). (Denote \( l(v) \) as the displacement when accelerating from \( v_i \) to \( v \). Obviously, \( l(v) \) is monotonically increasing with respect to \( v \), since \( l(v_i) - l_i < 0 \) and \( l(v_{i+1}) - l_i > 0 \), there is a unique \( v \in (v_i, v_{i+1}) \) so that \( l(v_i) - l_i = 0 \).)

(5) Let \( T_1 = T_3 = \frac{\pi A_i}{2f_t} \) and \( T_2 = \frac{(v_{i+1} - v_i)}{A_t} - T_1 \), calculate

\[
s = \frac{(v_{i+1} + v_i)(2T_1 + T_2)}{2} = (v_{i+1} + v_i)\frac{(v_{i+1} - v_i)}{A_t} + \frac{\pi A_i}{2f_t} \quad \text{#(29)}
\]

If \( s < l_i \), \( v_{i+1} \) does not need to be adjusted; otherwise, the relationship between actual maximum velocity \( v \), \( v_i \) and \( l_i \) has two conditions:

(5a) The actual maximum velocity \( v \) satisfying: \( v - v_i > \frac{\pi A_i^2}{2f_t} \). Then
\[(v + v_i) \left( \frac{v - v_i}{A_t} + \frac{\pi A_t}{2J_t} \right) = 2l_i \#(30)\]

Equation (30) can be transformed to a univariate quadratic equation:

\[\frac{v^2}{2A_t} - \frac{\pi A_t v}{4J_t} + \frac{\pi A_t v_i}{4J_t} - \frac{v_i^2}{2A_t} - l_i = 0 \#(31)\]

(5b) The actual maximum velocity \(v\) satisfying: \(v - v_i \leq \frac{\pi A_t^2}{2J_t}\). The relationship is the same as equation (27) and (28).

It can be proved that there is one and only one root \(v_{\text{update}}\) satisfying condition (5a) or (5b). Then \(v_{i+1}\) is updated to \(v_{\text{update}}\), go to (6).

(6) Let \(i = i + 1\), if \(i \leq n\), go to (2); otherwise, stop.

**STEP II**

In the second step, backward scanning is performed from the last to the first NURBS block. Since \(v_{n+1} = 0\) and it will not be updated, the tool is guaranteed to stop at the end of the NURBS curve. STEP II is similar to STEP I, and only some minor changes are required as follows: (1) \(i = 0\) is replaced by \(i = n + 1\); (2) \(v_i\) and \(v_{i+1}\) interchange; (3) \(i = i + 1\) is replaced by \(i = i - 1\); (4) \(i \leq n\) is replaced by \(i \geq 1\).

After these two steps, for any NURBS block \(C_i\), the feed rate can be smoothly changed from \(v_{i-1}\) to \(v_i\), which means the smooth trajectory is guaranteed.
3.3. Velocity scheduling module

After the process of the bidirectional scanning module, there are two possible motion states for each block. (1) ACC-or-DEC case: the case of acceleration or deceleration. (2) ACC-and-DEC case: the case of acceleration-uniform-deceleration or acceleration-deceleration. For the ACC-or-DEC case, the actual maximum velocity must be the start velocity or the end velocity, and the feed rate keeps accelerating or
decelerating along the block. For the ACC-and-DEC case, the feed rate at anyone endpoint of this block has not been adjusted by the bidirectional scanning module. If the block is long enough, the feed rate can reach the command velocity $F$. The feed rate accelerates from the start velocity $v_s$ to the command velocity $F$, maintains the velocity $F$ for a while, and finally decelerates to the end velocity $v_e$. If the block is short, the feed rate cannot reach the command velocity $F$. The feed rate accelerates from the start velocity $v_s$ to the actual maximum velocity $v_m$ ($v_m < F$), then decelerates to the end velocity $v_e$. For different cases, the processing details of the velocity scheduling module are as follows.

### 3.3.1. ACC-or-DEC case

Supposing that the start velocity $v_s$ is lower than the end velocity $v_e$, the feed rate keeps accelerating along the block. Since there is no uniform stage and deceleration stage, it is obvious that $T_6 = T_7 = T_8 = 0$.

According to the complete velocity profile in Section 3.1, if $v_e - v_s \leq \frac{\pi A_t^2}{2l_t}$, the acceleration first increases from zero to the actual maximum acceleration $A_1$, and then decreases to zero. According to equations (20) and (23), we can obtain that:

$$\frac{(v_e - v_s)}{T_1} = A_1 # (32)$$

$$\frac{(v_e + v_s)}{T_1} = l_t # (33)$$

Solve the above equation (32, 33), the actual maximum acceleration $A_1$, time interval $T_1$, and $T_3$ can be calculated as follows:

$$A_1 = \frac{v_e^2 - v_s^2}{l_t} # (34)$$

$$T_1 = T_3 = \frac{l_t}{v_e + v_s} # (35)$$

If $v_e - v_s > \frac{\pi A_t^2}{2l_t}$, the acceleration increases from zero to the tangential acceleration limitation $A_t$ first. Then keeps the acceleration constant for a while and decreases to zero. From equation (22), it is obvious that $A_1 = A_t$, $T_1 = T_3 = \frac{\pi A_t}{2l_t}$, and $T_2 = \frac{v_e - v_s}{A_t} - T_1$.

For the situation where the start velocity $v_s$ is higher than the end velocity $v_e$, the feed rate keeps decelerating along the block. The maximum acceleration in the deceleration stage $A_2$, the time intervals $T_5$, $T_6$ and $T_7$ can be obtained by the similar calculation process in the previous paragraph, which is omitted here.

### 3.3.2. ACC-and-DEC case

For this case, the key is whether the actual maximum velocity $v_m$ can reach the command velocity $F$, which can be judged by solving equation (25). If $T_4 \geq 0$, the actual maximum velocity $v_m$ can reach the command velocity $F$, and all the time
intervals and the actual maximum acceleration in the acceleration and deceleration stage can be obtained easily. If \( T_4 < 0 \), the actual maximum velocity \( v_m \) can’t reach the command velocity \( F \), the feed rate accelerates from the start velocity \( v_s \) to the actual maximum velocity \( v_m \), then decelerates to the end velocity \( v_e \), where \( v_m \) satisfying \( \max(v_s, v_e) < v_m < F \). The feed rate profile consists of an acceleration and a deceleration stage. When \( v_m \) is known, the time intervals and the actual maximum acceleration can be obtained by a similar process as in Section 3.3.1.

The only variable that needs to be solved is the actual maximum velocity \( v_m \) now. For simplicity, assuming that \( v_s > v_e \), and the process is similar when \( v_s \leq v_e \). The detailed process of solving the actual maximum velocity \( v_m \) is as follows:

1. Let \( T_1 = T_3 = \frac{\pi A_t}{2 l_t} = T_5 = T_7 \), \( T_2 = \frac{v_m - v_s}{A_t} - \frac{\pi A_t}{2 l_t} \) and \( T_6 = \frac{v_m - v_e}{A_t} - \frac{\pi A_t}{2 l_t} \). According to equation (23), the displacement can be expressed by

\[
s = \frac{(v_m + v_s)(2T_1 + T_2) + (v_m + v_e)(2T_5 + T_6)}{2} \tag{36}
\]

To guarantee the displacement equals to the arc length \( l_t \), a univariate quadratic equation as follows can be obtained according to equation (36).

\[
\frac{2}{A_t} v_m^2 + \pi A_t v_m + \frac{\pi A_t}{2 l_t} (v_s + v_e) - \frac{v_s^2 + v_e^2}{A_t} - 2l_t = 0 \tag{37}
\]

If there exists one root \( v_{m,j} \) satisfying:

\[
v_s + \frac{\pi A_t^2}{2 l_t} < v_{m,j} < F \tag{38}
\]

then \( v_m = v_{m,j} \), go to (4); otherwise, go to (2).

2. Let \( T_1 = T_3 = \sqrt{\frac{\pi (v_m-v_s)}{2 l_t}} \), \( T_2 = 0 \), \( T_5 = T_7 = \frac{\pi A_t}{2 l_t} \) and \( T_6 = \frac{v_m - v_e}{A_t} - \frac{\pi A_t}{2 l_t} \). To guarantee the displacement equals to the arc length \( l_t \), a univariate quartic equation as follows can be obtained according to equation (36).

\[
\frac{1}{4 A_t^2} v_m^4 - \frac{\pi}{4 l_t} v_m^3 + \left[ \frac{\pi^2 A_t^2}{16 l_t^2} + \frac{\pi v_s^2}{2 l_t} \right] v_m^2 + \left[ \frac{\pi v_s^3}{2 l_t} + \frac{\pi A_t v_e}{4 l_t} - \frac{v_e^2}{2 A_t} - l_i \right] v_m + \left( \frac{\pi v_s^2}{2 l_t} \right) ^2 + \left( \frac{\pi A_t v_e}{4 l_t} - \frac{v_e^2}{2 A_t} - l_i \right) ^2 = 0 \tag{39}
\]

If there exists one root \( v_{m,j} \) satisfying:

\[
\max(v_s, v_e) + \frac{\pi A_t^2}{2 l_t} < v_{m,j} < \min(v_s + \frac{\pi A_t^2}{2 l_t}, F) \tag{40}
\]

then \( v_m = v_{m,j} \), go to (4); otherwise, go to (3).

3. Let \( T_5 = T_7 = \sqrt{\frac{\pi (v_m-v_s)}{2 l_t}} \), \( T_1 = T_3 = \sqrt{\frac{\pi v_m v_s}{2 l_t}} \) and \( T_2 = T_6 = 0 \). To guarantee the displacement equals to the arc length \( l_t \), a univariate quartic equation as follows
can be obtained according to equation (36).

\[(v_e - v_s)^2 v_m^4 + \left[2(v_e - v_s)(v_s^2 - v_e^2) - \frac{8J_l l_i^2}{\pi}\right] v_m^3 + \left[(v_s^2 - v_e^2)^2 + 2(v_e - v_s)(v_s^2 - v_e^2) + \frac{2J_l l_i^2}{\pi}\right] v_m^2 + \left[2(v_s^3 - v_e^3) - \frac{8J_l l_i^2}{\pi} v_e\right] v_m + \left(v_s^3 - v_e^2 + \frac{2J_l l_i^2}{\pi}\right)^2 + \frac{8J_l l_i^2}{\pi} v_e^3 = 0\] \(^{(41)}\)

It can be proven that there exists one and only one root \(v_{m,j}\) satisfying:

\[v_s < v_{m,j} < \min(v_e + \frac{\pi A_t^2}{2J_t}, F)\] \(^{(42)}\)

then \(v_m = v_{m,j}\), go to (4).

\((4)\) Stop.

### 3.4. Round-off error elimination module

In most cases, the total interpolation time of the NURBS curve is not an integer multiple of the interpolation period, which causes the round-off error. The standard methods allocate the displacement corresponding to the time of less than one interpolation period to other interpolation periods according to a specific function. However, these methods will lead to the excess of chord error, feed rate, acceleration, or jerk in some points since the displacement of all interpolation periods is increased.

For each NURBS block \(C_i\), its interpolation time \(T_{total}^i = \sum_j^7 T_j^i\), then the total interpolation time of the NURBS curve is \(T_{total} = \sum_{i=1}^{n+1} T_{total}^i = \sum_{i=1}^{n+1} \sum_{j=1}^7 T_j^i\). To eliminate the round-off error, \(T_{total}\) is increased to \(T_{total} + \Delta t\), where the extended time \(\Delta t\) is calculated as follows:

\[\Delta t = \left[\frac{T_{total}/T_s}{T_s} - T_{total}\right]#^{(43)}\]

It is worth noting that \(\Delta t\) is less than the interpolation period, and \(T_{total} + \Delta t\) is an integer multiple of the interpolation period. The start velocity and end velocity of the blocks should not be changed in this module to ensure a continuous velocity profile on the NURBS curve. For the block of ACC-or-DEC case, as \((v_s + v_e)T_{total}^i = 2l_i\), the interpolation time of this block cannot change without changing the start velocity or the end velocity. For the block of ACC-and-DEC case, if the actual maximum velocity \(v_m\) is reduced appropriately, the interpolation time of this block can be increased by \(\Delta t\) without changing the start velocity or end velocity.

In the rest of this section, the superscript \(i\) is omitted for brevity. The first block of the ACC-and-DEC case is chosen to execute the round-off error elimination module. Assuming that \(v_s > v_e\), the process is similar when \(v_s \leq v_e\). Denote the
interpolation time of this block as \( T = \sum_{j=1}^{7} T_j \), where the time intervals \( T_j \) (\( j = 1,2,\ldots,7 \)) are obtained by the velocity scheduling module. In order to increase the interpolation time from \( T \) to \( T + \Delta t \), the round-off error elimination module reduces the actual maximum velocity from \( v_m \) to \( v'_m \). The detailed process of calculating the actual maximum velocity is as follows:

1. If \( v_s + \frac{\pi A_t^2}{2J_t} < v_m \), go to (3); otherwise, go to (2).

2. If \( \max(v_s, v_e + \frac{\pi A_t^2}{2J_t}) < v_m < \min(v_s + \frac{\pi A_t^2}{2J_t}, F) \), go to (4); otherwise, go to (5).

3. Let \( T_1 = T_3 = \frac{\pi A_t}{2J_t} = T_5 = T_7 \), \( T_2 = \frac{v_m-v_s}{A_t} - \frac{\pi A_t}{2J_t} \) and \( T_6 = \frac{v_m-v_e}{A_t} - \frac{\pi A_t}{2J_t} \).

According to equation (23), the displacement can be expressed by

\[
s = \frac{(v'_m+v_s)(2T_1+T_2)+(v'_m+v_e)(2T_5+T_6)}{2} + v'_m(T - 2T_1 - T_2 - 2T_5 - T_6)\tag{44}
\]

To guarantee the displacement equals to the arc length \( l_i \), a univariate quadratic equation as follows can be obtained according to equation (44).

\[
-\frac{1}{A_t}v_m^2 + (T + \frac{v_s}{A_t} \frac{\pi A_t}{2J_t})v'_m + \frac{\pi A_t}{4J_t}(v_s + v_e) - \frac{v_s^2 + v_e^2}{A_t} - l_i = 0\tag{45}
\]

If there exists one root \( v'_{m,j} \) satisfying

\[
v_s + \frac{\pi A_t^2}{2J_t} < v'_{m,j} < v_m\tag{46}
\]

then \( v'_m = v'_{m,j} \), go to (6); otherwise, go to (4).

4. Let \( T_1 = T_3 = \sqrt{\frac{\pi(v'_m-v_s)}{2J_t}} \), \( T_2 = 0 \), \( T_5 = T_7 = \frac{\pi A_t}{2J_t} \) and \( T_6 = \frac{v'_m-v_e}{A_t} - \frac{\pi A_t}{2J_t} \). To guarantee the displacement equals to the arc length \( l_i \), a univariate quartic equation as follows can be obtained according to equation (44).

\[
\frac{1}{4A_t^2}v'_m^4 + \left(\frac{a}{A_t} - \frac{\pi}{2J_t}\right)v'_m^3 + \left(\frac{a^2 + b}{A_t} + \frac{3\pi v_s^2}{2J_t}\right)v'_m^2 + \left(2ab - \frac{3\pi v_s^2}{2J_t}\right)v'_m + b^2 + \frac{3\pi v_s^3}{2J_t} = 0\tag{47}
\]

where \( a = \frac{\pi A_t}{4J_t} - T_s - \frac{v_e}{A_t} \) and \( b = l_i + \frac{v_s^2}{2A_t} - \frac{\pi A_t v_e}{4J_t} \). If there exists one root \( v'_{m,j} \) satisfying:

\[
\max(v_s, v_e + \frac{\pi A_t^2}{2J_t}) < v'_{m,j} < \min(v_s + \frac{\pi A_t^2}{2J_t}, v_m)\tag{48}
\]

then \( v'_m = v'_{m,j} \), go to (6); otherwise, go to (5).

5. Let \( T_1 = T_3 = \sqrt{\frac{\pi(v'_m-v_s)}{2J_t}} \), \( T_5 = T_7 = \sqrt{\frac{\pi(v'_m-v_e)}{2J_t}} \) and \( T_2 = T_6 = 0 \). To guarantee
the displacement equals to the arc length \( l_i \), a univariate quartic equation as follows can be obtained according to equation (44).

\[
T_s^2 v_m'^5 - \left( 2l_i T_s + 3v_e T_s^2 + \frac{J_i a^2}{2\pi} \right) v_m'^4 + \left( l_i^2 + 6l_i T_s v_e + 3v_e^2 T_s^2 - \frac{ab l_i}{\pi} \right) v_m'^3
\]

\[
- \left( 3v_e l_i^2 + 6l_i T_s v_e^2 + v_e^3 T_s^2 + \frac{J_i b^2}{2\pi} + \frac{J_i a c}{\pi} \right) v_m'^2
\]

\[
+ \left( 3v_e^2 l_i^2 + 2l_i T_s v_e^3 - \frac{J_i b c}{\pi} \right) v_m' - \left( v_e^3 l_i^2 + \frac{J_i c^2}{2\pi} \right) = 0\ #(49)
\]

where \( a = \frac{3\pi(v_e-v_o)}{2J_i} - T_s^2 \), \( b = \frac{3\pi(v_e^2-v_o^2)}{2J_i} + 2l_i T_s \) and \( c = \frac{\pi(v_e^3-v_o^3)}{2J_i} - l_i^2 \). It can be proven that there exists one and only one root \( v_{m,j} \) satisfying:

\[
v_{start} < v_{m,j} < \min(v_{end} + \frac{\pi A_i^2}{2J_i}, v_m)\ #(50)
\]

then \( v_m' = v_{m,j} \), go to (6).

(6) Stop.

In order to demonstrate the effect of our round-off error elimination module, our round-off error elimination module and traditional round-off error compensation module are applied to the same linear segment in Section 3.1 under the same condition. As shown in Fig. 5, the blue profile is the feed rate with our round-off error compensation module, and the black profile is the feed rate with the traditional round-off error elimination module. From the enlarged region 1, the black profile is a little above the red dotted line, which means the actual velocity exceeds the given maximum value, while the blue one is below the red dotted line. From the enlarged region 2, we know that the interpolation time of this linear segment with the traditional round-off error compensation module is 0.730s, and the interpolation time with our round-off error elimination module is 0.731s. Although the interpolation time is increased by an interpolation period, the velocity is strictly limited to the given maximum value, so this time loss is worthwhile.
Fig. 5 Simulation results of a linear segment with our round-off error elimination module and the traditional round-off error compensation module.

3.5. Summary of the proposed method

In order to give a clear illustration of the proposed feed rate scheduling method, the flowchart of the proposed method is summarized as follows. First, the parameters of the NURBS curve and the constraints, including chord error and dynamic limitations, are given. Then, the critical points of the NURBS curve are detected, which are utilized to divide the NURBS curve into several blocks. In Section 3.2, the maximum allowable feed rate corresponding to each critical point is adjusted to guarantee the continuity of the feed rate at the junctions between successive NURBS blocks. Later, for each block \( C_i \), the time intervals \( T_j^i \) \( (j = 1, 2, \ldots, 7) \), the actual maximum acceleration \( A_v^i \) and the actual maximum deceleration \( A_a^i \) are calculated following the process in Section 3.3. In order to eliminate the round-off error, the first block with ACC-and-DEC case (whose index equals \( k \)) is found, and the actual maximum velocity \( v_m^i \) is appropriately reduced, thereby extending the interpolation time by \( \Delta t \). The time intervals \( T_j^k \) \( (j = 1, 2, \ldots, 7) \), the actual maximum acceleration \( A_v^k \) and the actual maximum deceleration \( A_a^k \) are also recalculated. Finally, for each time \( t \), the displacement on the NURBS curve can be calculated, which will be utilized to interpolate the NURBS curve toolpath. It is worth noting that several univariate multiple equations are involved in the proposed method. We adopt the eigenvalue method to get the roots of the equation. The polynomial is converted into a companion matrix, then the eigenvalues of the matrix are solved.
4. Simulation and experimental results

In this section, two typical NURBS curves with different degrees are conducted as examples to evaluate the performance of the proposed feed rate scheduling method by simulation. The first one is a trident-shaped second-degree NURBS curve with seven control points. The other is a butterfly-shaped third-degree NURBS curve with 51 control points. The degrees, control points, knot vectors, and weight vectors of these two NURBS curves are given in Appendix A, and Appendix B. Limitations of chord error and kinematic constraints are given in Table 1. Then, the proposed method and the method in (Xinhua et al., 2016) (Denoted by Liu’s method after this) are applied to generate a feed rate profile to compare the effectiveness of the proposed method. In order to further verify the feasibility of the proposed method, a machining experiment of the butterfly-shaped NURBS curve is carried out using a CNC machining experimental platform.

In the interpolation stage, the arc-length compensation and feedback correction method (Zhao, Zhu, & Han, 2013) is applied to calculate the accurate parameter \( u \) of each interpolation point. The theoretical machining time is calculated according to equation (51).

\[
T_{\text{total}} = \sum_{i=1}^{n+1} \sum_{j=1}^{7} T_i^j \tag{51}
\]

The number of interpolation points is also utilized to compare the machining efficiency.

4.1. Simulation results of the trident-shaped curve

The trident-shaped curve geometry model is shown in Fig. 6. Take the reciprocal of the curvature radius of equation (7) to calculate the curvature value on the curve. Fig. 7 shows the curvature of the trident-shaped curve. Based on the chord error and dynamic constraints, five critical points are detected, as shown by the black solid points in Fig. 7. Then the start point and the end point of the trident-shaped curve are also defined as critical points. Since the trident-shaped curve is closed, its start point coincides with its endpoint. So, there are six critical points and six blocks.
According to the proposed feed rate scheduling method, the maximum allowable feed rate of the trident-shaped curve under multiple constraints (the red one) and the scheduled feed rate profile (the blue one) are shown in Fig. 8. We notice that the blue curve is always below the red curve, which means that the chord error, normal acceleration, and normal jerk constraints will not be violated at the critical points. The feed rate, tangential acceleration, tangential jerk, and chord error profiles are shown in Fig. 9. It can be seen that they are all strictly constrained by the values of the given
parameters in Table 1. Meanwhile, the acceleration and jerk profile is continuous, which means the feed rate profile is smooth enough. In addition, Fig. 10 shows that the motion of each axis is smooth. Although the proposed method doesn't exceptionally constrain the axial dynamics, there are no sudden changes in axial velocity or acceleration. Moreover, they are limited to the same order of magnitude as the given tangential dynamics.

The results of Liu's method are demonstrated in Figs. 11 and 12. In Fig. 11, the feed rate, acceleration, jerk, and chord error profile are all constrained by the given limits. The feed rate and acceleration of the X-axis and Y-axis are also shown in Fig. 12. Comparing the proposed method with Liu’s method, the proposed method is relatively efficient. Although the maximum feed rate and acceleration of both two methods can reach the given maximum values, the proposed method can maintain the maximum acceleration for a while. In contrast, Liu's method can only reach the maximum value at some moments. The maximum jerk can reach 29986mm/s³ in the proposed method but only 20920mm/s³ in Liu’s method. As a result of the sufficient utilization of the machine’s kinematic performance, the theoretical machining of the proposed method is 2.953s, which is 14.26% shorter than Liu’s method. Finally, after interpolation calculation, there are 2496 interpolation points in the results of the proposed method and 3429 interpolation points in the results of Liu’s method. The slight difference between the theoretical time and the actual number of interpolation points comes from the accumulation of the length difference between the arc length and the chord length of the NURBS curve. The static comparison of the simulation results is listed in Table 2.
Fig. 8 Limiting feed rate and scheduled feed rate of the trident-shaped curve.

(a)

(b)

(c)

(d)

Fig. 9 Simulation results of the trident-shaped curve by the proposed method. (a)
Scheduled feed rate profile. (b) Tangential acceleration profile. (c) Tangential jerk profile. (d) Chord error profile.

Fig. 10 Simulation results of the trident-shaped curve by the proposed method. (a) X-axis and Y-axis velocity profiles. (b) X-axis and Y-axis acceleration profiles.

Fig. 11 Simulation results of the trident-shaped curve by Liu’s method. (a) Scheduled feed rate profile. (b) Tangential acceleration profile. (c) Tangential jerk profile. (d) Chord error profile.
Fig. 12 Simulation results of the trident-shaped curve by Liu’s method. (a) X-axis and Y-axis velocity profiles. (b) X-axis and Y-axis acceleration profiles.

Table 2 Static comparison of the trident-shaped curve simulation results

<table>
<thead>
<tr>
<th>Test methods</th>
<th>Proposed method</th>
<th>Liu’s Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum feed rate (mm/s)</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>Maximum tangential acceleration (mm/s²)</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>Maximum tangential jerk (mm/s³)</td>
<td>29986</td>
<td>20920</td>
</tr>
<tr>
<td>Maximum chord error (mm)</td>
<td>1.2499 × 10⁻⁴</td>
<td>1.2500 × 10⁻⁴</td>
</tr>
<tr>
<td>theoretical machining time (s)</td>
<td>2.953</td>
<td>3.444</td>
</tr>
<tr>
<td>Number of interpolation point</td>
<td>2946</td>
<td>3429</td>
</tr>
</tbody>
</table>

4.2. Simulation results of the butterfly-shaped curve

The geometry model of the butterfly-shaped curve is shown in Fig. 13. Fig. 14 shows the curvature of the butterfly-shaped curve. Based on the chord error and dynamic constraints, 35 critical points are detected, as shown by the black solid points in Fig. 14. The start point and the endpoint of the butterfly-shaped curve are also defined as critical points. Since the butterfly-shaped curve is closed, its start point coincides with its endpoint. So, there are 36 critical points and 36 blocks.
**Fig. 13** The butterfly-shaped NURBS curve.

**Fig. 14** Curvature of the butterfly-shaped NURBS curve.
The maximum allowable feed rate of the butterfly-shaped curve under multiple constraints and the scheduled feed rate profile are obtained, as shown in Fig. 15. The red curve represents the limiting feed rate profile obtained by equation (12), and the blue curve represents the scheduled feed rate profile. It is noted that the blue curve is always below the red curve, which means that the chord error, normal acceleration, and normal jerk constraints will not be violated at the critical points. The feed rate, tangential acceleration, tangential jerk, and chord error profiles are shown in Fig. 16. It can be seen that they are all strictly constrained by the values of the given parameters in Table 1. Meanwhile, the acceleration and jerk profile is continuous, which means the feed rate profile is smooth enough. In addition, Fig. 17 shows the smooth motion of each axis.

The results of Liu’s method are demonstrated in Figs. 18 and 19. In Fig. 18, the feed rate, acceleration, jerk, and chord error profile are all constrained by the given limits. The feed rate and acceleration of the X-axis and Y-axis are also shown in Fig. 19. The maximum feed rate of the proposed method can reach $188\text{ }\text{mm/s}$, while it can only reach $166\text{ }\text{mm/s}$ using Liu’s method, which has been decreased by 11.7%. Although the maximum acceleration and jerk of both two methods can reach the given maximum values, the acceleration can maintain the maximum value for a while in the proposed method. In contrast, it can only reach the maximum value at some time in Liu’s method. The theoretical machining of the proposed method is $5.303\text{ }\text{s}$, which is 4.67% shorter than Liu’s method. Finally, after interpolation calculation, there are 5296 interpolation points in the results of the proposed method and 5554 interpolation points in the results of Liu’s method. The static comparison of the simulation results is listed in Table 3. Comparing with Liu’s method, the proposed method leads to greater machining efficiency. Moreover, it is worth noting that it performs better in trident-shaped curves than in butterfly-shaped ones. The reason is that the butterfly-shaped curve has a more complex geometric shape, more significant curvature, and its derivative, resulting in a smaller allowable maximum feed rate on the NURBS toolpath. In most cases, the acceleration needs to be reduced before reaching the maximum value, which makes the two methods plan the same feed rate profile. Nevertheless, the proposed method still has certain advantages in efficiency.
Fig. 15 Limiting feed rate and scheduled feed rate of the butterfly-shaped curve.

Fig. 16 Simulation results of the butterfly-shaped curve by the proposed method. (a) Scheduled feed rate profile. (b) Tangential acceleration profile. (c) Tangential jerk profile. (d) Chord error profile.
Fig. 17 Simulation results of the butterfly-shaped curve by the proposed method. (a) X-axis and Y-axis velocity profiles. (b) X-axis and Y-axis acceleration profiles.

Fig. 18 Simulation results of the trident-shaped curve by Liu’s method. (a) Scheduled feed rate profile. (b) Tangential acceleration profile. (c) Tangential jerk profile. (d) Chord error profile.
Fig. 19 Simulation results of the trident-shaped curve by Liu’s method. (a) X-axis and Y-axis velocity profiles. (b) X-axis and Y-axis acceleration profiles.

Table 3 Static comparison of butterfly-shaped curve simulation results

<table>
<thead>
<tr>
<th>Test methods</th>
<th>Proposed method</th>
<th>Liu's Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum feed rate (mm/s)</td>
<td>188</td>
<td>166</td>
</tr>
<tr>
<td>Maximum tangential acceleration (mm/s²)</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>Maximum tangential jerk (mm/s³)</td>
<td>29986</td>
<td>29985</td>
</tr>
<tr>
<td>Maximum chord error (mm)</td>
<td>1.2500 × 10⁻⁴</td>
<td>1.2500 × 10⁻⁴</td>
</tr>
<tr>
<td>theoretical machining time (s)</td>
<td>5.303</td>
<td>5.563</td>
</tr>
<tr>
<td>Number of interpolation point</td>
<td>5296</td>
<td>5554</td>
</tr>
</tbody>
</table>

4.3. Experimental results

In order to check whether the proposed method can guarantee the final machining quality, the butterfly-shaped NURBS curve is machined on the experimental platform shown in Fig. 20. For safety reasons, the command feed rate is set to 50 mm/s. The proposed method is implemented on a computer with the Intel i7-7700 HQ CPU and 8 GB RAM. The calculated interpolation points are transmitted to the CNC machining experimental platform to machine the butterfly-shaped NURBS curve. All three axes are driven by GR 3000 series servomotors, and the axial position feedbacks are collected with a sampling frequency of 1 kHz from the motor encoders.

The actual feed rate is calculated according to the axial position feedback and plotted in Fig. 21, blue. The feed rate of the tool almost does not exceed the given command feed rate $F = 50 \text{mm/s}$. Even if the given command feed rate was exceeded, the proportion of exceeding is tiny, less than 0.4%. Due to the high sampling frequency, the shape of the actual feed rate is zigzag, which can be seen from the enlarged view. The actual feed rate varies with the curvature, and the changing of the feed rate is smooth in all parts of the curve. For the critical points with large curvature, such as $A$ and $B$, the feed rate is reduced significantly due to the chord error and dynamic limitations. The machining results are shown in Fig. 22.
**Fig. 20** The CNC machining experimental platform.

**Fig. 21** The actual feed rate and its details.
5. Conclusions and future works

This paper proposes a novel and complete S-shape feed rate scheduling approach which consists of three modules. Firstly, the bidirectional scanning module based on our improved velocity profile is implemented to guarantee the continuity of the feed rate at the junctions between successive NURBS blocks. Then, the NURBS blocks are classified into two cases (i.e., ACC-or-DEC case and ACC-and-DEC case), and two different velocity scheduling steps are implemented. Finally, the round-off error elimination module is proposed to decrease the actual maximum feed rate of a specific NURBS block appropriately, which leads to the total interpolation time becoming an integer multiple of the interpolation period. The benchmarks prove the applicability of the proposed method.

The advantages of our method are summarized as follows:

(1) The proposed method is complete, and different velocity scheduling schemes are used for blocks of different lengths, which means it can handle more complex curves (such as the butterfly-shaped curve).

(2) The proposed method generates a smoother feed rate profile with continuous acceleration and jerk. Meanwhile, the actual maximum acceleration and jerk are under the confined range strictly.

(3) The total interpolation time is an integer multiple of the interpolation period, which eliminates the round-off error.

(4) Compared with the traditional method, the proposed method can save the
interpolation time by 4.67% to 14.26%.

In future studies, the author intends to expand the applicable object of this method from three-axis toolpaths to five-axis toolpaths to develop its potential further. In addition, the normal and axial kinematic characteristics will also be considered.

Acknowledgement
This work has been supported by National Key Research and Development Program of China (Grant No.2020YFA0713700), National Natural Science Foundation of China (Grants No. 12171023, No. 12001028, No. 62102013, and No. 62141605), China Postdoctoral Science Foundation (Grant No. 2021M690303), and the Fundamental Research Funds for the Central Universities (YWF-22-L-639)

Compliance with Ethical Standards
The authors declare that they have no conflict of interest that could have direct or potential influence or impart bias on the research reported in this paper. Consent to submit the paper for publication has been received explicitly from all co-authors.

Appendix A: Parameters of the trident NURBS curve
The degree: $p = 2$.
The control point (mm): $P= [(60, 0), (120, 120), (72, 48), (60, 120), (48, 48), (0, 120), (60, 0)]$.
The knot vector: $U= [0, 0, 0, 0.2, 0.4, 0.6, 0.8, 1, 1, 1]$.
The weight vector: $W= [1, 1, 1, 1, 1, 1, 1, 1]$.

Appendix B: Parameters of the butterfly-shaped NURBS curve
The degree: $p = 3$.
The control point (mm): $P= [(54.493, 52.139), (55.507,52.139), (56.082, 49.615), (56.780, 44.971), (69.575,51.358), (77.786, 58.573), (90.526, 67.081), (105.973,63.801), (100.400, 47.326), (94.567, 39.913), (92.369,30.485), (83.440, 33.757), (91.892, 28.509), (89.444,20.391), (87.621, 4.830), (80.945, 9.276), (79.834, 14.940), (49.220, 19.828), (52.060,24.994), (53.305, 36.359), (54.492, 22.122), (53.680, 36.359), (56.925, 24.995),(59.765, 19.828), (54.493, 14.940), (49.220, 19.828), (52.060,24.994), (53.305, 36.359), (48.992, 22.122), (44.814, 16.865),(38.802, 12.551), (32.911, 8.521), (29.152, 14.535), (28.040,9.267), (21.364, 4.830), (25.768, 15.447), (19.539, 20.391),(17.097, 28.512), (25.537, 33.750), (16.602, 30.496), (14.199,39.803), (8.668, 47.408), (3.000, 63.794), (18.465, 67.084),(31.197, 58.572), (39.411, 51.358), (52.204, 44.971), (52.904,49.614), (53.478, 52.139), (54.492, 52.139)].
The knot vector: $U = [0, 0, 0, 0, 0.0083, 0.015, 0.0361, 0.0855, 0.1293, 0.1509, 0.1931, 0.2273, 0.2435, 0.2561, 0.2692, 0.2889, 0.3170, 0.3316, 0.3482, 0.3553, 0.3649, 0.3837, 0.4005, 0.4269, 0.4510, 0.4660, 0.4891, 0.5000, 0.5109, 0.5340, 0.5489, 0.5731, 0.5994, 0.6163, 0.6351, 0.6447, 0.6518, 0.6683, 0.6830, 0.7111, 0.7307, 0.7439, 0.7565, 0.7729, 0.8069, 0.8491, 0.8707, 0.9145, 0.9639, 0.9850, 0.9917, 1.0, 1.0, 1.0, 1.0]$

The weight vector: $W = [1.0, 1.0, 1.0, 1.2, 1.0, 1.0, 1.0, 1.0, 1.0, 1.2, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]$

Reference


Liu, Huan, Qiang, Sun, Pengpeng, Qitong, Yuan, & Songmei. (2017). The optimal


Sang, Y., Yao, C., Lv, Y., & He, G. (2020). An improved feedrate scheduling method...


