The International Journal of Advanced Manufacturing Technology (2018) 98:905–918
https://doi.org/10.1007/s00170-018-2242-0

ORIGINAL ARTICLE

A real-time interpolation strategy for transition tool path with C^2 and G^2 continuity

Hui Wang[1] · Jianhua Wu[1] · Chao Liu[1] · Zhenhua Xiong[1]

Received: 10 September 2016 / Accepted: 30 May 2018 / Published online: 16 June 2018

Abstract

A typical interpolation strategy for line segments consists of a transition scheme, a Look-ahead ACC/DEC scheduling, and an interpolation algorithm. In these three parts, the main computation occurs in the first and second part. Some research work has been carried out to decrease the computation in the previous literatures, but these methods occupy a lot of computing resources for the optimization process during the calculation of transition curve parameters and feed rates. Consequently, the computational efficiency of interpolation strategy is greatly reduced. To deal with the issue, a real-time interpolation strategy is proposed in this paper. In the transition scheme, a Bézier curve is utilized to smooth the line segments. Based on the relationship among the approximation error, the approximation radius, and the transition curve, the curve can be directly generated when the approximation error is given. In the ACC/DEC scheduling, a 3-segment feed rate profile with jerk continuity is constructed. Meanwhile, a Look-ahead planning based on Backward Scanning and Forward Revision (BSFR) algorithm is utilized to eliminate redundant computation. Compared with Zhao's and Shi's strategy, the proposed strategy has the merits of C^2 and G^2 continuity for the tool path, jerk continuity for the tool movement, and distinguished real-time performance for interpolation. The experiments of 3D pentagram and 2D butterfly are carried out with different strategies and their results demonstrate that the interpolation efficiency can be greatly improved with the proposed strategy.

Keywords Real-time interpolation strategy · Transition module · Look-ahead ACC/DEC scheduling module · Interpolation module · Real-time interpolation task

1 Introduction

With the advantages of simple structure and calculation, line segments (or G01 blocks) have been widely utilized to describe the complicated surfaces and tool paths in the field of numerical control (NC), but there are still some serious drawbacks in the application [1]. Firstly, the derivates are not the same at the junctions of line segments. The velocity changes abruptly when the tool moves across the junction [2], which can lead to machine vibration. Secondly, it is difficult to achieve high-speed machining during the acceleration/deceleration (ACC/DEC) scheduling for line segments, because the line segments are too short and the scheduling is very complicated [3]. Moreover, a great many line segments are required to approximate the tool paths, and it also increases the burden of data processing [4]. To deal with this issue and realize real-time interpolation, some research work has been carried out to decrease the computation from two aspects, i.e., the transition scheme and the Look-ahead ACC/DEC scheduling. Parametric curves are utilized to transit line segments into a smooth tool path to avoid the varying velocity at junctions, and feed rate profile is generated gradually by combining the ACC/DEC scheduling with the Look-ahead planning to guarantee high-efficient machining. Therefore, to further improve the interpolation efficiency, both the transition scheme and the Look-ahead ACC/DEC scheduling should be comprehensively considered.

In the transition scheme, the smoothness of tool path generated by the transition scheme can be evaluated through parametric continuity and geometric continuity [5]. If the path has nth continuous order derivative, it has C^n parametric continuity. If the path has common tangent direction, continuous

✉ Zhenhua Xiong
mexiong@sjtu.edu.cn

[1] State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

curvature, and continuous curvature variation, it has G^1, G^2, and G^3 geometric continuity, respectively. Thus, both parametric continuity and geometric continuity should be considered simultaneously. To achieve G^1 continuity, He et al. [6] proposed a micro-line transition algorithm based on Ferguson spline that was affected by affine invariants, but the invariants were not easy to control. In [7], a cubic spline curve approach was introduced to rapidly generate the path with G^1 continuity. Bi et al. [8] proposed a cubic Bézier curve transition method to achieve G^2 continuity; however, the method was unable to divide the transition curve into two symmetrical curves for ACC/DEC scheduling. Zhao et al. [9] put forward a cubic B-spline with subdivision method. The B-spline curve can be symmetrically divided and controlled by seven control points, while the control points should be optimized. To avoid the optimization, Shi et al. [10, 11] proposed a transition method based on Pythagorean Hodograph (PH) curve and implemented the method on three-axis and five-axis tool machining. However, C^2 continuity was unachievable. Recently, Dai et al. [12] adopted a quintic Bézier curve for robot trajectory planning with C^2 and G^2 continuity. The C^2 continuity was achieved by the reasonable distribution of control points and re-parameterization of time-shift variables. The approximation error was uncontrollable because the relationship between the error and the curve was not discussed. In NC machining, the approximation error must be controlled. Therefore, the relationship among the approximation error, the approximation radius, and the quintic Bézier curve is deduced in this article to deal with this problem, in which the approximation error is given during the curve generation.

In the Look-ahead ACC/DEC scheduling, an S-shape ACC/DEC scheduling is widely utilized to generate the feed rate profile of the tool path. In general, a 7-segment ACC/DEC profile based on piecewise function was used [13, 14], and 17 types of feed rate profiles were deduced. To simplify the algorithm, a 5-segment ACC/DEC profile was presented [15]. However, these methods only achieve the acceleration continuity. In some applications, undesired effects on the kinematic chains and the inertial loads may be generated due to jerk vibration. Although the ACC/DEC scheduling with jerk continuity could still be deduced by piecewise function, up to 15 segments are needed and many more judgment conditions are introduced, which is unsuitable for real-time application. To realize the jerk continuity, Huang et al. [16] proposed an ACC/DEC scheduling based on sine series, whose order should be determined in advance. Chen et al. put forward an ACC/DEC scheduling based on quintic polynomial [17], but the actual maximal acceleration may not be consistent with the predetermined maximum. To obtain the jerk continuity with simplicity and reliability, an ACC/DEC scheduling based on Bézier curve is proposed here, and 20 types of feed rate profiles are deduced. When the proposed scheduling and the Look-ahead planning are combined with each other, a feed rate profile can be generated in real time.

Generally, the core of Look-ahead planning is a bidirectional scanning algorithm, which includes a backward scanning and a forward scanning. The bidirectional scanning algorithm was first designed by Dong et al. [18] to realize the feed rate profile with the shortest processing time. Based on this algorithm, the constrained feed rate optimization algorithm was then put forward to generate the feed rate profile by considering of the machine's kinematics and dynamics [19]. However, the algorithm was only applied in parametric curves. To solve the problem, Mattmuller et al. [20] modified the algorithm and made it suitable for line segments. Since the backward scanning and the forward scanning were implemented simultaneously, the algorithm required the ACC/DEC scheduling twice for each ACC/DEC scheduling unit (ASU). To further reduce the computation, this paper puts forward a BSFR algorithm, in which the ACC/DEC scheduling does not carry out twice for each ASU by simplifying the forward scanning. When the BSFR algorithm combines with the proposed ACC/DEC scheduling, a Look-ahead ACC/DEC scheduling module is constructed. Thus, the feed rate profiles of ASUs can be generated in real time.

Based on the above analysis, a real-time interpolation strategy for linear segments is proposed by taking into account of the transition scheme and the Look-ahead ACC/DEC scheduling comprehensively. The remainder of this article is organized as follows: In Sect. 2, the principle, the characteristic, and the algorithm of quintic Bézier transition curve are presented. The generation of smooth tool path with C^2 and G^2 continuity is given in detail. Section 3 gives the selection criterion of ASU. Meanwhile, a Look-ahead ACC/DEC scheduling based on BSFR algorithm is proposed. In Sect. 4, a detailed real-time interpolation process is described. The implementation and performance of the strategy are shown in Sect. 5. The conclusion is given in Sect. 6.

2 Transition scheme

2.1 Quintic Bézier transition curve

As shown in Fig. 1a, the solid black lines $\overrightarrow{Q_0Q_1}$ and $\overrightarrow{Q_1Q_2}$ represent two line segments. θ is the angle between $\overrightarrow{Q_0Q_1}$ and $\overrightarrow{Q_1Q_2}$, and the junction is Q_1. Due to tangency and curvature discontinuities at the junction, the velocity changes abruptly when the tool moves across the junctions. To deal with this issue, a transition method is introduced, which adopts a quintic Bézier curve $\overrightarrow{C}(u)$ (the red dotted line) to transit line segments. The computational procedure of $\overrightarrow{C}(u)$ is shown in Fig. 1b, which mainly consists of the following three steps:

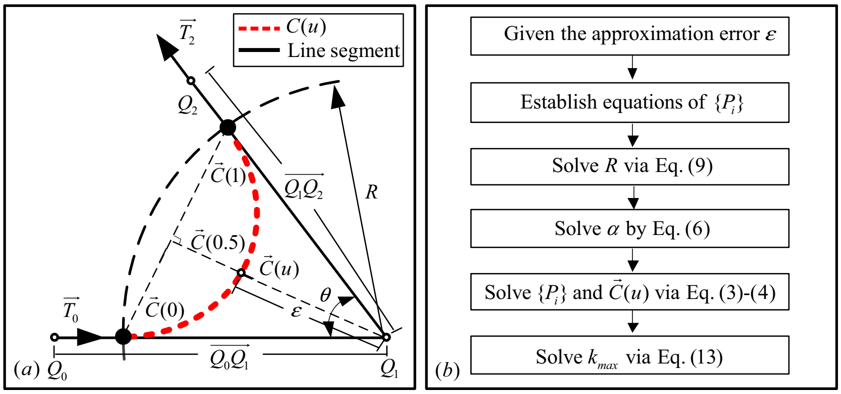

$\vec{T}_2$
$C(u)$
Line segment
Q_2
$\vec{C}(1)$
$\overrightarrow{Q_1Q_2}$
R
$\vec{C}(0.5)$
$\vec{C}(u)$
θ
ε
$\vec{T}_0$
$\vec{C}(0)$
(a) Q_0
$\overrightarrow{Q_0Q_1}$
Q_1
(b)
Given the approximation error ε
Establish equations of $\{P_i\}$
Solve R via Eq. (9)
Solve α by Eq. (6)
Solve $\{P_i\}$ and $\vec{C}(u)$ via Eq. (3)-(4)
Solve k_{max} via Eq. (13)

Fig. 1 Schematic diagrams of the transition method. **a** Line segments smoothed by the Bézier curve. **b** The computational procedure of the curve

Step 1. Establish equations of control points $\{P_\mathrm{i}\}$.

The expression of the quintic Bézier curve is [21]

$$\vec{C}(u)=\sum_{i=0}^{5}B_{i,5}(u)P_i \tag{1}$$

where u is the parameter with range of $u\in[0,1]$, and $\{B_{i,5}\}$ are the Bernstein polynomials. $\{P_\mathrm{i}\}$ are the control points. To generate a smooth tool path with C^2 continuity, line segments and the curve should have the same second derivatives at their junctions (two solid black circles in Fig. 1a). Thus, six equations are confirmed as

$$\begin{cases}\vec{C}(0)=P_0, & \vec{C}(1)=P_5\\ \vec{C}'(0)=\alpha\vec{T}_0, & \vec{C}'(1)=\alpha\vec{T}_2\\ \vec{C}''(0)=\vec{0}, & \vec{C}''(1)=\vec{0}\end{cases} \tag{2}$$

where $\vec{T}_0$ and $\vec{T}_2$ are the direction vectors of $\overrightarrow{Q_0Q_1}$ and $\overrightarrow{Q_1Q_2}$, respectively. $\vec{C}(0)$ and $\vec{C}(1)$ are the ends of curve, and the corresponding derivatives are $\vec{C}'(0)$ and $\vec{C}'(1)$. Similarly, the corresponding second derivatives are $\vec{C}''(0)$ and $\vec{C}''(1)$, α denotes the magnitude of $\vec{C}'(0)$ and $\vec{C}'(1)$, and $\vec{0}$ denotes zero vectors. From Eq. (1), it can be seen that the control points P_0 and P_5 are $\vec{C}(0)$ and $\vec{C}(1)$, respectively. Since the curve is symmetrical, $\vec{C}(0)$ and $\vec{C}(1)$ can be determined as

$$\begin{cases}\vec{C}(0)=P_0=Q_1-R\vec{T}_0\\ \vec{C}(1)=P_5=Q_1+R\vec{T}_2\end{cases} \tag{3}$$

where R is the approximation radius. Then, the equations of $\{P_\mathrm{i}\}$ are set up via Eqs. (1)–(3) and written as

$$\begin{aligned}P_1&=P_0+\frac{\alpha\vec{T}_0}{5},P_2=P_0+\frac{2\alpha\vec{T}_0}{5},P_3\\&=P_0-\frac{2\alpha\vec{T}_2}{5},P_4=P_0-\frac{\alpha\vec{T}_2}{5}\end{aligned} \tag{4}$$

As seen in Eq. (4), once the value α is calculated, all the control points will be determined.

Step 2. Solve equations of $\{P_\mathrm{i}\}$.

Define that the Euclidean norms of $\vec{C}'(0)$, $\vec{C}'(0.5)$, and $\vec{C}'(1)$ are equal to the value α [12]. In this way, the following equation is written as

$$\left\|\vec{C}'(0.5)\right\|=\left\|\vec{C}'(0)\right\|=\left\|\vec{C}'(1)\right\|=\alpha \tag{5}$$

Through solving Eq. (5), the value α is obtained as follows:

$$\alpha=\frac{30R\left|\vec{T}_0+\vec{T}_2\right|}{7\left|\vec{T}_0+\vec{T}_2\right|+16} \tag{6}$$

Substituting Eqs. (4) and (6) into Eq. (1), the curve $\vec{C}(u)$ can be obtained. Then, the re-parameterization [22] is implemented for the curve, which scales the parameter u from $u\in[0,1]$ to $u\in[0,\alpha]$. Consequently, the following equations are achieved as

$$\begin{cases}\vec{C}(0)=P_0, & \vec{C}(\alpha)=P_5\\ \vec{C}'(0)=\vec{T}_0, & \vec{C}'(\alpha)=\vec{T}_2\\ \vec{C}''(0)=\vec{0}, & \vec{C}''(\alpha)=\vec{0}\end{cases} \tag{7}$$

From Eq. (7), it can be seen that the first derivatives at the ends of curve are changed to unit vectors and the second derivatives at the ends of curve are equal to zero vectors. Therefore, when the approximation radius R is given, a smooth tool path is C^2 continuous. Meanwhile, since the second derivatives at the ends of curve are zero vectors, the tool path is also G^2 continuous (as seen in Eq. (10)). However, not only the smooth tool path should be generated, but also the approximation error ε should be derived and controlled in NC machining. In Step 3, the relationship among ε, R, and $\vec{C}(u)$ is deduced.

Step 3. Solve the approximation error ε and curvature maximum $k_{\max}$.

The approximation error ε defines the distance between the curve and the junction of line segments. From Fig. 1a, the error ε is defined as the distance between Q_1 and $\overrightarrow{C}(0.5)$

$$\varepsilon = \left\| Q_1 - \overrightarrow{C}(0.5) \right\| \tag{8}$$

With Eqs. (1), (3), (4), and (8), the relationship between ε and R is determined as

$$R = \frac{2\varepsilon}{\sqrt{2-2\overrightarrow{T}_0 \cdot \overrightarrow{T}_2}} \left(\frac{128 + 56\left|\overrightarrow{T}_0 + \overrightarrow{T}_2\right|}{128-19\left|\overrightarrow{T}_0 + \overrightarrow{T}_2\right|} \right) \tag{9}$$

Based on the above deduction, the curve $\overrightarrow{C}(u)$ can be calculated when ε is given. The computational procedure is $\varepsilon \to R \to \alpha \to \{P_\mathrm{i}\} \to \overrightarrow{C}(u)$. The corresponding equations are Eqs. (9), (6), (3), (4), and (1), respectively.

In addition, the maximum curvature $k_{\max}$ is also deduced, which represents the maximum bending degree of curve. $k_{\max}$ is utilized to determine the minimum feed rate of tool in NC machining. The curvature is expressed as [8]

$$k(u) = \frac{\left\| \overrightarrow{C}'(u) \times \overrightarrow{C}''(u) \right\|}{\left\| \overrightarrow{C}'(u) \right\|^3} \tag{10}$$

To solve Eq. (10), the derivatives of $\overrightarrow{C}(u)$ should be deduced in advance. Through deriving Eq. (1), the first and second derivative of $\overrightarrow{C}(u)$ are obtained as

$$\overrightarrow{C}'(u) = A\overrightarrow{T}_0 + B\overrightarrow{T}_2, \quad \overrightarrow{C}''(u) = C\overrightarrow{T}_0 + D\overrightarrow{T}_2 \tag{11}$$

where A, B, C, and D are the coefficients as follows:

$$\begin{cases} A = 5(6R-3\alpha)u^4 + 4(-15R + 8\alpha)\Big)u^3 + 3(10R-6\alpha)u^2 + \alpha \\ B = 5(6R-3\alpha)u^4 + 4(-15R + 7\alpha)u^3 + 3(10R-4\alpha)u^2 \\ C = 20(6R-3a)u^3 + 12(-15R + 8a)u^2 + 6(10R-6a)u \\ D = 20(6R-3a)u^3 + 12(-15R + 7a)u^2 + 6(10R-4a)u \end{cases} \tag{12}$$

Since $\overrightarrow{C}(u)$ is symmetrical, $k_{\max}$ is located at the middle of $\overrightarrow{C}(u)$. With Eqs. (10)–(12), $k_{\max}$ is determined as

$$k_{\max} = \frac{3 \times (1.875R\text{-}0.4375\alpha) \times \alpha\sin\theta}{\left(2 \times (1.875R-0.4375\alpha)^2 \times (1-cos\theta)\right)^{(3/2)}} \tag{13}$$

2.2 Transition module

Depending on the transition method, NC system can quickly generate a series of curves and utilize the curves to smooth line segments. However, when line segments are too short or the angles between them are too large, the generated curves may interlace with each other. To deal with the issue, a rectification algorithm is put forward here. With the cooperation of the rectification algorithm and the transition method, the interlacement can be eliminated.

The cooperation mechanism is described as below: During the curve generation, the transition method first evaluates the approximation radius R based on the relationship between ε and R (calculated by Eq. (9)). After that, the rectification algorithm judges whether there exists the phenomenon of interlacement by comparing the radius R with the length of adjacent line segments. For instance, $\left\|\overrightarrow{Q_0Q_1}\right\|$ and $\left\|\overrightarrow{Q_1Q_2}\right\|$ denote the length of adjacent line segments (as shown in Fig. 1a). If $R \le min\left(\left\|\overrightarrow{Q_0Q_1}\right\|/2,\ \left\|\overrightarrow{Q_1Q_2}\right\|/2\right)$, the phenomenon of interlacement may not happen and then the transition method generates the curve based on the radius R. If $R > min\left(\left\|\overrightarrow{Q_0Q_1}\right\|/2,\ \left\|\overrightarrow{Q_1Q_2}\right\|/2\right)$, the interlacement happens, and the rectification algorithm is activated which modifies the radius R to$R = \min\left(\left\|\overrightarrow{Q_0Q_1}\right\|/2,\ \left\|\overrightarrow{Q_1Q_2}\right\|/2\right)$. Then, the transition method generates the curve based on the modified radius R. With the same method, not until all line segments are smoothed, the rectification algorithm could be inactivated. For the ease of description, this mechanism is named as the transition module. Consequently, a smooth tool path including line segments and curves is obtained.

3 Look-ahead ACC/DEC scheduling based on BSFR algorithm

To generate the feed rate profiles in real time, the Look-ahead ACC/DEC scheduling is proposed, which includes the selection criterion of ASU, the ACC/DEC scheduling, and the BSFR algorithm.

3.1 Selection criterion of ASU

As illustrated in Fig. 2a, a typical tool path consists of a series of line segments $\overrightarrow{L}_0, \cdots, \overrightarrow{L}_i, \overrightarrow{L}_{i+1}, \cdots, \overrightarrow{L}_n$. By adopting the transition module, a smooth tool path is generated, which is composed of line segments $\overrightarrow{L}_0, \cdots, \overrightarrow{L}_i, \overrightarrow{L}_{i+1}, \cdots, \overrightarrow{L}_n$ and curves $\overrightarrow{C}_0, \cdots, \overrightarrow{C}_i, \overrightarrow{C}_{i+1}, \cdots, \overrightarrow{C}_n$. Since there are a lot of curves in the path, the curvature of path fluctuates frequently. In view of this, the tool should adjust its feed rate in real time

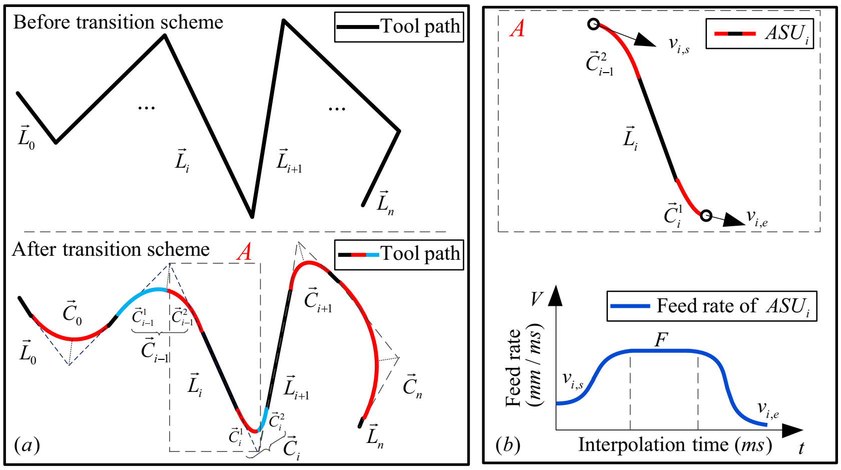


Fig. 2 Diagrams of selection criterion of ASU. **a** Smooth tool path. **b** An ASU and its feed rate profile

as the curvature changes. Accordingly, the path is divided into several ASUs at the maximum curvature of each curve. Then, the ACC/DEC scheduling is implemented for these ASUs, respectively. The selection criterion of ASU is as follows:

For example, one of ASUs, named as ASU$_i$, can be seen in Frame A in Fig. 2b, which contains one line segment $\overrightarrow{L}_i$ and two curves $\overrightarrow{C}^2_{i-1}$ and $\overrightarrow{C}^1_i$. Here, $\overrightarrow{C}^2_{i-1}$ and $\overrightarrow{C}^1_i$ are obtained by dividing $\overrightarrow{C}_{i-1}$ and $\overrightarrow{C}_i$ into halves, and their lengths are calculated by the adaptive Simpson's rule [23]. Based on the above-mentioned analysis, the start point of ASU$_i$ is the middle point of $\overrightarrow{C}_{i-1}$, which is corresponding to the maximum curvature of $\overrightarrow{C}_{i-1}$. Similarly, the end point of ASU$_i$ is corresponding to the maximum curvature of $\overrightarrow{C}_i$. To ensure the machining quality of ASU$_i$, the velocities of the two points are generated under system limits such as the chord error δ, the maximum feed rate F, the maximum acceleration $a_{\max}$, and the maximum jerk $j_{\max}$. Therefore, $v_{i,s}$, named as the theoretical velocity at the start point of ASU$_i$, can be restricted as [24, 25]

$$v_{i,s} = min\left(\frac{2}{T}\sqrt{\frac{1}{k^2_{i,\max}} - \left(\frac{1}{k_{i,\max}} - \delta\right)^2}, \ \sqrt{\frac{a_{\max}}{k_{i,\max}}}, \sqrt[3]{\frac{j_{\max}}{k^2_{i,\max}}}, \ F\right) \tag{14}$$

where T denotes the interpolation period. $k_{\text{i,max}}$ is the curvature at the start point of ASU$_i$, which is also the maximum curvature of $\overrightarrow{C}_i$. Similarly, $v_{\text{i,e}}$, named as the theoretical velocity at the endpoint of ASU$_i$, can also be obtained by Eq. (14). Based on these velocity values, an ACC/DEC scheduling is implemented for ASU$_i$, which aims to generate the corresponding feed rate profile (the blue line in Fig. 2b).

3.2 ACC/DEC scheduling based on Bézier curve

In this section, an ACC/DEC scheduling based on Bézier curve is proposed, which can realize the jerk continuity. The scheduling is deduced as follows: As seen in Fig. 3a, $V(t)$ denotes a standard S-shape feed rate profile. $S(t)$, $A(t)$, and $J(t)$ denote the corresponding displacement, the acceleration, and the jerk profile, respectively. $V(t)$ can be divided into three segments, which are the acceleration segment, the constant segment, and the deceleration segment, respectively. Here, the constant segment is depicted as a straight line, and other segments are denoted as Bézier curves. For the ease of description, $V_a(t)$ and $V_d(t)$ represent the acceleration and deceleration segment of the feed rate profile, respectively (as seen in Fig. 3b). Taking $V_a(t)$ as an example, v_s is the initial velocity at the instant time t_s. v_{max} is the maximum velocity at the instant time t_m, which is equal to the maximum feed rate F. Therefore, the variation of velocity is $v_a = v_{\max} - v_s$, and the interpolation time at the acceleration segment is $t_a = t_m - t_s$. Depending on Eq. (1), $V_a(t)$ can be expressed as a quintic Bézier curve $\overrightarrow{C}_a(t)$ and obtained as

$$V_a(t) = \overrightarrow{C}_a(t/t_a) = \sum_{i=0}^{5} B_{i,5}(t/t_a)P_{a,i} \ , \quad t\in[0, t_a] \tag{15}$$

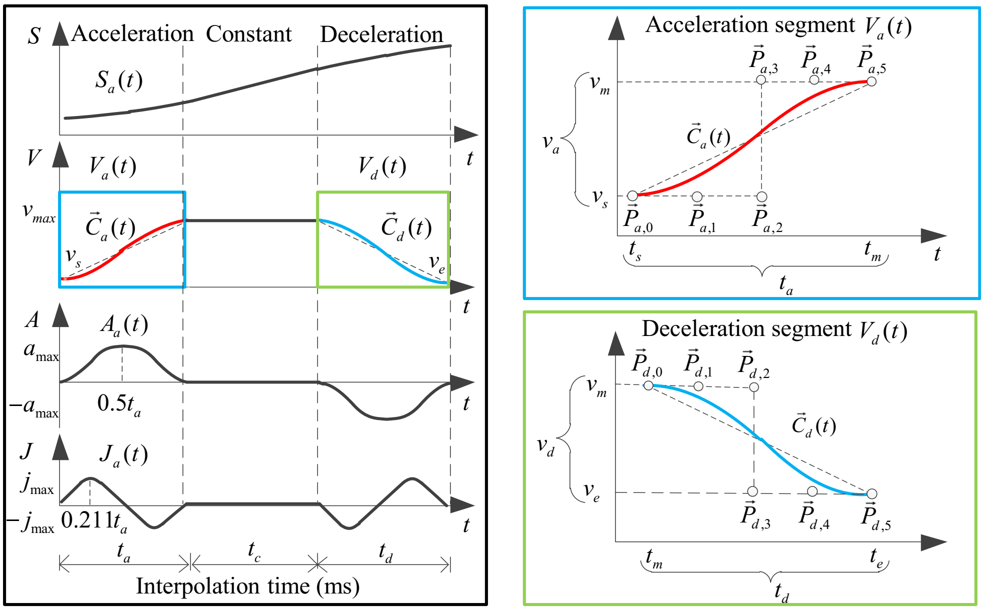
where $\{P_{\text{a,i}}\}$ are the control points and $\{B_{\text{i,5}}\}$ are the Bernstein polynomials. Based on the conclusion in Sect. 2, if the feed rate profile is C^2 continuous, the proposed ACC/DEC scheduling achieves the jerk continuity. Therefore, the first and the second derivative of $V_a(t)$ should be equal to ones of the constant segment at their junctions. After the calculation, $\{P_{\text{a,i}}\}$ are obtained as follows:

$$P_{a,0} = P_{a,1} = P_{a,2} = v_s, \quad P_{a,3} = P_{a,4} = P_{a,5} = v_m \tag{16}$$

Next, through the derivatives of $\overrightarrow{C}_a(t/t_a)$, the acceleration segments of $A(t)$ and $J(t)$ are deduced as

$$\begin{aligned} A_a(t) &= \overrightarrow{C}'_a(t/t_a) \times (t/t_a)' = \sum_{i=0}^{4} B_{i,4}(t/t_a) \times (P_{i+1} - P_i) \times (t/t_a) \\ &= 30 \times \left(t^2/t_a^3\right) \times (1 - t/t_a)^2 \times (v_{\max} - v_s) \end{aligned} \tag{17}$$

Fig. 3 Schematic diagrams of the ACC/DEC scheduling. **a** The displacement profile, the feed rate profile, the acceleration profile, and the jerk profile. **b** The acceleration and deceleration segment of the feed rate profile



(*a*) (*b*)

$$J_a(t) = \vec{C}'_a(t/t_a)\times(t/t_a)'' + \vec{C}''_a(t/t_a)\times(1/t_a)^2 \\ = 60t(t-t_a)(2t-t_a)(v_m-v_s)/t_a^5 \tag{18}$$

where $A_a(t)$ and $J_a(t)$ are the corresponding acceleration segments of $A(t)$ and $J(t)$. Similarly, $S_a(t)$ is achieved as

$$S_a(t) = \int_0^t \vec{C}_a(t/t_a)dt \\ = \left(v_s t + 5t^4/2t_a^3 - 3t^5/t_a^4 + t^6/t_a^5\right)(v_{max}-v_s) \tag{19}$$

where $S_a(t)$ is the corresponding acceleration segments of $S(t)$. When t is equal to $0.5t_a$, the maximum acceleration a_{max} can be determined as $1.875v_a/t_a$. When t is $0.211t_a$, the maximum jerk j_{max} is obtained as $10v_a/\sqrt{3}t_a$. Therefore, t_a is determined as

$$t_a = max\left(\frac{1.875v_a}{a_{max}}, \sqrt{\frac{5.733v_a}{j_{max}}}\right) \tag{20}$$

Since $V_d(t)$ illustrated by the blue line in Fig. 3b is similar with $V_a(t)$, it is not deduced here. At last, the ACC/DEC scheduling based on Bézier curve has been obtained, and there are 20 types of feed rate profiles in all. Based on the proposed scheduling, all ASUs' feed rate profiles can be generated. In NC machining, these profiles could not be generated simultaneously due to computation limitation of NC system. Therefore, to generate feed rate profiles step by step, the BSFR algorithm is put forward in Sect. 3.3.

3.3 Look-ahead ACC/DEC scheduling module

Here, a floating window where a specified number of line segments are covered is employed. In the window, the transition module first works to generate a smooth tool path. Then, the path is divided into a series of ASUs. To generate the corresponding feed rate profiles in real time and to satisfy the constraints of Eq. (14), the BSFR algorithm is proposed with two steps:

Step 1. Backward scanning.

Backward scanning is performed from the end of the current window. For example, as seen in Fig. 5a, $\mathrm{ASU_n}$ is the last ASU in the Window 0, whose length is s_n. $v_{n,s}$ is the theoretical velocity at the end point of $\mathrm{ASU_n}$, which is set to zero to ensure the tool can stop at the end of the Window 0. $v_{n,e}$ is the theoretical velocity at the start point of $\mathrm{ASU_n}$, which is determined via Eq. (14). With these theoretical velocities, the ACC/DEC scheduling is applied for $\mathrm{ASU_n}$ in the backward direction (i.e., from the end point to the start point), which is expected to generate the feed rate profile from $v_{n,s}$ to $v_{n,e}$ (as seen in Frame A in Fig. 5b). In the scheduling, the corresponding scheduled velocity at the start point $v_{n,ce}$ can be determined via Eq. (21). Sometimes, $v_{n,ce}$ may be possibly not equal to $v_{n,e}$. Under this case, the value of $v_{n,e}$ is replaced by that of $v_{n,ce}$, and the feed rate profile of $\mathrm{ASU_n}$ is obtained. After that, the scheduling is implemented again for the $\mathrm{ASU_{n-1}}$ in the backward direction. Define $v_{n-1,s}$ and $v_{n-1,e}$ as the theoretical velocities at the end point and start point of $\mathrm{ASU_{n-1}}$, respectively. Note that the value of $v_{n,e}$ is assigned to that of $v_{n-1,s}$, and $v_{n-1,e}$ is also obtained by Eq. (14). After the scheduling is finished, the feed rate profile of $\mathrm{ASU_{n-1}}$ is generated. In this way, the scheduling is carried out from $\mathrm{ASU_n}$ to $\mathrm{ASU_1}$ recursively (as seen in Fig. 5a). At last, a reverse feed rate profile, depicted as the red dotted line in Frame A in Fig. 5b, is acquired. The flow chart of back scanning is illustrated in Fig. 4a, where s_i and t_i denote the length and the interpolation time of $\mathrm{ASU_i}$ ($i = 0,1,\cdots,n$) respectively.

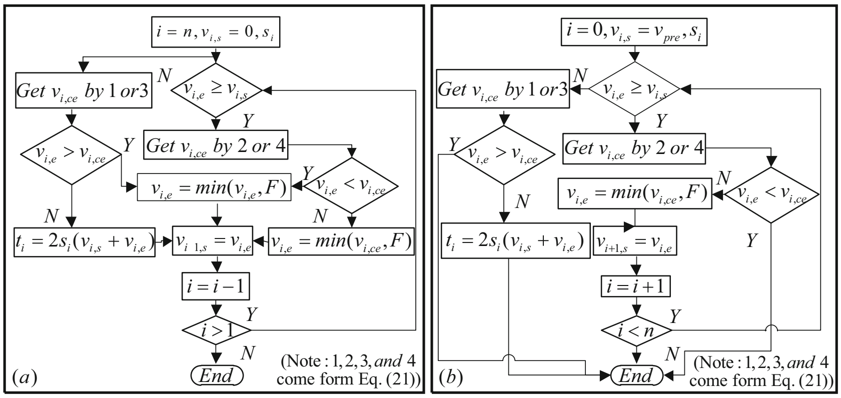


Fig. 4 The flow charts of BSFR algorithm. **a** The backward scanning. **b** The forward revision

Step 2. Forward revision.

Once Step 1 is completed, forward revision is implemented from the start of the current window. For example, as seen in Fig. 5a, ASU$_0$ is the first ASU in the Window 0. v_{pre} denotes the ending velocity of the previous window, which is passed to Window 0 as the value of $v_{0,s}$, i.e., the theoretical velocity at the start point of ASU$_0$. In addition, the theoretical velocity at the end point of ASU$_0$, named as $v_{0,e}$, is acquired from Step 1, which is equal to $v_{1,e}$ (as illustrated in Frame B in Fig. 5b). With these theoretical velocities, the ACC/DEC scheduling is carried out for ASU$_0$ in the forward direction (i.e., from the start point to the end point). In the scheduling, if $v_{0,ce}$, the corresponding scheduled velocity at the end point of ASU$_0$, is not equal to the value of $v_{0,e}$, the value of $v_{0,e}$ is replaced by that of $v_{0,ce}$. In this way, the scheduling is performed recursively from ASU$_0$ to the ASU whose scheduled velocity at the end point is equal to the corresponding theoretical velocity acquired from Step 1. At last, a forward feed rate profile, depicted as the blue solid line in Frame B in Fig. 5b, is generated. The flow chart of forward revision can be seen in Fig. 4b. By integrating the forward feed rate profile with the reverse feed rate profile, the actual feed rate profile in the Window 0 is obtained.

Based on the above-mentioned analysis, the core of the BSFR algorithm is as follows: the theoretical velocity at the end point of each ASU is first determined by Eq. (14). Then, it may be replaced by the corresponding scheduled velocity which is obtained by Eq. (21).

$$\begin{cases} 1: v_{i,ce} = \sqrt{v_{i,s}^2 - \dfrac{16 a_{max} s_i}{15}} & , \ if\ \dfrac{1.875}{a_{max}} \le \sqrt{\dfrac{5.7335}{j_{max}}} \\ 2: v_{i,ce} = \sqrt{\dfrac{16 a_{max} s_i}{15} + v_{i,s}^2} & , \ if\ \dfrac{1.875}{a_{max}} \le \sqrt{\dfrac{5.7335}{j_{max}}} \\ 3: \sqrt{\dfrac{10(v_{i,s} - v_{i,ce})}{\sqrt{3} j_{max}}}\left(\dfrac{v_{i,s}+v_{i,ce}}{2}\right) - s_i = 0, if\ \dfrac{1.875}{a_{max}} > \sqrt{\dfrac{5.7335}{j_{max}}} \\ 4: \sqrt{\dfrac{10(v_{i,ce} - v_{i,s})}{\sqrt{3} j_{max}}}\left(\dfrac{v_{i,s}+v_{i,ce}}{2}\right) - s_i = 0, if\ \dfrac{1.875}{a_{max}} > \sqrt{\dfrac{5.7335}{j_{max}}} \end{cases} \tag{21}$$

where S_{i} represents the length of ASU$_{\text{i}}$. $v_{\text{i,s}}$ is the theoretical velocity at the end point of ASU$_{\text{i}}$, and $v_{\text{i,e}}$ is that at the start point of ASU$_{\text{i}}$. By using Eq. (21), $v_{\text{i,ce}}$, named as the corresponding scheduled velocity at the start of ASU$_{\text{i}}$, can be calculated, which is used to modify the value of $v_{\text{i,e}}$. In this way, the feed rate profile of ASU$_{\text{i}}$ is obtained with the modified value of $v_{\text{i,e}}$.

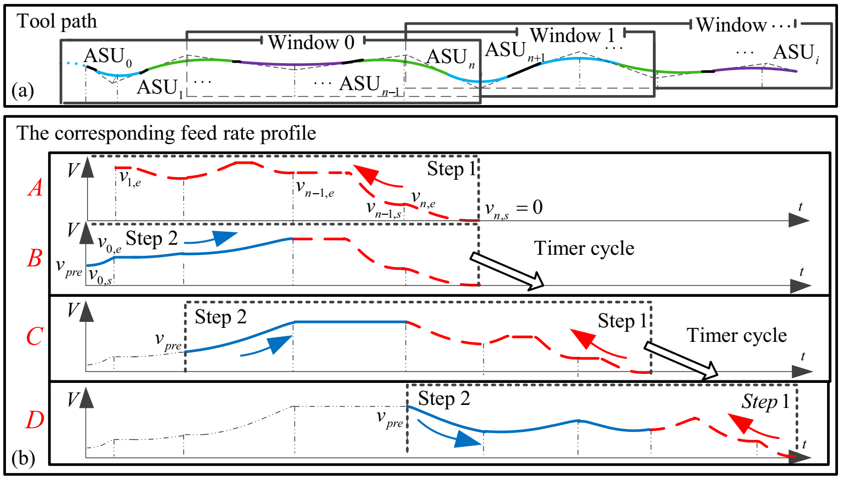


Fig. 5 Schematic diagrams of the BSFR algorithm. **a** Smooth tool path. **b** The corresponding feed rate profile

The proposed method can be applied in most of cases. However, a special case should be paid attention, in which the feed rate profile is generated without the modified value of $v_{\mathrm{i,e}}$. Equation (22) denotes this case. Define $\mathrm{ASU_m}$ as one of ASUs and s_{m} as the length of $\mathrm{ASU_m}$. When s_{m} is too short and the angle between adjacent ASUs is too small, $v_{\mathrm{m,ce}}$ is hardly to slow down to $v_{m,\mathrm{e}}$. That is to say, $\mathrm{ASU_m}$ is a steep region of the tool path. If the tool moves to this region, it is difficult to decrease its velocity to $v_{\mathrm{m,e}}$. To ensure the machining quality of $\mathrm{ASU_m}$, the value of $v_{\mathrm{m,e}}$ will not be replaced by that of $v_{\mathrm{m,ce}}$. Meanwhile, only deceleration segment exists in the feed rate profile, and the interpolation time at the deceleration segment t_d is

$$t_d = \frac{2s_m}{v_{m,s} + v_{m,e}} \tag{22}$$

Once the feed rate profile in current window is determined, other line segments are covered into Window 1 and the BSFR algorithm is implemented again until all line segments are processed (as seen in Fig. 5a and Frame C in Fig. 5b). Finally, with the combination of the proposed ACC/DEC scheduling and the proposed BSFR algorithm, the feed rate profile of tool path, as seen in Frame D, can be generated in real time. This combination mechanism is named as the Look-ahead ACC/DEC scheduling module.

4 Interpolation module and real-time interpolation strategy

In this section, a real-time interpolation strategy is proposed, which can generate a series of interpolation points in real time and utilize the points to guide the tool movement.

4.1 Interpolation module

Based on the transition module and the Look-ahead ACC/DEC scheduling module, a smooth tool path and the corresponding feed rate profile are generated, respectively. Then, according to [9], the interpolation point $B(t)$ can be quickly determined as follows: For instance, $B(t_{\mathrm{k}})$ is the interpolation point at the time t_{k}, and $B(t_{\mathrm{k+1}})$ is the interpolation point at the time $t_{\mathrm{k+1}}$, where $t_{\mathrm{k+1}} = t_{\mathrm{k}} + T$ and T is the interpolation period. If the $B(t_{\mathrm{k+1}})$ is on a line segment and $B(t_{\mathrm{k}})$ is still on the line segment, the line interpolator is applied

$$B(t_{k+1}) = B(t_k) + (V(t_k)T)\overrightarrow{T}_i \tag{23}$$

where $\overrightarrow{T}_i$ is the direction vector of line segment $\overrightarrow{L}_i$ and $V(t_{\mathrm{k}})$ denotes the velocity at the time t_{k}. Similarly, if both $B(t_{\mathrm{k}})$ and $B(t_{\mathrm{k+1}})$ are located on the same curve $\overrightarrow{C}(u)$, the parametric interpolator based on the Taylor's expansion algorithm is adopted

$$u_{k+1} = u_k + \frac{V(t_k)T}{\left\|\overrightarrow{C}'(u)\right\|_{u=u_k}} + \frac{1}{2}\left(\frac{a_kT^2}{\left\|\overrightarrow{C}'(u)\right\|_{u=u_k}} - (V(t_k)T)^2\frac{\left.\left(\overrightarrow{C}'(u)\cdot\overrightarrow{C}''(u)\right)\right|_{u=u_k}}{\left\|\overrightarrow{C}'(u)\right\|^4_{u=u_k}}\right) \tag{24}$$

where u_{k} is the curve parameter with respect to the time t_{k}. α_{k} denotes the acceleration value at the time t_{k}. $\overrightarrow{C}'(u)$ and $\overrightarrow{C}''(u)$ are the first and the second derivative of $\overrightarrow{C}(u)$, respectively. $\overrightarrow{C}'(u)\cdot\overrightarrow{C}''(u)$ represents the dot product between $\overrightarrow{C}'(u)$ and $\overrightarrow{C}''(u)$. The Euclidean norm in 3D space is expressed as $\|\cdot\|$. Substituting the parameter $u_{\mathrm{k+1}}$ into Eq. (1), the point $\overrightarrow{C}(u_{k+1})$ is obtained, which is the interpolation point $B(t_{\mathrm{k+1}})$.

4.2 Real-time interpolation strategy

To build a real-time interpolation strategy, all modules including the transition module, the Look-ahead ACC/DEC scheduling module, and the interpolation module are utilized. Furthermore, a timer is utilized to periodically activate an interrupt service routine (ISR) and to make the proposed modules executed in ISR. Define ISR period (i.e., timer cycle) as T_{t} and the interpolation period as T. Let T_{t} be equal to $m \times T$. Therefore, the interpolation strategy generates a total number of m interpolation points in each timer cycle. The flow charts of the strategy are shown in Fig. 6.

1. In the first timer cycle (expressed as I_0 in Fig. 6b), NC system puts the specified number of line segments into the window (Window 0 in Fig. 6a). Then, the timer module activates ISR and enables other modules in ISR. First, the transition module smoothes these line segments and generates a smooth tool path (the solid blue lines connected with the dotted red lines in Window 0). Second, the path is divided into ASUs. Then, the Look-ahead ACC/DEC scheduling module calculates feed rate profiles. At last, based on ASUs and feed

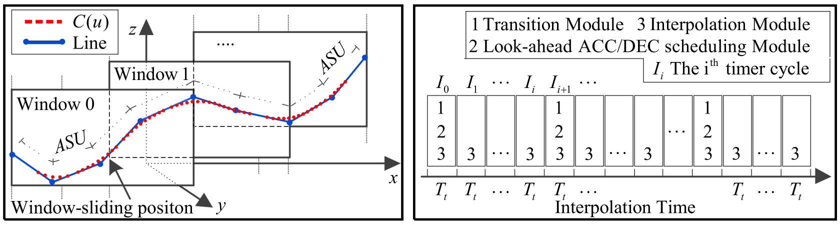

C(u)
Line
z
Window 1
....
ASU
Window 0
ASU
x
Window-sliding positon
y
1 Transition Module 3 Interpolation Module
2 Look-ahead ACC/DEC scheduling Module
I_i The i^th timer cycle
I_0 I_1 ⋯ I_i I_{i+1} ⋯
1 2 3
3
⋯
3
1 2 3
3
⋯
3
⋯
1 2 3
3
⋯
3
T_t T_t ⋯ T_t T_t ⋯ T_t ⋯ T_t
Interpolation Time

Fig. 6 The flow charts of interpolation strategy. **a** Smooth tool path. **b** Program of real-time interpolation

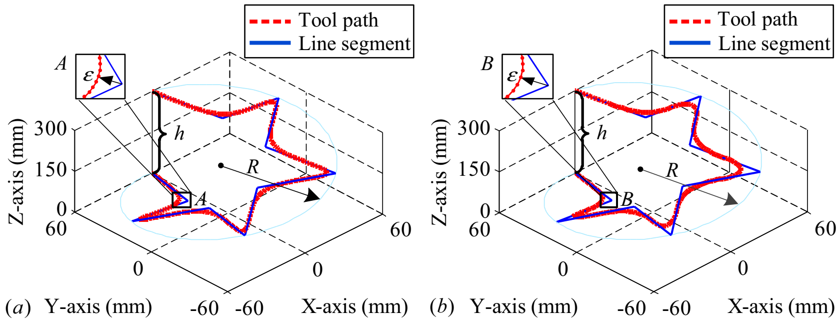


Fig. 7 The line segments and curves of the tool paths. **a** The tool path with $\varepsilon = 5$ mm. **b** The smooth tool path with $\varepsilon = 50$ *mm*

rate profiles, the interpolation module generates m interpolation points and utilizes the points to guide the tool movement along the path.

2. In the next timer cycle, NC system exams whether the tool has reached the window-sliding position (the position where the Window 0 and Windows 1 overlap). If not reached, only the interpolation module works in this ISR (I_1 in Fig. 6b). Accordingly, new interpolation points, with the same number as before, are generated and make the tool continue to move. Otherwise, the window slides forward (Window 1 in Fig. 6a). That is to say, the line segments in front of the window-sliding position are removed from the window, and the same number of new line segments is added into the window. Then, all modules are involved.

5 Real-time interpolation experiment

In this section, the real-time interpolation experiment is performed to verify the proposed strategy. The hardware of experiment is a PC configured with Intel® Core i5 3.2 GHz CPU and Kingston® DDR 3 1600 MHz 4G SDRAM. Meanwhile, the proposed real-time interpolation strategy is coded in Visual Studio 2010, and all modules are packed into C++ class. Furthermore, the multimedia timer is utilized to generate ISR periodically.

5.1 Interpolation task for 3D pentagrams

Two instances of *3D* pentagrams consisting of ten blue solid lines are displayed in Fig. 7a, b, respectively. The radius of the circumscribed circle is $R = 60$ *mm*, and the height of the pentagram is $h = 300$ *mm*. The approximation error ε, shown in Frames A and B, are chosen as 5 and 50 mm, respectively. Correspondingly, two tool paths depicted as the red dotted plot are generated by the transition module. Figure 8a, b shows that the second derivatives of two tool paths are both continuous (namely, C^2 continuity). Next, the real-time interpolation task will be implemented for these *3D* pentagrams, and interpolation parameters are set in Table 1.

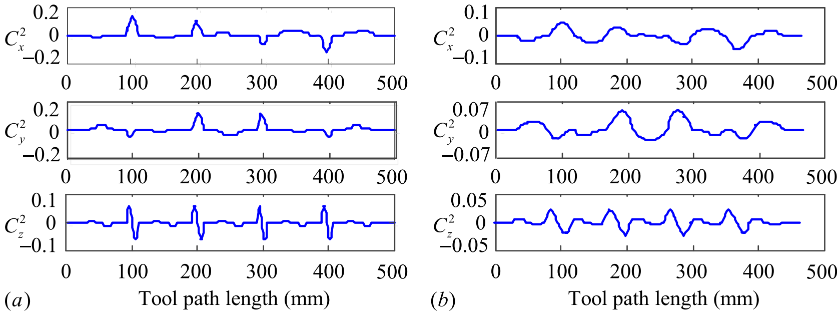


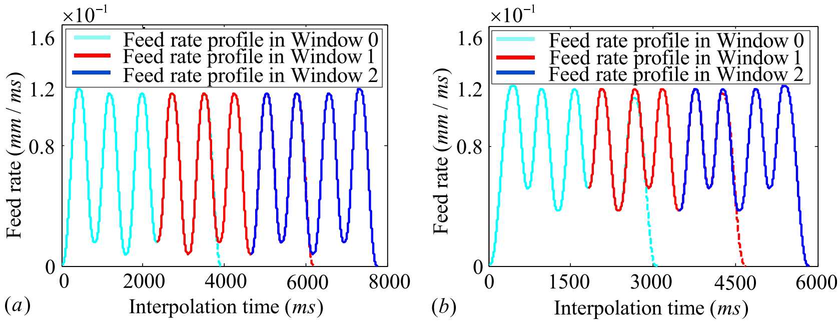
Fig. 8 The properties of the tool paths. **a** Second derivatives of tool paths with $\varepsilon = 5$ mm. **b** second derivatives of tool paths with $\varepsilon = 50$ mm

Table 1 Parameters of the real-time interpolation task

Parameters Graphics	L_{tol} (piece)	L_{win} (piece)	L_{ove} (piece)	ε (mm)	δ (mm)	F (mm/ms)	$a_{\max}$ (mm/ms^2)	$j_{\max}$ (mm/ms^3)	T_{t} (ms)	T (ms)
3D pentagram	10	5	2	5	0.000001	0.12	0.0005	0.0005	10	1
3D pentagram	10	5	2	50	0.00001	0.12	0.0005	0.0005	10	1
2D Butterfly	500	5	2	1	0.000001	0.05	0.001	0.001	10	1

L_{tol} total number of line segments, L_{win} the number of line segments in each window, L_{ove} the number of line segments shared by two adjacent windows, ε approximation error, δ chord error, F maximum feed rate, a_{max} maximum acceleration, j_{max} maximum jerk, T_t ISR period (timer cycle), T interpolation period

Fig. 9 The feed rate profiles generated by the proposed Look-ahead ACC/DEC scheduling. **a** The corresponding tool path with $\varepsilon = 5$ mm. **b** The corresponding tool path with $\varepsilon = 50$ mm



The main procedure to interpolate the *3D* pentagrams is presented as follows: Since L_{win} and L_{ove} are equal to 5 and 2 respectively, three windows are needed. These windows are named as Window 0, Window 1, and Window 2, respectively. Taking Window 0 as an example, when five line segments, expressed as L_0–L_4, are imported into the Window 0 at the first ISR, the transition module is first activated. Accordingly, five ASUs, denoted as ASU$_0$–ASU$_4$, are generated. Then, the corresponding feed rate profiles are calculated through the Look-ahead ACC/DEC scheduling module, which are depicted as green line in Fig. 9a, b. At the end of the green line, the value is set to 0. This means that the Look-ahead ACC/DEC scheduling module can make the velocity reduce to zero at the end of the ASU$_4$. Next, the interpolation module is enabled, and the interpolation task is carried out only from ASU$_0$ to ASU$_2$ through several ISRs. When the interpolation task for ASU$_2$ is completed, another three line segments, expressed as L_5–L_7, are covered into Window 1. Note that line segments in the Window 1 are L_3–L_7. Similarly, ASU$_3$–ASU$_7$ are generated, and their corresponding feed rate profiles, indicated by the red line in Fig. 9a, b, are generated. After that, the interpolation task is executed again from ASU$_3$ to ASU$_5$ through several ISRs. In this way, the interpolation task is implemented until the last line segment L_9 is processed.

The corresponding feed rate profiles generated by the proposed Look-ahead scheduling module are expressed as solid blue lines in Fig. 10a, b. Meanwhile, red dotted lines denote

Fig. 10 The feed rate profiles generated by the ACC/DEC scheduling with/without Look-ahead planning. **a** The corresponding tool path with $\varepsilon =$ 5 mm. **b** The corresponding tool path with $\varepsilon = 50$ mm

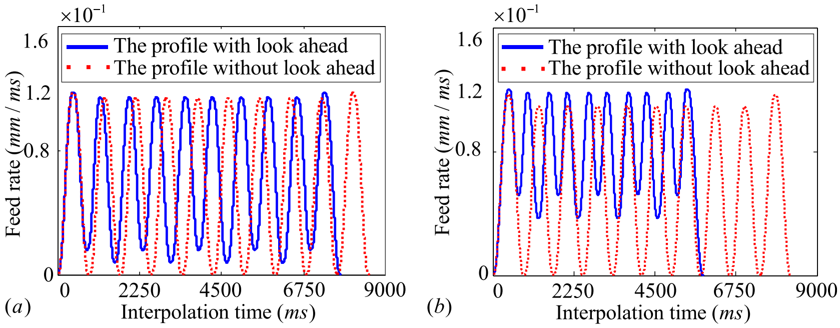
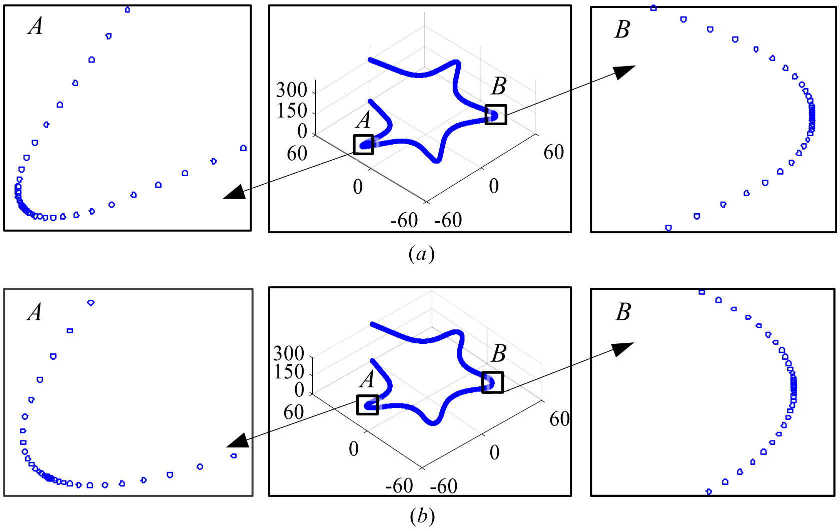


Fig. 11 Interpolation points generated by the proposed strategy. **a** Interpolation task for tool path with $\varepsilon = 5$ mm. **b** Interpolation task for tool path with $\varepsilon = 50$ mm



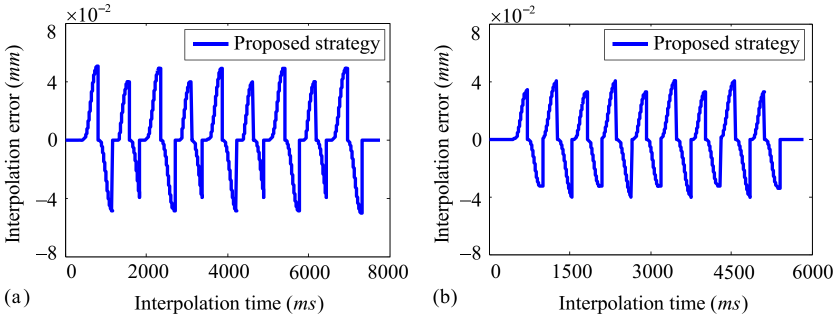


Fig. 12 Interpolation errors with the proposed strategy. **a** Interpolation task for tool path with $\varepsilon = 5$ mm. **b** Interpolation task for tool path with $\varepsilon = 50$ mm

the feed rate profiles generated by the ACC/DEC scheduling without integrating the Look-ahead planning. Compared the red lines with the blue ones, it is concluded that the interpolation time can be efficiently reduced by the proposed Look-ahead ACC/DEC scheduling module. Besides, the frequent start-stop of tool can also be avoided.

After the real-time interpolation task for the two tool paths, a total number of 777 and 585 ISRs are produced, respectively. At last, 7773 and 5853 interpolation points, seen in Fig. 11a, b, are generated in real time. Since the points are too many to illustrate in Fig. 11, the points are sampled every 16 points and drawn in Frames A and B. In the two frames, the tool is first decelerated. When having reached the minimum velocity at the curve's midpoint, the tool is subsequently accelerated. Thus, interpolation errors are within 10^{-2} mm, which can be seen in Fig. 12a, b.

As shown in Table 1, 10 interpolation points should be calculated in each ISR, so the execution time of each ISR should be less than the ISR period T_t and as short as possible. Figure 13 shows the execution time of the proposed strategy. From Fig. 13, it can be observed that all execution time is less than T_t (10 *ms*). Meanwhile, most ISRs have less computation since only the interpolation module works. However, three ISRs especially the first ISR have large computation (arrows in the Fig. 13). In the three ISRs, new line segments are converted in the window. Accordingly, the transition module, the Look-ahead ACC/DEC scheduling module, and the interpolation module are utilized.

Meanwhile, Zhao's strategy [9] and Shi's strategy [10] are used to evaluate the performance of the proposed real-time interpolation strategy. Their execution times are also illustrated in Fig. 13. When the interpolation module is only implemented in the ISR, the execution times of these strategies have no significant differences. This is reasonable since all strategies employ the Taylor's expansion algorithm as the core of the interpolation module. However, if all modules are implemented in the ISR, the execution time of the proposed strategy is obviously shorter than that of other strategies (discussed in the next section), i.e., the proposed strategy is more computationally efficient than other strategies. In these instances of 3*D* pentagrams, the computational efficiency is only improved in three ISRs (arrows in Fig. 13). Therefore, a more complex example is given to show the efficiency of the proposed strategy.

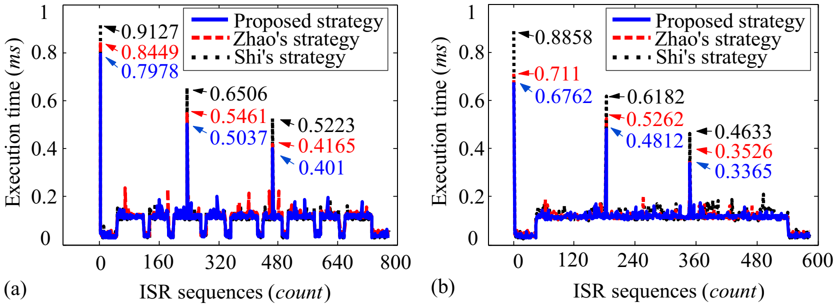


Fig. 13 The execution times with different strategies. **a** interpolation task for tool path with $\varepsilon = 5$ mm. **b** interpolation task for tool path with $\varepsilon = 50$ mm

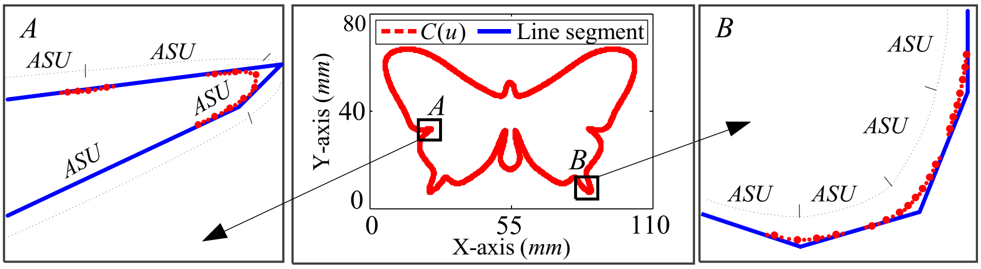


Fig. 14 The line segments and curves of the tool path ($\varepsilon = 1$ mm)

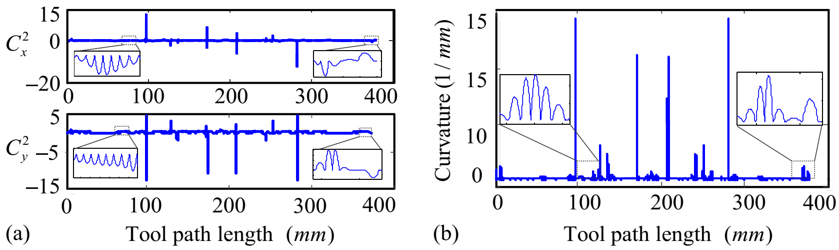


Fig. 15 The properties of the tool path. **a** The second derivatives. **b** The curvature

5.2 Interpolation task for 2D butterfly

A butterfly graphic consists of 500 line segments, which are depicted as blue solid lines in Fig. 14. Through the transition module, a smooth tool path, indicated as the red dotted graphic, is generated. As seen in Fig. 15a, b, the generated tool path achieves both C^2 and G^2 continuity. Similarly, the real-time interpolation task is executed for the butterfly graphic, and interpolation parameters are listed in Table 1.

During the real-time interpolation task, the tool path is divided into 500 ASUs. It is noted that the corresponding feed rate profiles should be generated in real time. As seen in Fig. 16, the red dotted line denotes the feed rate profiles generated by the ACC/DEC scheduling without integrating the Look-ahead planning. It takes 37,469 *ms* to finish the interpolation task. Meanwhile, due to the frequent start-stop of the tool, the machining quality decreases inevitably (Frame B in Fig. 16). On the contrary, when the proposed Look-ahead ACC/DEC scheduling is implemented, the interpolation time has been greatly reduced to 22,701 *ms* (the blue solid line in Fig. 16). In this way, the frequent start-stop of the tool can be avoided (Frame A in Fig. 16).

After the real-time interpolation task, the total number of the generated interpolation points is 22,701. Figure 17 shows the distribution of these points. Since the points are too many to show in Fig. 17, the points are sampled every 2 points and drawn in Frames A and B. In two frames, it can be seen that the tool movement undergoes two different states in each ASU, i.e., a deceleration stage and an acceleration stage. The interpolation errors indicated in Fig. 18a are within 10^{-2} mm.

Similarly, Zhao's strategy and Shi's strategy are also introduced to evaluate the performance of the proposed strategy. Zhao adopted a cubic B-spline curve to smooth line segments, which can generate a G^2 continuous tool path. To achieve the B-spline curve, 5 control points and a knot vector including 9 knots need to be first calculated. Then, a new knot $\overline{u} = 0.5$ is required to insert the original knot vector two times. Correspondingly, control points need to be re-calculated (7 control points in all). Meanwhile, the strategy employs the Look-ahead planning based on bidirectional scanning algorithm [18], which requires the ACC/DEC scheduling twice for each ASU. The red dotted line in Fig. 18b indicates the number of calling the ACC/DEC scheduling in some ISRs where only the ACC/DEC scheduling works. Meanwhile, the corresponding execution time, represented by the red dotted line, can be seen in Fig. 18c. In contrast, the proposed strategy uses the quintic Bézier curve as the transition curve, where only six control points should be calculated. In the

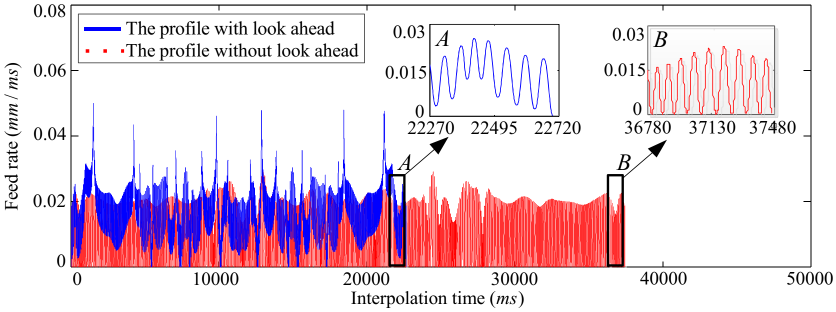


Fig. 16 The feed rate profiles generated by the proposed ACC/DEC scheduling with/without Look-ahead planning

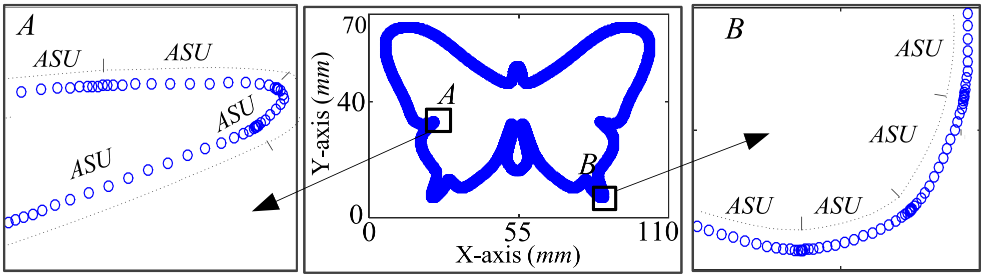


Fig. 17 Interpolation points generated by the proposed strategy

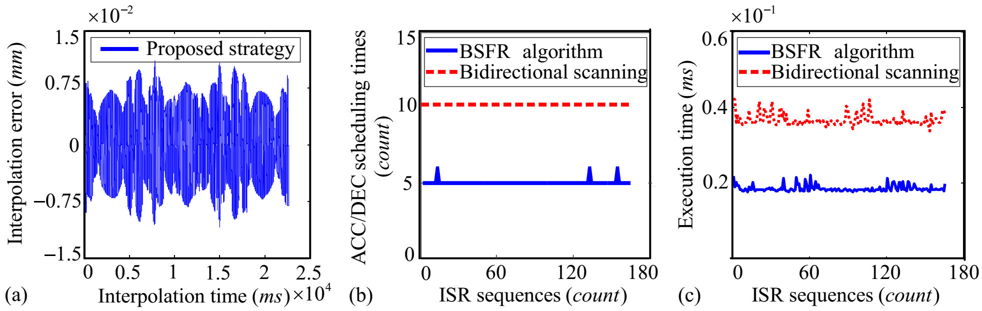


Fig. 18 Efficiency and accuracy of the different strategies. **a** Interpolation errors of the proposed strategy. **b** The numbers of calling the ACC/DEC scheduling. **c** The execution times of Bidirectional scanning algorithm and BSFR algorithm

BSFR algorithm, it is noted that the ACC/DEC scheduling is only implemented one time for most of ASUs, which is described as the blue solid line in Fig. 18b. Accordingly, the execution time of the ACC/DEC scheduling is greatly reduced (the blue solid line in Fig. 18c). Figure 19 shows the efficiencies of two strategies. The execution times of the proposed strategy and Zhao's strategy are expressed by the blue solid line and the red dotted line, respectively. In some ISRs in which only interpolation module works, two strategies have no significant differences. However, in other ISRs where the transition module, the Look-ahead ACC/DEC scheduling module, and the interpolation module are all utilized, the proposed strategy is more efficient than the Zhao's strategy.

Figure 20 displays the efficiencies of the proposed strategy and Shi's strategy. Shi utilized a *PH* curve with 6 control points to transit line segments and generated a G^2 continuous tool path [10]. However, in order to obtain the control points, a local coordinate system {*T-N*} must be established in advance, and a coordinate transformation is needed. Once applied to *3D* space, the generation of the *PH* curve becomes more complicated than that in a *2D* space. On the contrary, the proposed strategy uses a quintic Bézier curve with six control points. First, the strategy uses Eq. (9) to acquire the relationship between ε and R. Then, α can be easily determined from Eq. (6). Next, six control points are quickly achieved. As seen in Fig. 20, the proposed strategy is very efficient, i.e., it only takes about half the execution time of Shi's strategy.

6 Conclusions

This paper proposes a real-time interpolation strategy for line segments, which consists of a transition scheme, a Look-ahead ACC/DEC scheduling, and an interpolation algorithm. The strategy has the following advantages: (1) The relationship among the approximation error, the approximation radius, and the transition curve are deduced in transition scheme. Based on the relationship, the curve can be directly generated when the approximation error is given. (2) In the Look-ahead ACC/DEC scheduling, both an ACC/DEC scheduling based on Bézier curve and a Look-ahead planning based on BSFR algorithm are proposed to eliminate redundant computation. Compared with Zhao's and Shi's strategy, the proposed strategy has the merits of high computational efficiency for interpolation while ensuring both C^2 & G^2 continuity for the tool path and jerk continuity for the tool movement. Interpolation

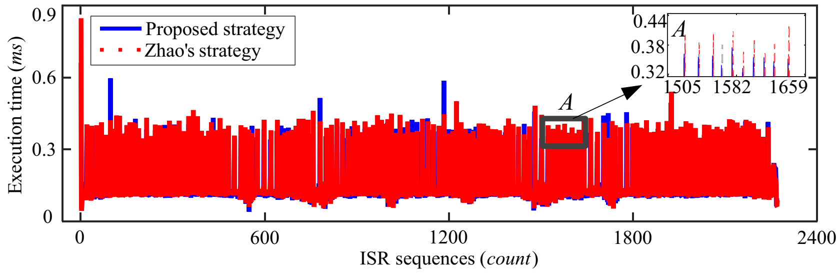


Fig. 19 The execution times of the proposed strategy and Zhao's strategy

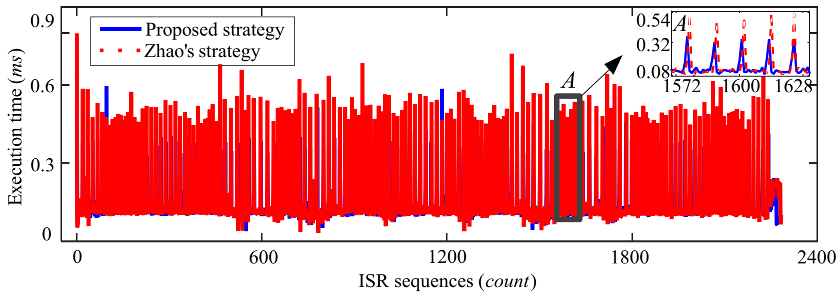


Fig. 20 The execution times of the proposed strategy and Shi's strategy

experiments are implemented, and the results verify the efficiency and reliability of the proposed strategy.

Funding information This work was supported, in part, by the National Key Basic Research Program of China (grant no. 2013CB035804), National Science and Technology Major Project (grant no. 2015ZX04005006), and State Key Lab of Digital Manufacturing Equipment & Technology (grant no. DMETKF2016002).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Sencer B, Ishizaki K, Shamoto E (2015) A curvature optimal sharp corner smoothing algorithm for high-speed feed motion generation of NC systems along linear tool paths. Int J Adv Manuf Technol 76: 1977–1992
2. Tulsyan S, Altintas Y (2015) Local toolpath smoothing for five-axis machine tools. Int J Mach Tool Manu 96:15–26
3. Han J, Jiang Y, Tian X (2017) A local smoothing interpolation method for short line segments to realize continuous motion of tool axis acceleration. Int J Adv Manuf Technol 5-8:1–14
4. H. Wang, C. Liu, J. Wu, X. Sheng, Z. Xiong (2014) Multi-channel transmission mechanism based on embedded motion control system, in Intelligent Robotics and Applications-International Conference 441–452
5. Bashir U, Abbas M, Ali JM (2013) The G^2 and C^2 rational quadratic trigonometric Bézier curve with two shape parameters with applications. Appl Math Comput 219:10183–10197
6. He J, You Y, Wang H (2008) A Micro-line Transition Algorithm based on Ferguson spline for high speed machining (HSM). Chin J Mech Eng 19:2085–2089
7. Zhang LB, You YP, He J, Yang XF (2011) The transition algorithm based on parametric spline curve for high-speed machining of continuous short line segments. Int J Adv Manuf Technol 52:245–254
8. Q. Bi, Y. Wang, L. Zhu, H. Ding (2011) A practical continuous-curvature Bézier transition algorithm for high-speed machining of linear tool path, in Intelligent Robotics and Applications-International Conference 465–476
9. Zhao H, Zhu LM, Ding H (2013) A real-time look-ahead interpolation methodology with curvature-continuous B-spline transition scheme for CNC machining of short line segments. Int J Mach Tools Manuf 65:88–98
10. Shi J, Bi QZ, Wang YH, Liu G (2014) Development of real-time look-ahead methodology based on quintic PH curve with G^2 continuity for high-speed machining. Appl Mech Mater 464:258–264
11. Shi J, Bi QZ, Zhu LM, Wang YH (2014) Corner rounding of linear five-axis tool path by dual PH curves blending. Int J Mach Tool Manu 88:223–236
12. Z. Dai, X. Sheng, J. Hu, H. Wang, D. Zhang (2015) Design and implementation of Bézier curve trajectory planning in DELTA parallel robots. In Intelligent robotics and applications-international conference, 9245 420–430
13. Dong J, Wang T, Li B (2014) Smooth feedrate planning for continuous short line tool path with contour error constraint. Int J Mach Tool Manu 76:1–12
14. Lee AC, Lin MT, Pan YR (2011) The feedrate scheduling of NURBS interpolator for CNC machine tools. Comput Aided Des 43:612–628
15. Heng M, Erkorkmaz K (2010) Design of a NURBS interpolator with minimal feed fluctuation and continuous feed modulation capability. Int J Mach Tool Manu 50:281–293
16. Huang J, Zhu LM (2016) Feedrate scheduling for interpolation of parametric tool path using the sine series representation of jerk profile. Proc Inst Mech Eng B J Eng Manuf 231:2359–2371. https://doi.org/10.1177/0954405416629588
17. Chen JH, Yeh SS, Sun JT (2011) An S-curve acceleration/deceleration design for CNC machine tools using Quintic Feedrate function. Comput-Aided Des Appl 8:583–592
18. Dong J, Stori JA (2006) A generalized time-optimal bidirectional scan algorithm for constrained feed-rate optimization. J Dyn Syst Meas Control 128:725–739
19. Dong J, Stori JA, Dong J, Stori JA (2007) Optimal feed-rate scheduling for high-speed contouring. J Manuf Sci Eng 129:497–513
20. Mattmüller J, Gisler D (2009) Calculating a near time-optimal jerk-constrained trajectory along a specified smooth path. Int J Adv Manuf Technol 45:1007–1016
21. Piegl L, Tiller W (1997) The NURBS book. Springer, Berlin
22. L. Biagiotti, C. Melchiorri (2008) Trajectory planning for automatic machines and robots. Springer, Berlin, pp. 406–414
23. Mathews JH, Fink KD (2004) Numerical methods using Matlab. Prentice Hall, New Jersey, pp. 280–307
24. Lei WT, Sung MP (2007) Fast real-time NURBS path interpolation for CNC machine tools. Int J Mach Tool Manu 47:1530–1541
25. Pessoles X, Landon Y, Rubio W (2010) Kinematic modelling of a 3-axis NC machine tool in linear and circular interpolation. Int J Adv Manuf Technol 47:639–656