ORIGINAL ARTICLE

A local smoothing interpolation method for short line segments to realize continuous motion of tool axis acceleration

Jiang Han[1] · Yang Jiang[1] · Xiaoqing Tian[1] · Feihsu Chen[2,3] · Chienyu Lu[2] · Lian Xia[1]

Received: 25 May 2017 / Accepted: 24 October 2017 / Published online: 14 November 2017

Abstract In traditional processing, a large number of G01 blocks are adopted to discretize free surface or curve for NC machining. But, the continuity of G01 line segments is only C^0, which may lead to discontinuity of axis acceleration, resulting in the frequent fluctuation of tool motion at the junctions in high-speed machining, deteriorating the quality of work piece, and reducing processing efficiency. To solve this problem, a local smoothing interpolation method is proposed in this paper. At first, the analytic relationship between the continuity of the trajectory and the continuity of the axes motion is first systematically described by formula. Based on this relationship, a local smoothing algorithm and a feed-rate scheduling method are proposed to generate a C^2 continuous tool path motion with axis-acceleration continuity. The local smoothing algorithm smoothes the corners of G01 blocks by the cubic B-spline according to the cornering error tolerance specified by the user. After the feed rate at critical points of smoothed tool path was determined by a modified bidirectional scanning algorithm by considering constrains of chord error and kinematic property, an iterative S-shape feed rate scheduling is employed to minimize residual distance caused by round of time while ensuring the continuity of feed rate and acceleration. Then, a look-ahead interpolation strategy combined with smoothing algorithm and feed-rate scheduling as mentioned is proposed for real-time interpolation of short line segments. At last, simulations are conducted to verify the effectiveness of the proposed methods. Compared with the traditional G01 interpolation, it can significantly improve the processing efficiency and shorten the processing time within error tolerance.

Keywords Local smoothing · C^2 continuity · B-spline · Bidirectional scanning · Look-ahead interpolation

1 Introduction

As the development of free form surfaces machining technology, traditional linear (G01) and circular arc (G02/G03) interpolation cannot meet the requirements of high-speed and high-accuracy anymore [1]. But, most of the post-processing software are still in the form of a huge number of small linear segments to approximate complex surfaces, and only C^0 continuous can be guaranteed at the junction of blocks. In the high-speed processing, such trajectory may cause the discontinuity of the acceleration, triggering the oscillations of tool and deteriorating the accuracy and surface quality. In order to improve the continuity of the trajectory and realize the high-order continuous interpolation method, a large number of scholars have made relevant research on the smoothing of the traditional G01/G02/G03 trajectory, mainly from the following two different aspects to solve the problem, i.e., global smooth and local corner smooth.

Global smooth is used to fit several G01 points generated by CAM by using a high-order continuous curve. And, the fitting approach can be divided into interpolation method and approximation method [2]. The interpolation method [2–5] constructs a curve that will accurately pass the given G01 points, but it does not need to be accurate most of the time, as long as the approximation error is guaranteed within a

✉ Lian Xia
xialian@hfut.edu.cn

[1] School of Mechanical Engineering, Hefei University of Technology, Hefei 230009, China

[2] Department of Industrial Education and Technology, National Changhua University of Education, Changhua, Taiwan 500

[3] Anhui Hongqing Precision Machine Co., Ltd, Anqing, Anhui, China

given tolerance. The approximation method [6, 7] generally uses the least-square or other optimization method to approximate the given G01 points with a curve that does not strictly pass the points but can act as a compressor with less data to express a large number of points. The main problem of this method is that the error does not have an analytical expression, and real-time interpolation is not easy to achieve.

The global smooth generally needs for continuous judgment, to search breakpoint, if the continuity of original trajectory generated by CAM is poor, it will increase the processing burden, and there is mostly no analytical formula for the approximation error. In order to address these problems, local smooth method was proposed, which rounds the G01 corner at the junctions that is only C^0 consecutive with high order curve. This method requires lower demand to original trajectory continuity and has a good control of smoothness. Different from the global method, the tool path generated by the local method is a blended trajectory consisted of curve and linear segments. As the development of parameter curve, most scholars used the parameter curve to blend the corner. Pateloup et al. [8] and Zhao et al. [9] utilized cubic B-spline to smooth three-axis linear tool path corner. Beudaert et al. [10] developed this method to five-axis cases by utilizing a pair of B-spline curves. Similarly, Tulsyan et al. [11] employed quintic B-spline to smooth five-axis Tool Tip Center (TCP) trajectory, in order to guarantee C^3 continuous at the junctions. However, the high order B-spline needed by Tulsyan et al. [11] unavoidably increased the computation burden. Yutkowitz et al. [12] proposed a three-axis tool path corner-smoothing method utilizing two fourth-order polynomials. Farouki et al. [13] proposed a blending method using seven degree PH curves. Bi et al. [14] adopted a pair of cubic Bezier curves to round the corner. Although the above scholars have proposed methods to improve the continuity of trajectory, they did not explain the relationship between the continuity of trajectory and the continuity of acceleration of each axis. In this paper, such relationship is systematically analyzed by formula. On this basis, a local smooth method and a feed rate profile scheduling algorithm to ensure the acceleration of each axis in the process of trajectory are proposed.

After the trajectory is generated, it is also necessary to schedule the feed rate. Considering different constraints, it can generate different feed rate profiles. Chord error constraint, acceleration constraint, and jerk constraint are usually considered in high-speed machining [15–17]. Feed rate scheduling can be divided into offline scheduling and online scheduling. Offline scheduling [15] can achieve optimal speed profile, but offline scheduling has its shortcomings. After offline scheduling, the parameters in the high-speed machining process are not convenient to adjust. Online scheduling needs to consider the problem of simplifying calculation and try to avoid iterative computation. With the improvement of computer hardware performance, online interpolation is becoming more and more common. Because of the efficiency of bidirectional scanning algorithm [9, 18], it is used to calculate the speed at the critical point in real time. However, the specific bidirectional scanning algorithm needs to be combined with the actual feed rate profile scheduling method. The S-shape feed rate profile has been widely studied for its smoothness with limitation of jerk. Erkorkmaz et al. [19] proposed a seven-phase S-shape feed rate profile. As the seven-phase S-shape feed rate profile is complex, some scholars made a simplification based on it and proposed a five-phase S-shape feed rate profile [4, 20]. However, the acceleration of five-phase profile cannot maintain the maximum value and may increase the final cycle time because of the lack of two uniform acceleration phases. And, the majority of the literature does not mention how to deal with the problem of the residual distance caused by the round of the interpolation cycle. Ignoring this problem will lead the feed rate of the final run results to be discontinuous, causing a sudden change, resulting in vibration. This paper presents a modified bidirectional scanning which was used as preparation of a seven-phase S-shape feed rate scheduling. The acceleration of this scheduling method can reach and maintain the maximum value. On the other hand, residual distance will be compared with the given error limit for iteration scheduling, so that the feed rate profile maintain continuous. Combined with the smoothing algorithm, it can realize smooth motion of tool axis acceleration in the end.

In this paper, a local smoothing interpolation method for short line segments to realize continuous motion of tool axis acceleration is proposed. The rest of this paper is organized as follows: a local tool path smoothing algorithm is presented in Sect. 2. In Sect. 3, combined with feed rate scheduling, a real-time interpolation algorithm is developed. Simulations are conducted in Sect. 4, and the conclusions are given in Sect. 5.

2 Local tool path smoothing scheme

In order to guarantee the continuity of the motion of each axis of machine during processing, it should ensure the C^2 continuity of the trajectory with respect to arc length and the continuity of tangential velocity and acceleration. Please refer to the Appendix section for the proof of this conclusion. Based on this conclusion of relationship between track continuity and movement continuity for each axis, to avoid complex calculation while ensuring the continuity, cubic B-splines are used to generate C^2 continuous tool paths after smoothing. Suppose $\mathbf{C}(u)$ represents an pth-degree B-spline which is defined as follows [21]:

$$\mathbf{C}(u) = \sum_{i=0}^{n} N_{i,p}(u)\mathbf{P}_i, \ \ for\ 0 \le u \le 1 \tag{1}$$

where $\{\mathbf{P}_i\}$ are the control points ($n + 1$ points), and $\{N_{i,p}(u)\}$ are the pth-degree B-spline basis functions defined on the nonperiodic (or non-uniform) knot vector (*m* + *1* knots) **U** with **U** given by

$$\mathbf{U}=\left\{0,\ldots,0,u_{p+1},\ldots.u_{m-p-1},1,\ldots.,1\right\},where\ m$$
$$=n+p+1 \qquad (2)$$

The $\{N_{i,p}(u)\}$ can be calculated by using the recurrence formulas as follows:

$$\begin{gathered}N_{i,0}(u)=\begin{cases}1, if\ u_i\le u<u_{i+1}\\ 0, otherwise\end{cases}\\ N_{i,p}(u)=\frac{u-u_i}{u_{i+p}-u_i}N_{i,p}(u)+\frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1}(u),\\ where\frac{0}{0}=0\end{gathered} \qquad (3)$$

Set $p = 3$, $n = 4$, and $\mathbf{U} = [0, 0, 0, 0, 0.5, 1, 1, 1, 1]$. Then, $\{N_{i,p}(u)\}$ can be calculated according to Eq. (3).The next step is to get the control points $\{\mathbf{P}_i\}$ under a series of limiting condition. Finally, the B-spline **C**(u) can be obtained to smooth the corners. Figure 1 describes a typical corner in the tool path composed of linear segments, i.e., $\overline{\mathbf{Q}_0\mathbf{Q}_1}$ and $\overline{\mathbf{Q}_1\mathbf{Q}_2}$, whose lengths are denoted as L_1 and L_2, respectively. The resulting tool path is composed of two lines: $\overline{\mathbf{Q}_0\mathbf{P}_0}, \overline{\mathbf{P}_4\mathbf{Q}_2}$ and a cubic B-spline with five control points as $\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$, and $\mathbf{P}_4$. In the following, the C^2 continuity constraint and approximation error constraint are constructed as the limiting conditions for the control points $\{\mathbf{P}_i\}$.

2.1 C^2 continuity constraint

Local tool path smoothing results in a curve-line-blending trajectory, and C^2 continuity means that the blended tool path has second-order parametric continuity. The parametric functions of the line can be expressed as

$$\mathbf{L}=\begin{cases}x(t)=x_0+at\\ y(t)=y_0+bt\\ z(t)=c_0+ct\end{cases} \qquad (4)$$

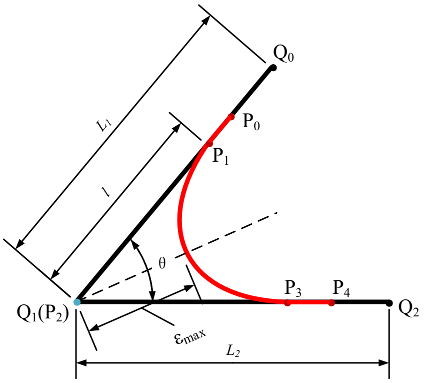


Fig. 1 Local tool path smoothing

while $(a, b, c)^T$ is the vector of line, (x_0, y_0, z_0) is one point on the line, and *t* is the parameter. So, the derivatives of the position vector **P** with respect to the tool path length *s* for a linear tool path profile can be derived:

$$\begin{cases}\mathbf{L}_s=\dfrac{d\mathbf{L}}{ds}=\dfrac{d\mathbf{L}}{dt}\dfrac{dt}{ds}=\dfrac{1}{\sqrt{a^2+b^2+c^2}}\begin{bmatrix}a\\ b\\ c\end{bmatrix}\\ \mathbf{L}_{ss}=\dfrac{d\mathbf{L}^2}{ds^2}=\mathbf{0}\end{cases} \qquad (5)$$

The non-uniform knot vector **U** ensures that the B-spline curve will pass through the first and the last control point $\mathbf{P}_0$ and $\mathbf{P}_4$. In order to achieve C^2 continuity at the junction points $\mathbf{P}_0(u = 0)$ and $\mathbf{P}_4(u = 1)$, the following equation should be satisfied:

$$\begin{cases}\mathbf{C}_s\big|_{u=0,1}=\dfrac{d\mathbf{C}}{ds}=\mathbf{C}_u\bullet u_s=\mathbf{L}_s\\ \mathbf{C}_{ss}\big|_{u=0,1}=\dfrac{d^2\mathbf{C}}{ds^2}=\mathbf{C}_{uu}\bullet {u_s}^2+\mathbf{C}_u\bullet u_{ss}=\mathbf{L}_{ss}=\mathbf{0}\end{cases} \qquad (6)$$

where

$$\begin{cases}u_s=\dfrac{1}{\|\mathbf{C}_u\|}\\ u_{ss}=-\dfrac{\mathbf{C}_u\bullet {\mathbf{C}_{uu}}^T}{\|\mathbf{C}_u\|^4}\end{cases}$$

By substituting $u = 0$ into Eq. (6), the solution leads to the relationships of control points as

$$\frac{\mathbf{P}_1-\mathbf{P}_0}{\|\mathbf{P}_1-\mathbf{P}_0\|}=\frac{\mathbf{Q}_1-\mathbf{Q}_0}{\|\mathbf{Q}_1-\mathbf{Q}_0\|} \qquad (7)$$

$$24\mathbf{P}_0-36\mathbf{P}_1+12\mathbf{P}_2=0\Rightarrow\|\mathbf{P}_0-\mathbf{P}_2\|=\frac{3}{2}\|\mathbf{P}_1-\mathbf{P}_2\| \qquad (8)$$

where $\|\cdot\|$ is the Euclidean norm.

Equation (7) implies that $\mathbf{Q}_0$, $\mathbf{P}_0$, and $\mathbf{Q}_1$ should be collinear, defining the length of line $\overline{\mathbf{P}_1\mathbf{P}_2}$ as *l*. From Eq. (8), the length of line $\overline{\mathbf{P}_0\mathbf{P}_2}$ is evaluated as *1.5l*. To simplify the solution of control point locations, the third control point $\mathbf{P}_2$ is positioned at the corner point $\mathbf{Q}_1$. Since the knot vector is set as $\mathbf{U} = [0, 0, 0, 0, 0.5, 1, 1, 1, 1]$, the B-spline is symmetrical about the angular bisector of the corner angle. For this property, another relationship between $\mathbf{P}_2$, $\mathbf{P}_3$, and $\mathbf{P}_4$ can be

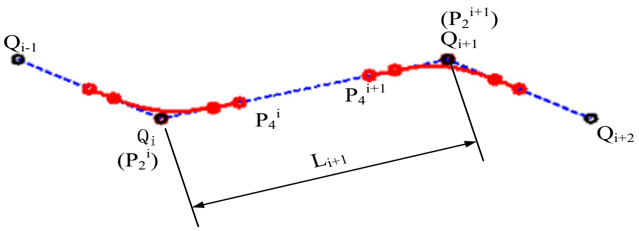


Fig. 2 Example of two corners

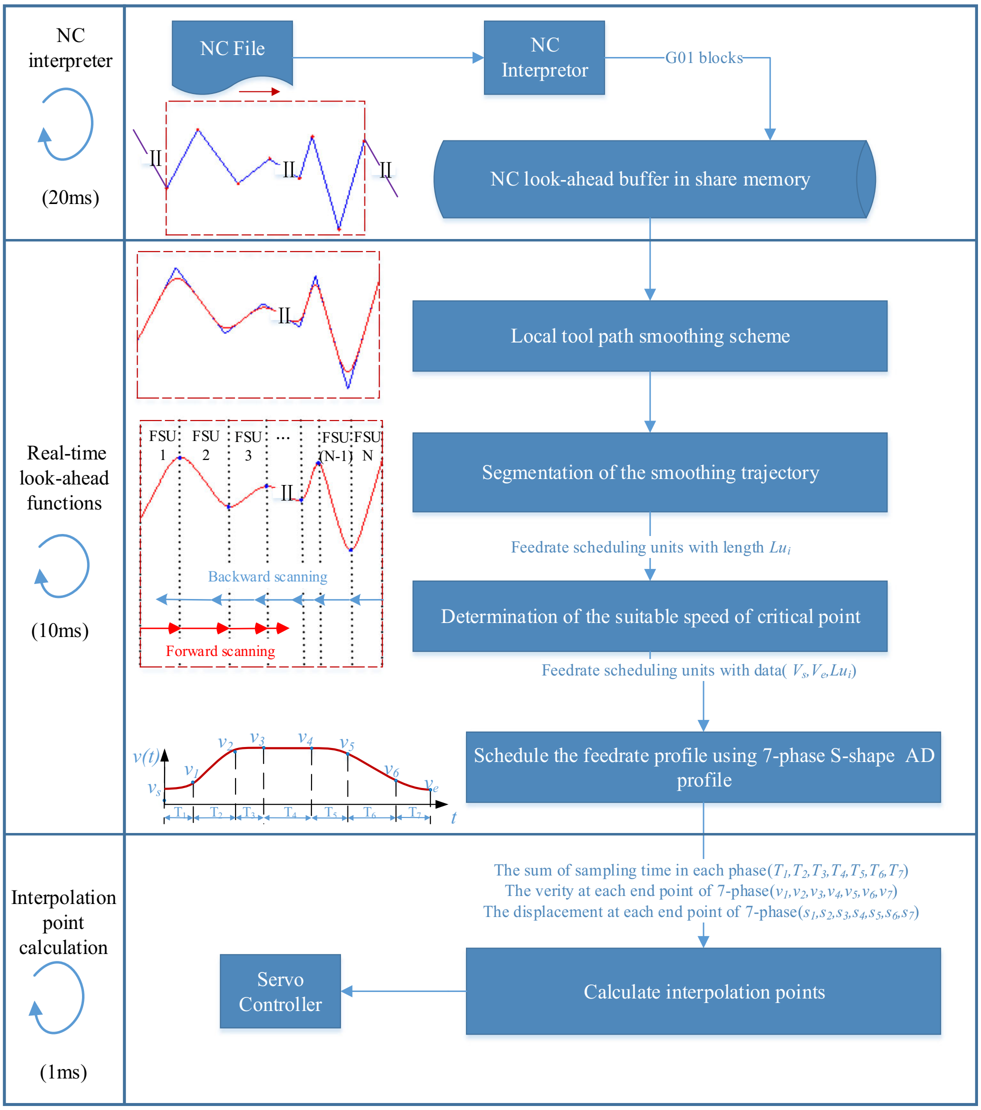

NC interpreter
(20ms)
NC File
NC Interpretor
G01 blocks
II
II
II
NC look-ahead buffer in share memory
Real-time look-ahead functions
(10ms)
II
Local tool path smoothing scheme
FSU 1
FSU 2
FSU 3
...
FSU (N-1)
FSU N
II
Segmentation of the smoothing trajectory
Feedrate scheduling units with length Lu_i
Backward scanning
Forward scanning
Determination of the suitable speed of critical point
Feedrate scheduling units with data(V_s,V_e,Lu_i)
$v(t)$
v_s
v_1
v_2
v_3
v_4
v_5
v_6
v_e
T_1
T_2
T_3
T_4
T_5
T_6
T_7
t
Schedule the feedrate profile using 7-phase S-shape AD profile
Interpolation point calculation
(1ms)
The sum of sampling time in each phase(T_1,T_2,T_3,T_4,T_5,T_6,T_7)
The verity at each end point of 7-phase(v_1,v_2,v_3,v_4,v_5,v_6,v_7)
The displacement at each end point of 7-phase(s_1,s_2,s_3,s_4,s_5,s_6,s_7)
Servo Controller
Calculate interpolation points

Fig. 3 The flowchart of the proposed smoothing interpolation method

easily derived, where the length of line $\overline{\mathbf{P}_3\mathbf{P}_2}$ and $\overline{\mathbf{P}_4\mathbf{P}_3}$ are l and $1.5l$, respectively. Once the length l is determined, the control points $\{\mathbf{P}_i\}$ will be fixed. The length l will be evaluated by the following discussion.

2.2 Approximation error constraint

Since the symmetry of the constructed spline is across the angular bisector of the corner angle, the maximum approximation error (ε_{max}) occurs at the middle point of the spline, and its value is the length from $\mathbf{P}_2$ to $\mathbf{C}(0.5)$. Note the cornering angle as θ. The maximum approximation error is evaluated as

$$\varepsilon_{\max} = \|P_2 - C(0.5)\| = \left\|P_2 - \left(\frac{1}{4}P_1 + \frac{1}{2}P_2 + \frac{1}{4}P_3\right)\right\| \\ = \frac{1}{2} l\cos\left(\frac{\theta}{2}\right) \tag{9}$$

On the other hand, the maximum approximation error ε_{max} should be limited by user-defined tolerance e,

$$\varepsilon_{\max} = \frac{1}{2} l\cos\left(\frac{\theta}{2}\right) \le e \Rightarrow l \le \left(\frac{2e}{\cos\left(\frac{\theta}{2}\right)}\right) \tag{10}$$

Consider that the length of $\overline{\mathbf{P}_0\mathbf{P}_2}$ should be no more than the length of $\overline{\mathbf{Q}_0\mathbf{Q}_1}$. The same happens to $\overline{\mathbf{P}_2\mathbf{P}_4}$ with $\overline{\mathbf{Q}_1\mathbf{Q}_2}$. In general, there are two corners existing in the start and the end of one G1 block, as shown in Fig. 2. So, the sum of lengths $\overline{\mathbf{P}_2^i\mathbf{P}_4^i}$ and $\overline{\mathbf{P}_0^{i+1}\mathbf{P}_2^{i+1}}$ should be less than that of $\overline{\mathbf{Q}_i\mathbf{Q}_{i+1}}$. Finally, the length l should be determined by

$$l = \min\left(\frac{2e}{\cos\left(\frac{\theta}{2}\right)}, \frac{L_1}{6}, \frac{L_2}{6}\right) \tag{11}$$

Then, the arc length Cl_i of ith corner spline $\mathbf{C}_i(u)$ can be evaluated by adaptive Simpson's method [22] offline, which is than stored into buffer for interpolation.

3 Real-time interpolation algorithm

In order to feed position commands to the position controller of the CNC, a real-time interpolation algorithm is proposed in this section. As per the flowchart shown in Fig. 3, there are three main procedures in the whole interpolation algorithm.

In the interpreter, the introduction of the look-ahead window can ensure that the interpolation algorithm can handle trajectory comprising a large number of linear segments. When the tail segment of the window forward direction has completed the interpolation process, the look-ahead window would move forward. Then, the real-time look-ahead functions will be implemented for the trajectory in this new window.

The real-time look-ahead functions are composed of local tool path smoothing, segmentation of the smoothed trajectory, determination of the suitable feed rate at critical point, and the feed-rate scheduling process. At first, the smoothed trajectory by using the scheme in Sect. 2 is segmented into the feed rate scheduling units (FSUs) by the points $\mathbf{C}_i(0.5)$ on the ith corner spline before scheduling the feed rate profile. The length Lu_i of each unit is approximately calculated, which is used as input information for the following steps. Then, the suitable speed of start/end points V_s/V_e of each feed-rate scheduling unit is determined by a modified bidirectional scanning process which does not need to solve the equations. Based on the data (Lu_i, V_s, V_e), a seven-phase S-shape feed rate profile as shown in Fig. 4 is scheduled in each feed rate scheduling unit to get the sum of sampling time T_i in each stage, the feed rate v_i, and the arc-length s_i at the end of each stage.

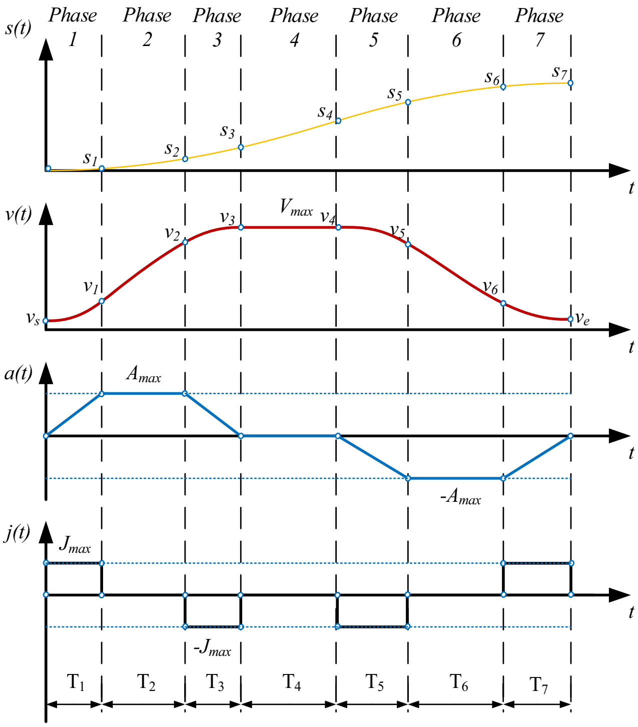


Fig. 4 S-shape feed rate profile with seven phases

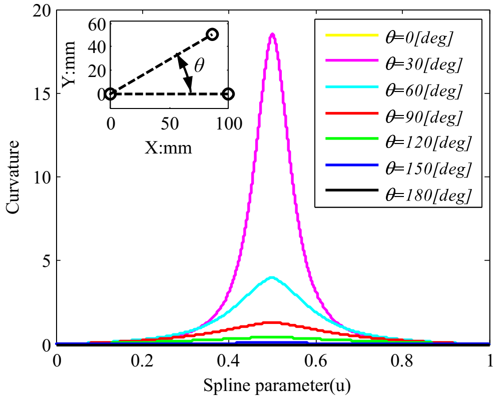


Fig. 5 The curvature profiles of the B-spline sub-segment in FSU

Finally, the arc-length increment Δs at each sampling time is calculated. According to the location of Δs, a linear interpolation or spline interpolation is adopted for computation of the current interpolation point which is used as position command feed to controller.

3.1 Segmentation of the smoothed trajectory

Although the smoothed trajectory in Sect. 2 has C^2 continuity, it still could violate the kinematic property for high-speed machining at critical points which has the local maximum curvature [23]. Considering the similar problem, there is curve segmentation module or curve splitting module in most NURBS interpolator before determining the corresponding

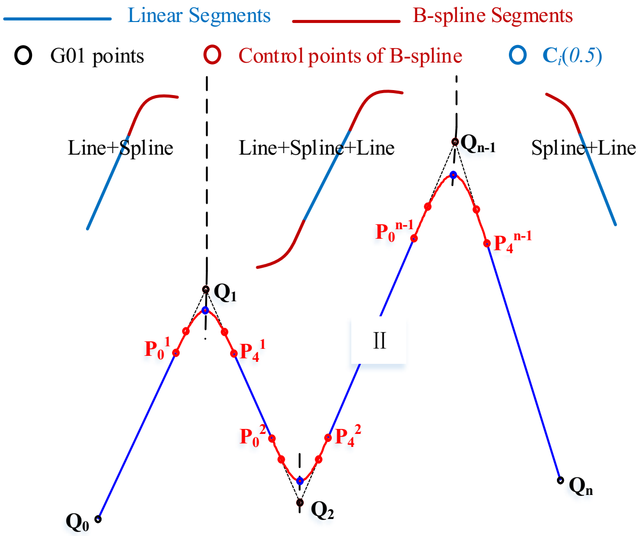


Fig. 6 Segmentation of the smoothed trajectory

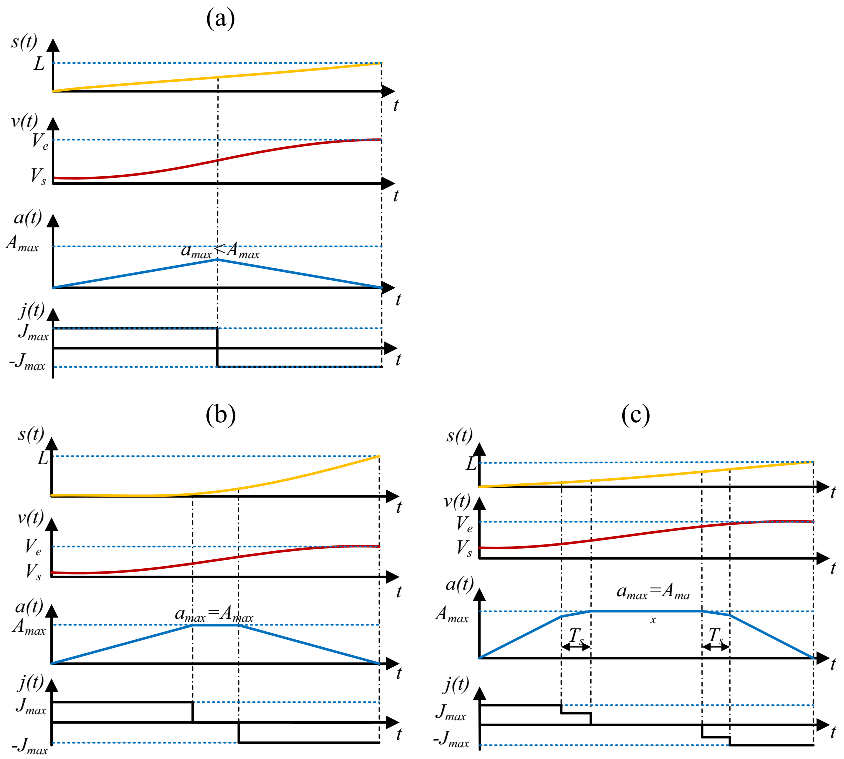


Fig. 7 Discrete S-shape acceleration profiles

feed rates at these critical points with different constraints [23–25]. Likewise, the trajectory resulted in Sect. 2 is firstly segmented into FSUs by the critical points in this paper. And, the determination of the suitable feed rates for the critical points is discussed in the following sub-section.

As mentioned in Sect. 2, the smoothed trajectory is curve-line-blending. The curvature for linear sub-trajectory is always zero. For curve sub-trajectory, the curvature can be calculated by

$$k(u) = \frac{\left\| \mathbf{C}^{'}(u) \times \mathbf{C}^{''}(u) \right\|}{\left\| \mathbf{C}^{'}(u) \right\|} \tag{12}$$

The curvature profiles for some curve sub-trajectories are shown in Fig. 5.The smoothed trajectories result from two G01 blocks with various cornering angles. The length of each block is set as 100 mm. And, the maximum smoothed approximation error ε_{max} is set as 0.1 mm. From Fig. 5, it can be seen that the curve parameter u corresponding to the local maximum curvature is approximately equal to 0.5. Consequently, the points $\mathbf{C}_i(0.5)$ which have the curvature as $k_i(0.5)$ at each concern curve $\mathbf{C}_i(u)$ are taken as the critical points. Then, the trajectory is segmented into the feed rate scheduling units by the points $\mathbf{C}_i(0.5)$ on the ith corner spline. Therefore, there

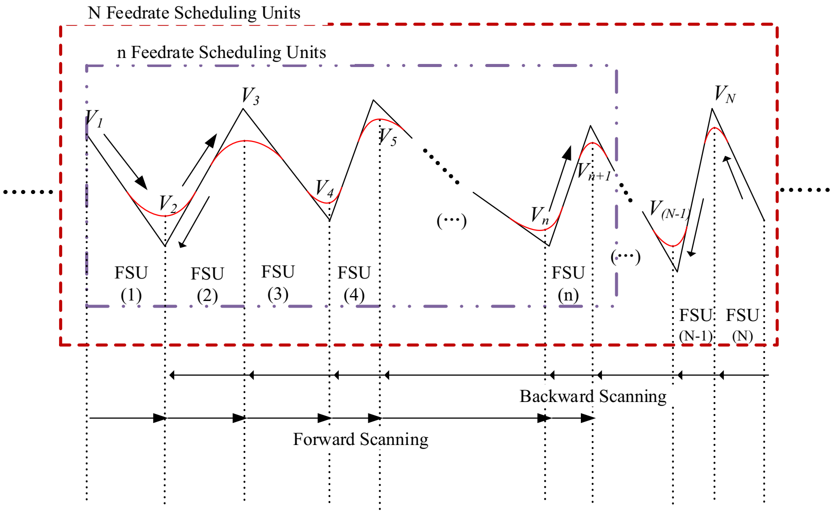


Fig. 8 Bidirectional scanning of FSUs

Table 1 Expressions for the seven-phase S-shape AD profile

τ	j(t)	a(t)	v(t)	s(t)
$\tau_1 \in [0, T_1)$	$J_{\max}$	$J_{\max}\tau_1$	$V_s + \frac{1}{2}J_{\max}{\tau_1}^2$	$V_s t + \frac{1}{6}J_{\max}{\tau_1}^3$
$\tau_2 \in [0, T_2)$	0	a_1	$v_1 + a_1\tau_2$	$s_1 + v_1\tau_2 + \frac{1}{2}a_1{\tau_2}^3$
$\tau_3 \in [0, T_3)$	$-J_{\max}$	$a_2 - J_{\max}\tau_3$	$v_2 + a_2\tau_3 - \frac{1}{2}J_{\max}{\tau_3}^2$	$s_2 + v_2\tau_3 + \frac{1}{2}a_2{\tau_3}^2 - \frac{1}{2}J_{\max}{\tau_3}^3$
$\tau_4 \in [0, T_4)$	0	0	v_3	$s_3 + v_3\tau_4$
$\tau_5 \in [0, T_5)$	$-J_{\max}$	$-J_{\max}\tau_5$	$v_4 - \frac{1}{2}J_{\max}{\tau_5}^2$	$s_4 + v_4\tau_5 - \frac{1}{6}J_{\max}{\tau_5}^3$
$\tau_6 \in [0, T_6)$	0	a_5	$v_5 + a_5\tau_6$	$s_5 + v_5\tau_6 + \frac{1}{2}a_5{\tau_6}^2$
$\tau_7 \in [0, T_7)$	$J_{\max}$	$\mathrm{a}_6 + J_{\max}\tau_7$	$v_6 + a_6\tau_7 + \frac{1}{2}J_{\max}{\tau_7}^2$	$s_6 + v_6\tau_7 + \frac{1}{2}a_6\tau_7 + \frac{1}{2}J_{\max}{\tau_7}^3$

exist n feed rate scheduling units for a smoothed trajectory with n G01 blocks. Each unit is consisted of one linear sub-segment and one or two half of B-spline curve sub-segments, as shown in Fig. 6. The length of ith feed rate scheduling unit Lu_i can be calculated by

$$Lu_i = \begin{cases} Ls_i + Cl_i/2 \quad \mathit{if}\ i = 1 \\ Cl_{(i-1)}/2 + Ls_i + Cl_i/2 \quad \mathit{if}\ 1 < i < n \\ Cl_{(i-1)}/2 + Ls_i \quad \mathit{if}\ i = n \end{cases} \tag{13}$$

where $Cl_i(i = 1, 2...n)$ have been discussed in Sect. 2, and $Ls_i(i = 1,2...n)$ is the length of ith linear sub-segment which is equal to the length of $\overline{\mathbf{Q}_0\mathbf{P}_0^1}$ ($i = 1$) or $\overline{\mathbf{P}_4^i\mathbf{P}_0^{i+1}}$ ($1 < i < n$) or $\overline{\mathbf{P}_4^{n-1}\mathbf{Q}_n}$ ($i = n$). Finally, the values of Lu_i and $k_i(0.5)$ are stored into buffer for the following procedure.

3.2 Determination of the suitable speed of critical point

Before scheduling the feed rate profile of FSU, the suitable feed rates at its start/end points V_s/V_e which are the critical point feed rates should be determined as well. A modified bidirectional scanning as shown in Fig. 8 for seven-phase S-shape feed rate profile based on [25] is proposed for this determination procedure, which does not require solving the equations of V_s/V_e in each FSU. And, the constraints of command feed rate, chord error, and acceleration/jerk limitations including centripetal/tangent one are considered.

3.2.1 Modified backward scanning

Supposing that there exist N feed rate scheduling units in the look-ahead window as shown in Fig. 3 and the number of feed rate scheduling unit in look-ahead window for the real-time interpolation is $n(n \leq N)$, modified backward scanning should be performed from the N-th to the second feed rate scheduling unit. The start point feed rate of the backward scanning which is at the end of N-th feed rate scheduling is set as zero to make sure the tool motion stop at the end of the look-ahead window. To constrain the tangent acceleration/jerk limitations, the discrete S-shape acceleration profiles as shown in Fig. 7 are adopted in these feed rate scheduling units. The algorithm of the computation of final feed rate V_e after the acceleration is given as follows:

Algorithm 3.1. Computation of final feed rate after discrete S-shape acceleration

Input: the initial feed rate V_s, the length for acceleration S_a, the sampling period T_s, the tangent acceleration/jerk limitations A_t/J_t
Output: the final feed rate V_e after acceleration

```
(1)Set $a = 0, v_e = V_s, V_e = V_s, a_{temp} = 0, T_1 = 0, T_2 = 0,$
   $l_{acc} = 0$
(2)$T_i = T_1 + 1, a_{temp} = T_1 \bullet J_t \bullet T_s,$
$\mathit{if}\ a_{temp} \leq A_t,$
    $a = a_{temp}, v_e = V_s + T_1 \bullet a \bullet T_s,$
    $l_{acc} = T_1 \bullet T_s(V_s + v_e),$
$\mathit{else}$
    $T_1 = T_1 - 1, T_2 = T_2 + 1,$
    $\mathit{if}\ (a \neq A_t \& T_2 = 1)$
        $T_2 = T_2 + 1,$
        $v_e = v_e + (a + A_t) \bullet T_s$
    $\mathit{else}\quad v_e = v_e + A_t \bullet T_s$
    $a = A_t,$
    $l_{acc} = \frac{1}{2}T_s(2T_1 + T_2)(V_s + v_e)$
(3)$\mathit{if}\ S_a - l_{acc} > 0, V_e = v_e,$ go $\mathit{to}\ \mathit{step}(2);$
   $\mathit{else}\ \mathit{return}\ V_e$
```

where a, a_{temp}, and l_{acc} are intermediate variables for the procedure.

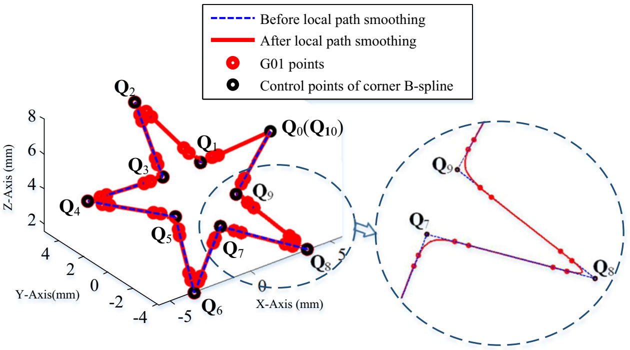


Fig. 9 3D pentagram contour for MATLAB simulation

In the end of acceleration, other constraints of command feed rate V_{max}, chord error δ, and normal acceleration/jerk limitations A_n/J_n are considered to determine the suitable feed rate at the start point of these feed rate scheduling units. The details of the modified backward scanning process are given as follows, where $V'_{(i+1)}$ is the result end point feed rate of ith feed rate scheduling unit for this procedure (Fig. 8).

Step 1: Set $i = N$, and $V'_{(i+1)} = 0$;

Step 2: According to the length of ith feed rate scheduling unit Lu_i, set $S_a = Lu_i$, $\mathrm{V_s} = V'_{(i+1)}$, then implement the algorithm of S-shape acceleration to get the final feed rate V_e, which also considers the chord error, normal acceleration/jerk [26]. The maximum normal acceleration/jerk is supposed to be equal to the tangent ones. The result point feed rate V'_i for *(i-1)*-th feed rate scheduling in backward scanning should be determined by

$$V^{'}_{i}{=}\min\left\{V_e, \frac{2}{T_s}\sqrt{\frac{2\delta}{k_i}-\delta^2}, \sqrt{\frac{A_n}{\max(\mathbf{q}_{ss})}}, \sqrt[3]{\frac{J_n}{\max(\mathbf{q}_{sss})}}, V_{\max}\right\} \tag{14}$$

Table 2 The corner point data of the pentagram contour

Corner point	Data (X, Y, Z) unit mm
Q_0	(5.706339, 1.605699, 4.927051)
Q_1	(1.175571, 1.401259, 4.809017)
Q_2	(0, 5.196152, 7)
Q_3	(− 1.175571, 1.401259, 4.809017)
Q_4	(− 5.706339, 1.605699, 4.927051)
Q_5	(− 1.902113, − 0.5352331, 3.690983)
Q_6	(− 3.526712, − 4.203776, 1.572949)
Q_7	(0, − 1.732051, 3)
Q_8	(3.526712, − 4.203776, 1.572949)
Q_9	(1.902113, − 0.5352331, 3.690983)
Q_{10}	(5.706339, 1.605699, 4.927051)

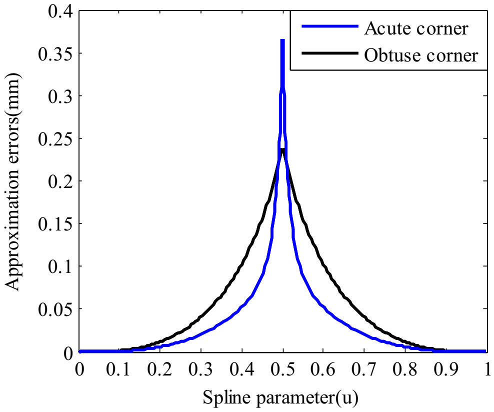


Fig. 10 The actual approximation error profiles of the smoothed corners

where k_i is the value of $k_i(0.5)$, δ is the chord error, and V_{max} is the command feed rate.

Step 3: Let $i = i$-1, if $i = 1$, go to step 4. Otherwise go to step 2.

Step 4: Store V'_i ($i = 2, 3 \ldots N$) into the buffer.

3.2.2 Modified forward scanning

Supposing the start feed rate of the first feed rate scheduling unit V_1 in the look-ahead window is known, modified forward scanning should be performed from the first FSU to the nth one to determine the end feed rate of each unit for real-time interpolation. As constraints of command, feed rate, chord error, and tangent acceleration/jerk limitations have been considered for $V'_i(i = 2,3..n)$ in backward scanning, and the tangent acceleration/jerk limitations are only considered in forward scanning. The details of the modified backward scanning process are given as follows, where $V_{(i+1)}$ is the result end point feed rate of ith feed rate scheduling unit.

Step 1: Set $i = 1$

Step 2: According to the length of ith feed rate scheduling unit Lu_i, set $S_a = Lu_i$, $\mathrm{V_s} = V_i$, then implement the algorithm of S-shape acceleration to get the final feed rate V_e. Compare the backward result feed rate $V'_{(i+1)}$ with V_e, the suitable end feed rate for the ith FSU is determined by

$$V_{i+1} = \min\{V_e, V'_{i+1}\}$$

Step 3: Let $i = i + 1$, if $i > n$, go to step 4. Otherwise go to step 2.

Step 4: Store V_i ($i = 2, 3 \ldots n$) into the buffer.

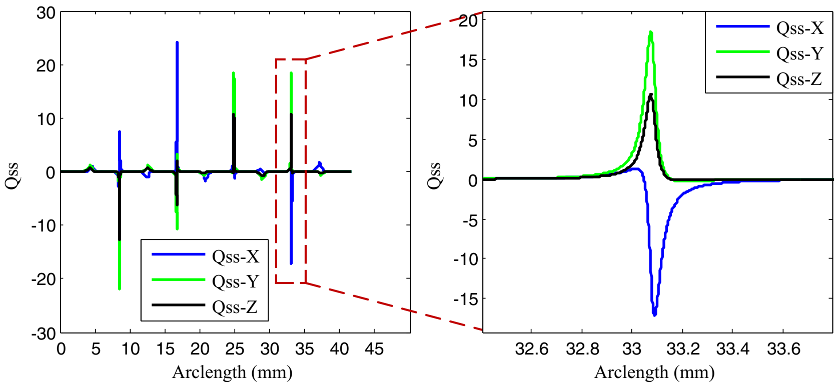


Fig. 11 $\mathbf{Q}_{ss}$ profile of the smoothed path

Finally, the information of *i*th FSU for feed rate scheduling is prepared as (V_i,$V_{(i+1)}$, Lu_i).

3.3 S-shape feed rate profiling with seven phases

When the modified bidirectional scanning process finished, the feed rate scheduling parameters of *i*-th FSU (V_{max}, A_{max}, J_{max}, V_i, $V_{(i+1)}$, Lu_i) are obtained. Then, the number of interpolation sampling time contained in each phase of feed rate profile needs to be determined. Since the number of interpolation cycle must be an integer, in order to ensure the interpolation results will not exceed the target position, the number of each interpolation cycle is rounded to negative direction in general; this process will make the final interpolation path shorter than the original path, resulting in a remaining distance. When the remaining distance is too long, it will cause the velocity discontinuity. In this paper, the online feed rate scheduling algorithm is used. An iterative way is adopted to limit the difference between the remaining distance before and after a scheduling controlled within a given tolerance L_{error} to minimize the remaining distance.

As shown in Fig. 4, the S-shape feed rate profile used in this paper contains seven stages. The first three stages are the acceleration segment. The total displacement corresponding to this segment is denoted as S_a. The fourth stage is the uniform segment. The total displacement is taken as S_u. The last three stages are the deceleration segment, and the total displacement at this stage is recorded as S_d. According to the given parameter conditions, the total time of some stages T_i may be zero. The feed rate scheduling process for once can be described as follows:

The *k*th time feed rate scheduling process for the *i*th FSU:

Step 1: Initialization: $v_s = v_3 = V_i, v_e = V_{(i+1)}, T_i = 0$ $(\mathrm{i} = 1, 2..7)$, $S_d = S_a = S_u = 0, L^k_{left} = L^{k-1}_{left}$ (where k = 0,1,...and $L^0_{left} = Lu_i$).

Step 2: If $v_3 > v_e$, calculate the distance S_d from v_3 to v_e and store the corresponding deceleration times T_5, T_6, T_7; go to step 3. Otherwise, go directly to step 3.

Step 3: Determine whether the remaining distance $Left = Lu_i - S_d - S_a - S_u$ is greater than zero; if yes, go to step 4. Otherwise, output the previous one *Left* as L^k_{left} which is greater than zero and its corresponding T_i, v_s, and v_e, and end the program.

Step 4: The v_s accelerate one time step, i.e., $T_1 = T_3 = T_1 + T_s$ or $T_2 = T_2 + T_s$ or $T_4 = T_4 + T_s$. Recalculate v_3; if $v_3 < V_{max}$, calculate the corresponding acceleration distance S_a. Otherwise, calculate the corresponding uniform segment S_u. Then, go to step 2.

The iterative way of above feed rate scheduling process is described as follows:

If the difference between L^k_{left} and L^{k-1}_{left} is greater than the set residual tolerance L_{error}, restart the above feed rate

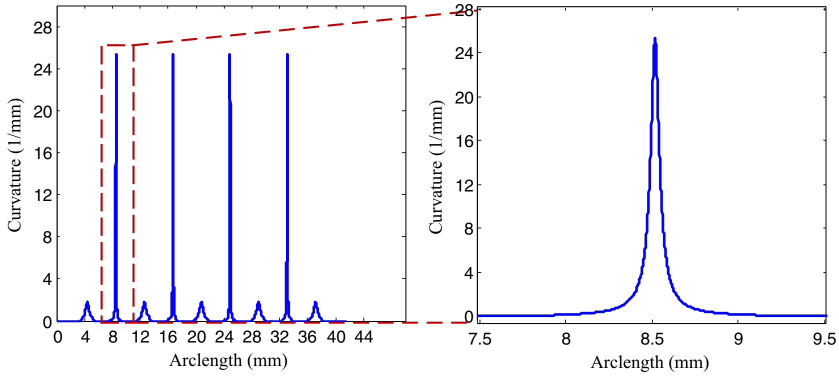


Fig. 12 Curvature profile of the smoothed path

Table 3 The local smoothing time consuming for each corner of the pentagram contour

Corner point	**Q**$_1$	**Q**$_2$	**Q**$_3$	**Q**$_4$	**Q**$_5$	**Q**$_6$	**Q**$_7$	**Q**$_8$	**Q**$_9$
Smoothing time(ms)	0.934	0.774	0.811	0.439	0.448	0.754	0.462	0.458	0.482

scheduling program process again with the previously obtained v_e as the starting speed v$_s$, $V_{(i+1)}$ still as the end of the speed v_e until the final difference is less than L_{error}.

Data update of the next FSU:

To ensure continuity of motion, the value of Lu_{i+1} for the next FSU will be updated to the sum of original Lu_{i+1} and the last L_{left} of the ith FSU.

After the data T_i, v$_s$ and v_e are obtained, through the formulas in Table 1, and the total displacement s at each sampling time can be calculated, then the increment displacement Δs_k^i at kth sampling time is derived.

3.4 Computation of interpolation points

Based on the Δs_k^i at kth sampling time of the ith FSU obtained in feed rate profiling, the interpolation points $\mathbf{InterP}_k^i = \left(X_k^i, Y_k^i, Z_k^i\right)$ should be calculated. According to the location of Δs_k^i in the ith FSU, there are two methods for interpolation. For the linear segment in FSU, the linear interpolation is implemented, which is given as

$$\mathbf{InterP}_k^i = \mathbf{InterP}_{(k-1)}^i + \frac{\left(\mathbf{InterP}_{(k-1)}^i - \mathbf{Q}^i\right)}{\left\|\mathbf{InterP}_{(k-1)}^i - \mathbf{Q}^i\right\|}\Delta s_k^i \tag{15}$$

For the B-spline segments, the parametric interpolation is adopted. Since there is no analytical relation between arc length and parameter for B-spline curve, the second-order Taylor's expansion method is usually adopted in traditional parametric interpolation algorithms [27–29]. But, the spline curve derivative with respect to the parameters is not easy to calculate, obtained by numerical analysis; the local truncation error of the Heun's method [30] and second-order Taylor expansion are both (h^3). But, the Heun's method only needs to calculate the value of first-order derivative, which can save the calculation time. In this paper, the Heun's method is used to calculate the curve parameters u_k^i corresponding to Δs_k^i; the formula is given as follows:

Table 4 Parameters used for the experimental test

Parameter	Symbol	Value
Chord error constrain		1 μm
Maximum approximation error		5 μm
Command feed rate constrain	F	50 mm/s
Tangent acceleration limitation	A_t	1000 mm/s^2
Normal acceleration limitation	A_n	1000 mm/s^2
Tangent jerk limitation	J_t	30,000 mm/s^3
Normal jerk limitation	J_n	30,000 mm/s^3
The sampling period	T_s	1 ms
The remaining distance tolerance	L_{error}	1 μm

$$\begin{cases} \tilde{u}_k^i = u_{(k-1)}^i + \left(\dfrac{du}{ds}\right)_{u=u_{(k-1)}^i}\Delta s_k^i \\ u_k^i = u_{(k-1)}^i + \dfrac{1}{2}\left(\left(\dfrac{du}{ds}\right)_{u=u_{(k-1)}^i} + \left(\dfrac{du}{ds}\right)_{u=\tilde{u}_k^i}\right)\Delta s_k^i \end{cases} \tag{16}$$
$$\frac{du}{ds} = \frac{1}{\dfrac{ds}{du}} = \frac{1}{\|\mathbf{C}_u\|}$$

where Δs_k^i would be replaced by the following value at first, if it is located at the junction of line and B-spline $\mathbf{C}_i$(u):

$$\begin{aligned} &if\left(\frac{Cl_{(i-1)}}{2} - Ls_i\right) < \sum_{j=1}^{k}\Delta s_j^i < Lu_i \\ &\Delta s_k^i = \sum_{j=1}^{k}\Delta s_j^i - \frac{Cl_{(i-1)}}{2} - Ls_i \quad , where\ Cl_0 = 0 \end{aligned} \tag{17}$$

After the parametric interpolation, the Cox-de Boor algorithm is adopted to evaluate the new interpolation point $\mathbf{InterP}_k^i$ by substituting the new parameter u_k^i into the path description $C_i\left(u_k^i\right)$ for its numerical stability and computational efficiency. Finally, the position commands are sent to the servo controller to perform real-time motion control.

4 Simulation results

In this section, analytic simulations are performed to verify the C^2 continuity of the proposed local tool path smoothing scheme and the proposed real-time interpolation algorithm. The algorithm is developed by VS 2005 in an Intel Core i3-

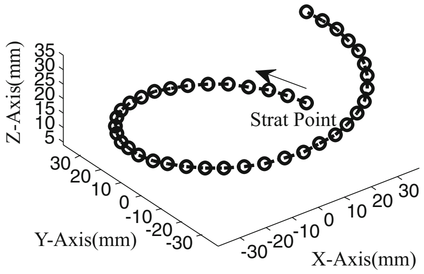


Fig. 13 The spiral tool tip trajectory

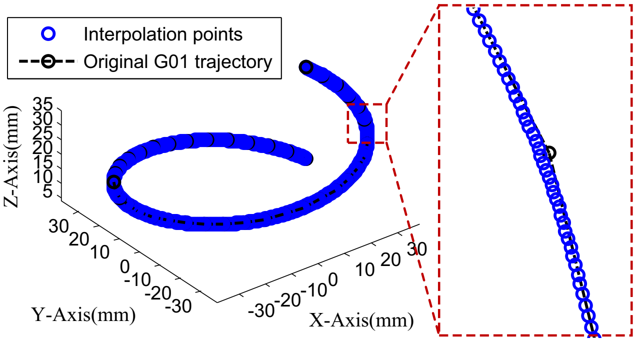


Fig. 14 Local smoothing interpolation for the spiral trajectory

2120 3.3 GHz personal computer with Windows 8 operating system. And, the figure is plotted by MATLAB.

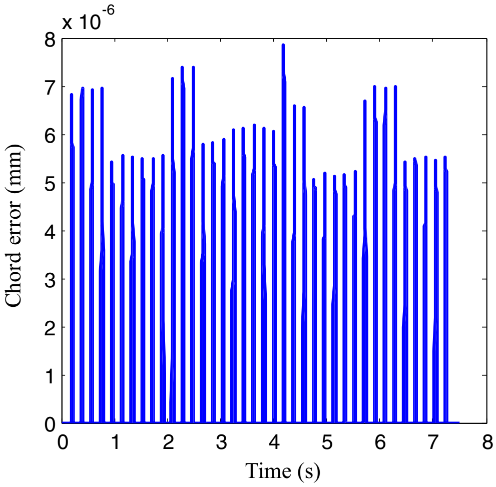


Fig. 16 Chord error with constraints of smoothing interpolation

4.1 Verification of C^2 continuity of the smoothed tool path

In simulation, the proposed local tool path smoothing scheme is implemented for a 3D pentagram linear contour shown in Fig. 9. The corner points are marked with $\mathbf{Q}_i$, $i = 0,1,...9$, whose data is listed in Table 2. And, the length of each G01 block is all the same as 4.5396 mm. There are two kinds of corner in this pentagram contour, sharp corners, and obtuse corners. To make the smooth effect obvious to see, the maximum approximation error is set as 1 mm. The resulted trajectory of local tool path corner smoothing algorithm is shown in Fig. 9.

In the two corners of the pentagram trajectory, which are acute angle and obtuse angle, the actual approximation error which is defined as the shortest distance from the point on the original straight corner to the curve after the smoothing with respect to the parameter u of the corner smoothing curve is depicted in Fig. 10. It can be seen that the approximation error constrained on the given limit of 1 mm, and the parameter u corresponding to the maximum approximation error is equal to 0.5, which is consistent with the previous description in Sect. 2. In order to verify that the resulting smooth track has C^2 continuity, the second derivatives $\mathbf{Q}_{ss}$ profile of the resulting contour with respect to the arc-length are depicted in Fig. 11 according to Eqs. (19) and (20). As observed, the $\mathbf{Q}_{ss}$ profiles are continuous. It demonstrates that the resulted trajectory is C^2 continuous. And, the curvature with respect to the arc-length is presented in Fig. 12. As observed, in order to

Fig. 15 The velocity/acceleration/jerk profiles of smoothing interpolation

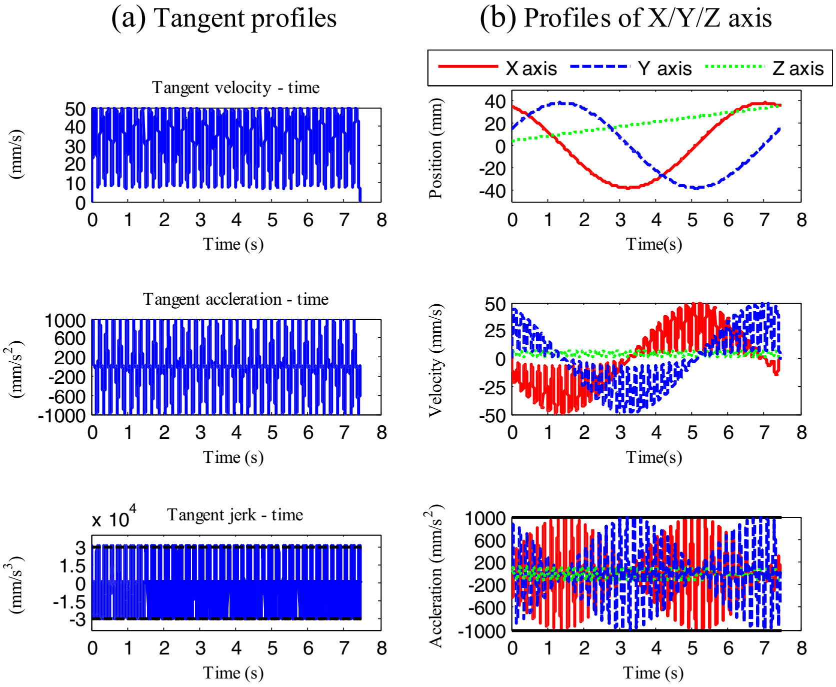
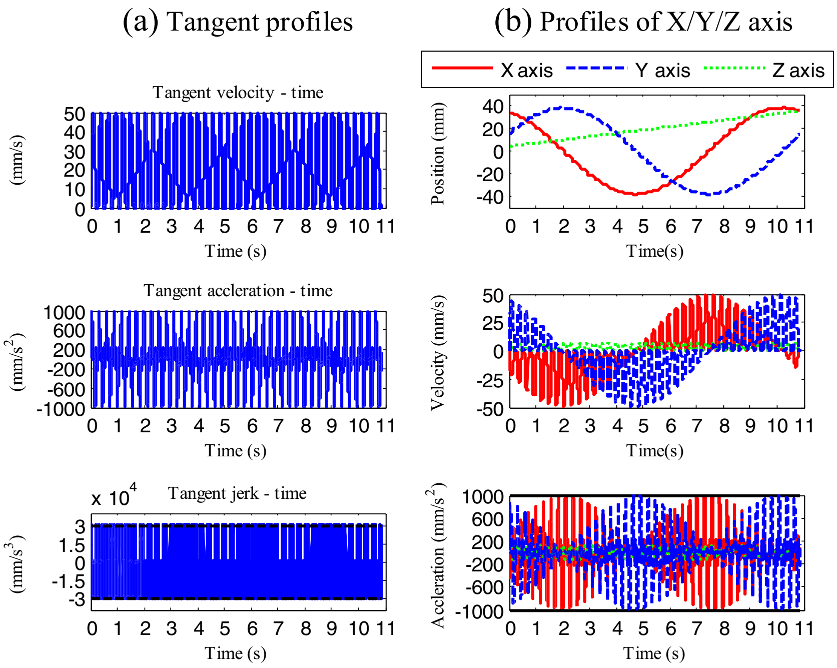


Fig. 17 The velocity/acceleration/jerk profiles of G01 interpolation



meet the approximation error limit, the curvature at the sharp corner corresponds to a larger one. These prove that the proposed local tool path smoothing algorithm is effective.

The time required for the calculation of the necessary information of each corner curve such as the control points in the figure is shown in Table 3. It can be concluded that the average time of the operation is less than 0.1 ms, which is fast. It proves that the proposed smoothing scheme is suitable for real-time interpolation process.

4.2 Process efficiency testification

-The interpolation algorithm as shown in Fig. 3 is implemented in VS 2005. After interpolation, the reference commands are stored and plotted in MATLAB. A 3D spiral linear trajectory which is a scaled result as shown in [31] is used in this simulation. To make the comparative analysis, the point-to-point G01 interpolation with the same S-shape feed rate scheduling algorithm as the proposed smoothing interpolation is also adopted to this trajectory. The set parameters of interpolation and the limitations used for the simulation are summarized in Table 4.

The 3D spiral trajectory as shown in Fig. 13 consists of 39 linear segments whose lengths are all the same as 6.1809 mm. The experimental results of this trajectory are shown in Figs. 14, 15, 16, and 17.

As shown in Figs. 15 and 17, the tangential velocity profile of smoothing interpolation at the corner is not down to zero, while the tangential feed rate of point-to-point interpolation is zero at the corner for there is only C^0 continuous. And, the acceleration and jerk are all within set limitations.

The chord error profile of the proposed smoothing interpolation is calculated in MATLAB and is plotted in Fig. 16. As observed, the chord error is under the set constrain. Please note that the chord error is much smaller than the set value due to the consideration of the normal acceleration/jerk limitations.

The total cycle time of point-to-point interpolation is 10.88 s, while it reduces to 7.49 s by using smoothing interpolation. The total cycle time is reduced by about 31.2%.

From these experimental results, it can be determined that the proposed smooth interpolation algorithm can achieve non-stop processing methods. For the same trajectory, the processing time can be greatly reduced. As the proposed feed rate scheduling algorithm takes the chord error and machine kinematic performance into account, when the tool moves to a larger curvature point, it will slow down in advance, so that the chord error is within the set constrain. In the meantime, the speed and acceleration of each axis are within the set limits. The validity of the smoothing interpolation algorithm is verified.

5 Conclusion

A local smoothing interpolation method for short line segments. Cubic B-spline with five control points are designed to blend the discrete linear segments to get C^2 continuous tool path, and a real-time look-ahead interpolation algorithm consists of four modules: smoothing module, modified bidirectional scanning module, feed rate profiling module, and interpolation module.

Compared with the previous work, the algorithms presented in this paper have the following advantages: (1) the relationship between the continuity of the trajectory and the continuity of the axes is described by formula. (2) The local smoothing scheme can realize C^2 continuity without complex calculation, and also control the approximation error. (3) The acceleration of each axis after feed rate scheduling is continuous and within limitation.

Finally, the effectiveness of the proposed algorithms is proved by the results of simulation.

Funding information The authors would like to thank the Important Science and Technology Specific Projects of Anhui Province, Nos. JZ2016AKKZ1067 and JZ2016AKKZ1069, and the National Natural Science Foundation of China under Grant Nos. 51575154 and 51505118.

Appendix

In order to guarantee the continuity of the motion of each axis of machine during processing, it should ensure the C^2 continuity of the trajectory with respect to arc length and the continuity of tangential velocity and acceleration. The proof of it is as follows:

Note that the trajectory of motion is **q**, according to the chain derivation rule, and the following expression can be derived:

$$\begin{aligned}
\mathbf{q}_t &= \mathbf{q}_s \bullet F, \textit{where } \mathbf{q}_t = \frac{d\mathbf{q}}{dt} = \begin{pmatrix} v^x \\ v^y \\ v^z \end{pmatrix}, \mathbf{q}_s = \frac{d\mathbf{q}}{ds}, F = \frac{ds}{dt} \\
\mathbf{q}_{tt} &= \mathbf{q}_{ss} \bullet F^2 + \mathbf{q}_s \bullet A, \textit{where } \mathbf{q}_{tt} = \frac{d^2 q}{dt^2} = \begin{pmatrix} a^x \\ a^y \\ a^z \end{pmatrix}, \mathbf{q}_{ss} = \frac{d^2\mathbf{q}}{ds^2}, A = \frac{d^2 s}{dt^2} \\
\mathbf{q}_{ttt} &= \mathbf{q}_{sss} \bullet F^3 + 3\mathbf{q}_{ss} \bullet F \bullet A + \mathbf{q}_s \bullet J, \textit{where } \mathbf{q}_{ttt} = \frac{d^3\mathbf{q}}{dt^3} = \begin{pmatrix} j^x \\ j^y \\ j^z \end{pmatrix}, \mathbf{q}_{sss} = \frac{d^3\mathbf{q}}{ds^3}, J = \frac{d^3 s}{dt^3}
\end{aligned} \tag{18}$$

where s is the arc-length of trajectory; F, A, and J are the tangential velocity, acceleration, and jerk, respectively.

When F, A, and $\mathbf{q}_s$, $\mathbf{q}_{ss}$ are continuous, it can guarantee that $\mathbf{q}_t$, $\mathbf{q}_{tt}$ are continuous. So, by ensuring the C^2 continuity of the trajectory, and through the S-shaped feed rate scheduling to make F and A continuous, it can ensure the continuous movement of each axis.

If the trajectory is a B-spline as $\mathbf{q}(u)$, $\mathbf{q}_s$, $\mathbf{q}_{ss}$, and $\mathbf{q}_{sss}$ can be expressed by

$$\begin{aligned}
\mathbf{q}_s &= \mathbf{q}_u \bullet u_s \\
\mathbf{q}_{ss} &= \mathbf{q}_{uu} \bullet {u_s}^2 + \mathbf{q}_u \bullet u_{ss} \\
\mathbf{q}_{sss} &= \mathbf{q}_{uuu} \bullet {u_s}^3 + 3\mathbf{q}_{uu} \bullet u_s \bullet u_{ss} + \mathbf{q}_u \bullet u_{sss}
\end{aligned} \tag{19}$$

where u is the parameter of the B-spline $\mathbf{q}(u)$, and u$_s$, u$_{ss}$, and u$_{sss}$ can be derived as

$$\begin{aligned}
u_s &= \frac{1}{\|\mathbf{q}_u\|} \\
u_{ss} &= -\frac{\mathbf{q}_u \bullet {\mathbf{q}_{uu}}^T}{\|\mathbf{q}_u\|^4} \quad \textit{while curvature}\rho = \frac{\|\mathbf{q}_u \times \mathbf{q}_{uu}\|}{\|\mathbf{q}_u\|^3} \\
u_{sss} &= \frac{4(\mathbf{q}_u \bullet {\mathbf{q}_{uu}}^T)^2 - (\mathbf{q}_{uu} \bullet {\mathbf{q}_{uu}}^T + \mathbf{q}_u \bullet {\mathbf{q}_{uu}}^T)(\mathbf{q}_u \bullet {\mathbf{q}_u}^T)}{\|\mathbf{q}_u\|^7}
\end{aligned} \tag{20}$$

From Eqs.(19) and (20), when $\mathbf{q}_u$, $\mathbf{q}_{uu}$ are continuous, it can lead to the continuity of $\mathbf{q}_s$, $\mathbf{q}_{ss}$, which means the C^2 continuity of B-spline with respect to parameter u can lead to the C^2 continuity of B-spline with respect to arc-length s.

References

1. Yang J, Altintas Y (2013) Generalized kinematics of five-axis serial machines with non-singular tool path generation. Int J Mach Tools Manuf 75:119–132. https://doi.org/10.1016/j.ijmachtools.2013.09.002
2. Tsai M, Nien H, Yau H (2010) Development of a real-time lookahead interpolation methodology with spline-fitting technique for high-speed machining. Int J Adv Manuf Technol 47(5–8):621–638. https://doi.org/10.1007/s00170-009-2220-7
3. Yang J, Chen Y, Chen Y, Zhang D (2015) A tool path generation and contour error estimation method for four-axis serial machines. Mechatronics 31:78–88. https://doi.org/10.1016/j.mechatronics.2015.03.001
4. Wang J, Yau H (2009) Real-time NURBS interpolator: application to short linear segments. Int J Adv Manuf Technol 41(11–12):1169–1185. https://doi.org/10.1007/s00170-008-1564-8
5. Li W, Liu Y, Yamazaki K, Fujisima M, Mori M (2008) The design of a NURBS pre-interpolator for five-axis machining. Int J Adv Manuf Technol 36(9–10):927–935. https://doi.org/10.1007/s00170-006-0905-8
6. Zhang M, Yan W, Yuan C, Wang D, Gao X (2011) Curve fitting and optimal interpolation on CNC machines based on quadratic B-splines. Sci China Inf Sci 54(7):1407–1418. https://doi.org/10.1007/s11432-011-4237-4
7. S Yeh HS (2009) Implementation of online NURBS curve fitting process on CNC machines. Int J Adv Manuf Technol 40(5–6):531–540. https://doi.org/10.1007/s00170-007-1361-9
8. Pateloup V, Duc E, Ray P (2010) Bspline approximation of circle arc and straight line for pocket machining. Comput Aided Des 42(9):817–827. https://doi.org/10.1016/j.cad.2010.05.003

9. Zhao H, Zhu L, 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. https://doi.org/10.1016/j.ijmachtools.2012.10.005
10. Beudaert X, Lavernhe S, Tournier C (2013) 5-axis local corner rounding of linear tool path discontinuities. Int J Mach Tools Manuf 73:9–16. https://doi.org/10.1016/j.ijmachtools.2013.05.008
11. Tulsyan S, Altintas Y (2015) Local toolpath smoothing for five-axis machine tools. Int J Mach Tools Manuf 96:15–26. https://doi.org/10.1016/j.ijmachtools.2015.04.014
12. SJ Yutkowitz. (2005). Apparatus and method for smooth cornering in a motion control system
13. Farouki RT (2014) Construction of rounded corners with Pythagorean-hodograph curves. Comput Aided Geom D 31(2): 127–139. https://doi.org/10.1016/j.cagd.2014.02.002
14. Q Bi, Y Wang, L Zhu, H Ding. (2011). A practical continuous-curvature Bézier transition algorithm for high-speed machining of linear tool path (7102, pp. 465–476). Berlin, Heidelberg: Springer Berlin Heidelberg. (Reprinted. doi: https://doi.org/10.1007/978-3-642-25489-5_45
15. Annoni M, Bardine A, Campanelli S, Foglia P, Prete CA (2012) A real-time configurable NURBS interpolator with bounded acceleration, jerk and chord error. Comput Aided Des 44(6):509–521. https://doi.org/10.1016/j.cad.2012.01.009
16. Lu L, Zhang L, Ji S, Han Y, Zhao J (2016) An offline predictive feedrate scheduling method for parametric interpolation considering the constraints in trajectory and drive systems. Int J Adv Manuf Technol 83(9–12):2143–2157. https://doi.org/10.1007/s00170-015-8112-0
17. Sencer B, Altintas Y, Croft E (2008) Feed optimization for five-axis CNC machine tools with drive constraints. Int J Mach Tools Manuf 48(7–8):733–745. https://doi.org/10.1016/j.ijmachtools.2008.01.002
18. Dong J, Ferreira PM, Stori JA (2007) Feed-rate optimization with jerk constraints for generating minimum-time trajectories. Int J Mach Tools Manuf 47(12–13):1941–1955. https://doi.org/10.1016/j.ijmachtools.2007.03.006
19. Erkorkmaz K, Altintas Y (2001) High speed CNC system design. Part I: jerk limited trajectory generation and quintic spline interpolation. Int J Mach Tools Manuf 41(9):1323–1345. https://doi.org/10.1016/S0890-6955(01)00002-5
20. Lin M, Tsai M, Yau H (2007) Development of a dynamics-based NURBS interpolator with real-time look-ahead algorithm. Int J Mach Tools Manuf 47(15):2246–2262. https://doi.org/10.1016/j.ijmachtools.2007.06.005
21. Piegl L, Tiller W (1997) NURBS book. Springer, Berlin/Heidelberg
22. Noble B (1964) Numerical methods. Oliver and Boyd, New York
23. Liu M, Huang Y, Yin L, Guo J, Shao X, Zhang G (2014) Development and implementation of a NURBS interpolator with smooth feedrate scheduling for CNC machine tools. Int J Mach Tools Manuf 87:1–15. https://doi.org/10.1016/j.ijmachtools.2014.07.002
24. Lee A, Lin M, Pan Y, Lin W (2011) The feedrate scheduling of NURBS interpolator for CNC machine tools. Comput Aided Des 43(6):612–628. https://doi.org/10.1016/j.cad.2011.02.014
25. Du X, Huang J, Zhu L (2015) A complete S-shape feed rate scheduling approach for NURBS interpolator. J Comput Des Eng 2(4): 206–217. https://doi.org/10.1016/j.jcde.2015.06.004
26. Beudaert X, Pechard P, Tournier C (2011) 5-axis tool path smoothing based on drive constraints. Int J Mach Tools Manuf 51(12):958–965. https://doi.org/10.1016/j.ijmachtools.2011.08.014
27. Lei WT, Wang SB (2009) Robust real-time NURBS path interpolators. Int J Mach Tools Manuf 49(7–8):625–633. https://doi.org/10.1016/j.ijmachtools.2009.01.007
28. Zhang X, Song Z (2012) An iterative feedrate optimization method for real-time NURBS interpolator. Int J Adv Manuf Technol 62(9–12):1273–1280. https://doi.org/10.1007/s00170-011-3847-8
29. Baek DK, Yang S, Ko TJ (2013) Precision NURBS interpolator based on recursive characteristics of NURBS. Int J Adv Manuf Technol 65(1–4):403–410. https://doi.org/10.1007/s00170-012-4179-z
30. Chen M, Zhao W, Xi X (2015) Augmented Taylor's expansion method for B-spline curve interpolation for CNC machine tools. Int J Mach Tools Manuf 94:109–119. https://doi.org/10.1016/j.ijmachtools.2015.04.013
31. Yang J, Altintas Y (2015) A generalized on-line estimation and control of five-axis contouring errors of CNC machine tools. Int J Mach Tools Manuf 88:9–23. https://doi.org/10.1016/j.ijmachtools.2014.08.004