Int J Adv Manuf Technol (2017) 88:961–969
DOI 10.1007/s00170-016-8785-z

ORIGINAL ARTICLE

A novel approach for NURBS interpolation through the integration of acc-jerk-continuous-based control method and look-ahead algorithm

Liu Xinhua[1] · **Peng Junquan**[1] · **Si Lei**[1,2] · **Wang Zhongbin**[1]

Received: 12 February 2016 / Accepted: 15 April 2016 / Published online: 4 May 2016

Abstract In order to reduce high chord errors and improve poor flexibility of interpolation, a novel approach for non-uniform rational B-spline (NURBS) interpolation through the integration of an acc-jerk-continuous-based control method and look-ahead algorithm is proposed. At first, the principles of NURBS curve interpolation and implementation are described. Then, modules of the proposed algorithm, including the feed rate planning and acc-jerk-continuous modules, are established. In the feed rate planning module, sharp corners are detected and the curve is split into some NURBS blocks according to sharp corners. The knot parameter and restrict feed rate at the sharp corners are obtained as well as the length of each block is calculated. In the acc-jerk-continuous control module, information including the knot parameter, arc length, and start and end feed rates are handled to plan an acc-jerk-continuous trajectory. Finally, simulation with one free-form NURBS curve is conducted and comparison with the S-shaped scheduling algorithm is carried out to verify the feasibility and applicability of the proposed algorithm.

Keywords NURBS interpolation · Acc-jerk-continuous-based control method · Look-ahead algorithm

1 Introduction

Computer numerical control (CNC) machining has become an important part in modern manufacturing fields with growing demands of high-accuracy and high-efficiency machining. Complex machined with conventional linear and circular interpolation will bring such a problem. The curve of parts needs to be discretized into a large number of small line segments; the more small line segments generate, the larger the transmission data will be. Since the non-uniform rational B-spline (NURBS) curve has lots of advantages in designing complicated shapes compared with conventional linear and circular interpolation, its interpolation methods have been developed in computer-aided design/computer-aided manufacturing (CAD/CAM) applications [1, 2].

Although many methods connected with NURBS interpolation have been proposed to obtain efficient and accurate machining, most of them have some common disadvantages such as high chord errors and poor flexibility. To the best of our knowledge, the problems of NURBS interpolation for CNC machines have almost not been dealt with. Based on our past researches on NURBS interpolation algorithms, this paper tries to tackle these problems.

Bearing the above observations in mind, a novel approach for NURBS interpolation through the integration of an acc-jerk-continuous-based control method and look-ahead (IAL) algorithm is applied. The rest of this paper is

✉ Peng Junquan
vrlabpaper@126.com

✉ Wang Zhongbin
wangzbpaper@126.com

[1] School of Mechanical and Electrical Engineering, China University of Mining & Technology, Xuzhou, China

[2] School of Information and Electrical Engineering, China University of Mining & Technology, Xuzhou, China

organized as follows. In Section 2, some related works are outlined based on the literature. The principles of NURBS curve interpolation and implementation are presented in Section 3. In Section 4, the integrated approach is proposed. In Section 5, simulations with one free-form NURBS curve is conducted to verify the feasibility and applicability of the proposed method. Our conclusions and future works are summarized in Section 6.

2 Literature review

Recent publications relevant to this paper are mainly concerned with two research streams: the look-ahead algorithm and the acceleration/deceleration (ACC/DEC) control method for NURBS interpolation. In this section, we try to summarize the relevant literature.

2.1 The look-ahead algorithm for NURBS interpolation

For the look-ahead algorithm, lots of research has been done since the last decade. In [3], Nam and Yang proposed a look-ahead scheme with a jerk-limited acceleration for smoothing feed rate profile. In [4], Tsai et al. applied a hybrid digital convolution technique to develop a look-ahead scheme that smoothed the feed rate between the joint of two curves. In [5], Liu et al. developed an algorithm that integrated look-ahead, jerk-limited trajectory planning techniques to reduce velocity and acceleration fluctuations. In [6], Yau et al. proposed a real-time look-ahead algorithm by fitting continuous short blocks into Bezier curves under a given chord tolerance. In [7], Lin et al. proposed a dynamics-based interpolator with real-time look-ahead algorithm to generate a smooth and jerk-limited ACC/DEC feed rate profile. In [8], Ye et al. presented an interpolation method based on the look-ahead algorithm to machine complicated contours of special parts. In [9], Tsai et al. proposed an integrated look-ahead dynamics-based algorithm with the consideration of geometric and servo errors simultaneously. In [10], Wang et al. introduced a real-time look-ahead algorithm to detect sharp corners. In [11], Emami and Arezoo introduced a look-ahead trajectory generation method which determines the deceleration stage according to the fast estimated arc length. In [12], Zhang et al. proposed a new and more efficient look-ahead method and a feed rate override method to boast the global machining speed. In [13], Wang and Cao developed a novel look-ahead and adaptive speed control algorithm to improve the efficiency of rapid linking of feed rate for high-speed machining. In [14], Zhao et al. presented a real-time look-ahead scheme which comprised path-smoothing, bi-directional scanning, and feed rate scheduling to acquire a feed rate profile with smooth acceleration. In [15], Wang et al. developed a real-time look-ahead interpolation algorithm based on Akima curve fitting. In [16], Jin et al. proposed a parametric interpolation method integrated to a novel real-time look-ahead algorithm as well as presented some improvements in the fine interpolation stage to further enhance the machining accuracy. In [17], Qiao et al. studied the flexible ACC/DEC and look-ahead trajectory planning method systematically.

2.2 The ACC/DEC control method for NURBS interpolation

Back in 2003, in [18], Yong and Narayanaswami investigated an off-line NURBS interpolation method to simultaneously keep both the chord error within a specified tolerance and the ACC/DEC of corner machining as well as the ACC/DEC profile polynomial. In [19], Park et al. proposed a two-stage interpolation method that compensates for interpolation errors within the machine basic length unit which yield undesirable acceleration and jerk values around the regions. In [20], Xu et al. proposed an adaptive interpolation scheme incorporating a machine's dynamic capability consideration based on the trapezoidal or triangular ACC/DEC feed rate profile. In [21], Sekar et al. developed a method for obtaining continuous velocity and acceleration profiles which reduces the jerk-related problems in high-speed machining. In [22], Wang and Yau proposed the use of a real-time NURBS interpolator with a look-ahead function to handle numerous short linear segments adopting the S-shaped jerk-limited acceleration method. In [23], Shen et al. generated the velocity profile around a corner in the ACC/DEC stage according to the trapezoidal or triangular ACC/DEC profile. In [24], Du et al. developed an adaptive NURBS curve interpolator with real-time and flexible ACC/DEC control scheme by considering a preset jerk range with the S-shaped velocity profile. In [25], Wang et al. introduced the adaptive ACC/DEC and jerk-limited module to smooth the velocity sharp corner with the S-shaped velocity profile. In [26], Annoni proposed a NURBS interpolator that is able to satisfy all the manufacturing technology requirements based on the cubic polynomial method. In [27], Wang developed a curvature-based NURBS surface interpolator with look-ahead ACC/DEC control with cubic spline methods. In [28], Lee et al. proposes an off-line feed rate scheduling method of CNC machines constrained by chord tolerance, acceleration, and jerk limitations with the sine-curve velocity profile.

2.3 Discussion

Although many approaches for NURBS interpolation have been proposed in the above literature, they have some common disadvantages, summarized as follows. Firstly, most of the researchers present their methods for NURBS

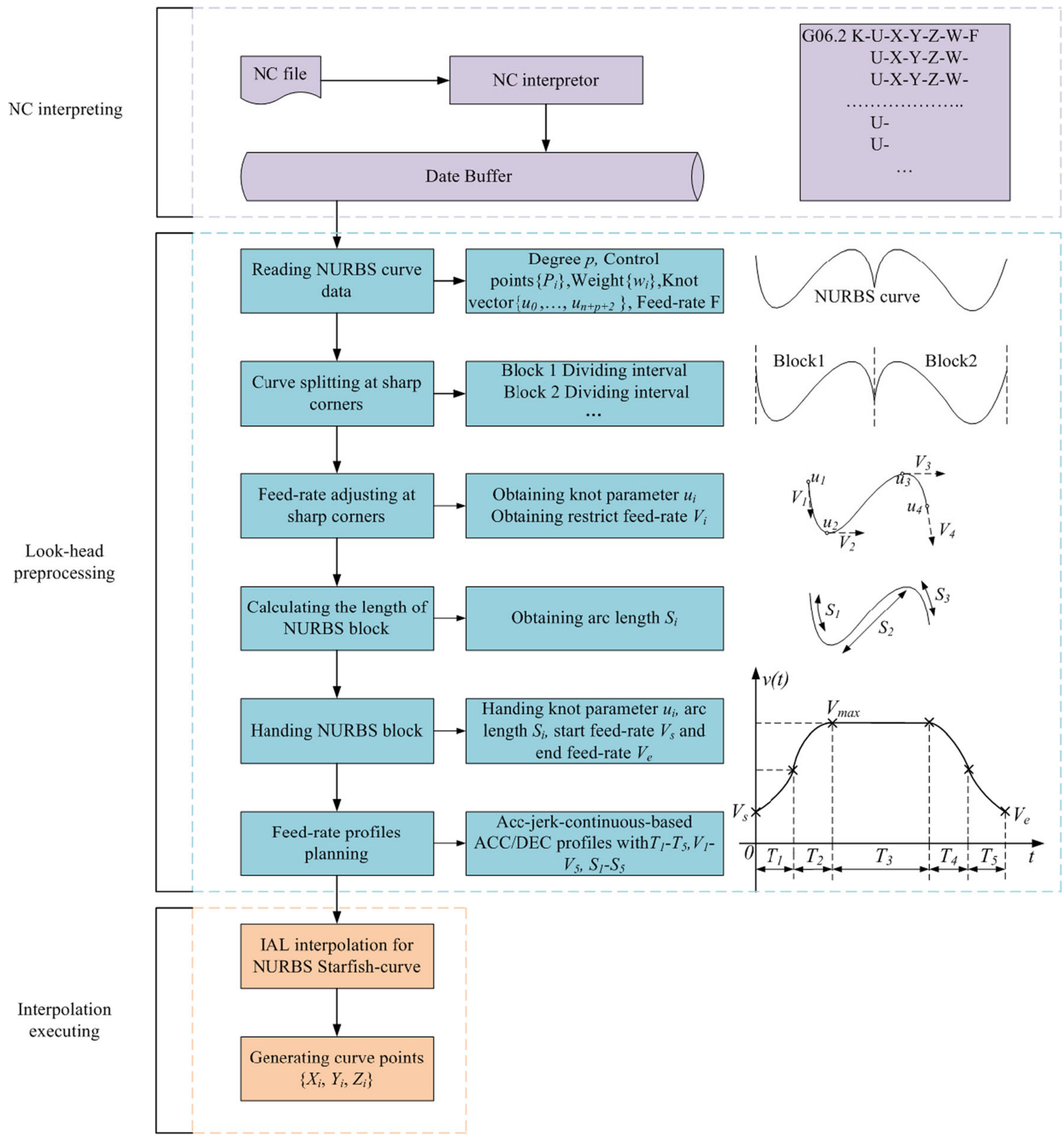

Fig. 1 System architecture of the proposed NURBS interpolation method

interpolation with the second-order Taylor expansion method. However, the second-order Taylor expansion method refers to the second derivative of the NURBS curve and is complicated. Secondly, either the ACC/DEC control method for NURBS interpolation is complicated and time-consuming or the jerk of the ACC/DEC control method is discontinuous as well as it will bring flexible impulse to CNC machines. In order to solve the above problems, this paper proposes a novel approach for NURBS interpolation through the integration of an acc-jerk-continuous-based control and look-ahead algorithm. Simulation with one free-form NURBS curve is conducted to verify the feasibility and applicability of the proposed method.

3 Principle of NURBS curve interpolation and implementation

Suppose $C(u)$ represents a NURBS curve and is given by [29]

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

where $\{P_i\}$ are the control points, $\{w_i\}$ is the weight of $\{P_i\}$, $(n+1)$ is the number of the control points, and p is the degree of NURBS curve. $\{N_{i,p}(u)\}$ is the pth-degree B-spline basis function and can be calculated using the recursive formulas given as follows:

$$N_{i,0}(u) = \begin{cases} 1 & if \quad u_i \le u \le u_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{3}$$

where $\{u_0,\ldots, u_{n+p+2}\}$ represents the knot vectors and u is the interpolation parameter.

To implement NURBS interpolation, the Adams–Bashforth method (ABM) is adopted here. Using the ABM method, the curve approximation up is given as follows:

$$\begin{cases} u_{m+1} = u_m + \frac{T}{24}(55f_k - 59f_{k-1} + 37f_{k-2} - 9f_{k-3}) \\ f_k = \frac{du}{dt}\Big|_{t=t_m} = \frac{V(u_m)}{\|C'(u_m)\|} = \frac{V(u_m)}{\sqrt{\left(\frac{dC_x(u_m)}{du_m}\right)^2 + \left(\frac{dC_y(u_m)}{du_m}\right)^2 + \left(\frac{dC_z(u_m)}{du_m}\right)^2}} \end{cases} \tag{4}$$

where $V(u_m)$, T, and $C'(u_m)$ denote the feed rate, the sampling time, and the first derivatives of the NURBS curve, respectively, and $C_x(u_m)$, $C_y(u_m)$, and $C_z(u_m)$ denote the x, y, and z components of a point on the NURBS curve corresponding to the parameter u_m, respectively. The Cox–de Boor algorithm is chosen to compute the values of $C(u_m)$ and $C'(u_m)$ for implementation.

4 The proposed IAL algorithm

4.1 System architecture

In this subsection, the system architecture of the proposed IAL algorithm is established in Fig. 1, which is utilized to implement CNC control tasks. The IAL interpolation consists of two different parts: look-ahead preprocessing and interpolation executing. Look-ahead preprocessing consists of two different modules: the feed rate planning module and the acc-jerk-continuous control module. In the feed rate planning module, sharp corners are firstly detected and the curve is split into some NURBS blocks according to sharp corners. Then, the knot parameter u_i and the restrict feed rate V_i at the sharp corners are obtained as well as the length of each block S_i is calculated. Finally, in the acc-jerk-continuous control module, information including the knot parameter u_i, arc length S_i, start feed rate V_s, and end feed rate V_e are handled to plan an acc-jerk-continuous trajectory, and then ACC/DEC parameters are passed to complete the interpolation executing process.

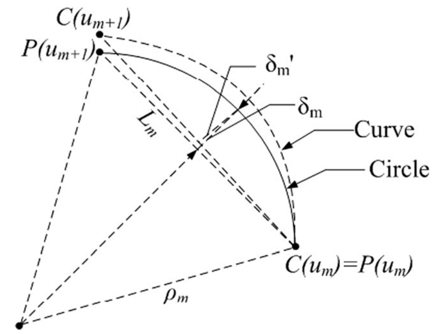

Fig. 2 Chord error between two interpolated points

4.2 Feed rate planning module

The task of the feed rate planning module is to obtain feed rate at the sharp corners. According to Eq. 4, the ABM interpolation algorithm cannot make the curve speed accurately equal to the desired value $V(u_m)$. To achieve high machine accuracy, the chord error must be controlled within a tolerance range during the interpolation, as shown in Fig. 2. There are two common methods to calculate the chord error. Equation 5 is a direct method with the assumption that the contour error δ_m' is located in the middle point of two adjacent interpolated points. Equation 6 is the other method using curve curvature to calculate the chord error δ_m.

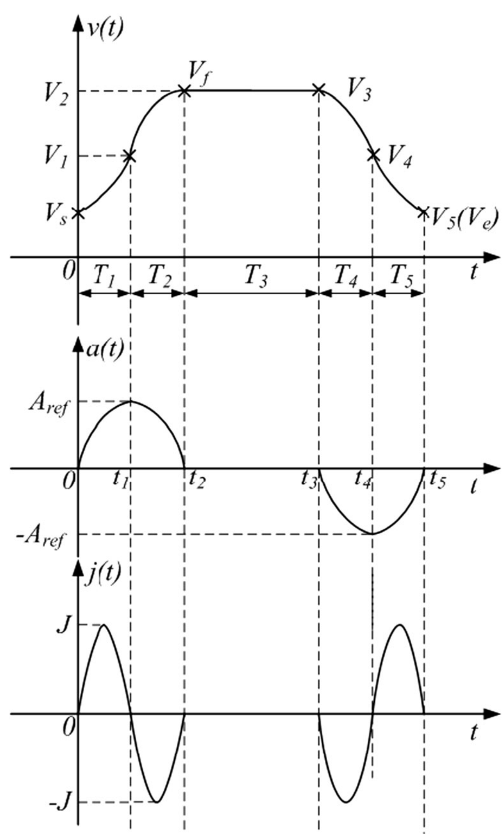

Fig. 3 The acc-jerk-continuous feed rate profile

$$\delta'_m = \left| C\left(\frac{u_m + u_{m+1}}{2}\right) - \frac{C(u_m) + C(u_{m+1})}{2} \right| \tag{5}$$

$$\delta_m = \rho_m - \sqrt{\rho_m{}^2 - \left(\frac{L_m}{2}\right)^2} \tag{6}$$

where $\rho_m = 1/k_m$; ρ_m is the radius of curvature at the parameter value u_m. The curvature k_m can be given by Eq. 7.

$$k_m = \frac{\left\| C'(u_m) - C''(u_m) \right\|}{\left\| C'(u_m) \right\|^3} \tag{7}$$

This paper uses Eq. 5 to calculate the chord errors since it is time-consuming to calculate the second derivative of the NURBS curve.

The predicted feed rate V_{pre} and the expected feed rate V_{exp} are given by Eqs. 8 and 9, respectively.

$$V_{pre} = \frac{d\sqrt{(C_x(u_m) - C_x(u_{m-1}))^2 + (C_y(u_m) - C_y(u_{m-1}))^2 + (C_z(u_m) - C_z(u_{m-1}))^2}}{dt} \tag{8}$$

$$V_{\exp} = \sqrt{\frac{\delta_{\max}}{\delta'_m}} V_{pre} \tag{9}$$

where $\delta_{\max}$ is the contour error limitation.

While the tool moves across the sharp corners, it may result in violent change of acceleration or jerk on each axis. So, three constraints of chord error, acceleration, and jerk are considered simultaneously in Eq. 10.

$$V_i = \min\left\{ \sqrt{\frac{\delta_{\max}}{\delta'_m}} V_{\text{pre}}, \frac{V_{\exp}}{2}\sqrt{\frac{A_{\max}}{2\delta_{\max}}}, \frac{V_{\exp}}{4}\sqrt[3]{\frac{J_{\max} V_{pre} T}{\delta^2_{\max}}} \right\} \tag{10}$$

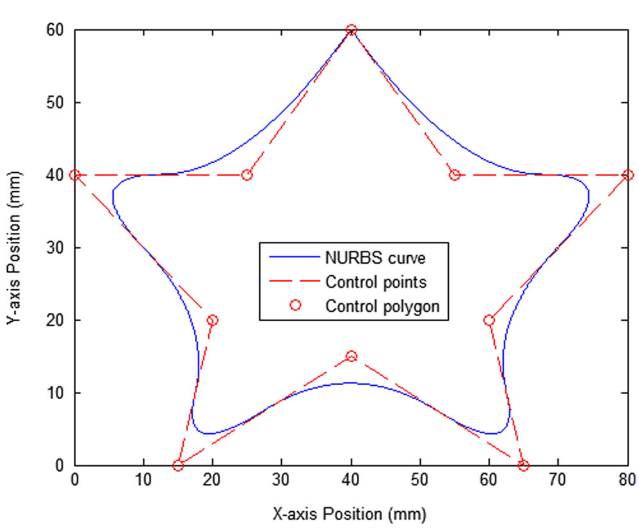

Fig. 4 A starfish curve

Table 1 Interpolation conditions for the starfish curve

Parameters	Symbols	Values
Feed rate	F	200 mm/s
Maximum acceleration	$A_{\max}$	3000 mm/s^2
Maximum jerk	$J_{\max}$	60,000 mm/s^3
Maximum chord error	$\delta_{\max}$	0.001 mm
Sampling time	T	0.001 s

where V_i, $A_{\max}$, and $J_{\max}$ are suitable feed rate at the sharp corners, maximum acceleration, and maximum jerk, respectively.

After obtaining the sharp corners, the curve can be divided into several segments. The length of each segment, L_{seg}, and the cumulative segment length, S_i, are calculated by the following equations [7]:

$$L^n_{\text{seg}} = \sum_{i=1}^{N_n} |C(u_{i+1}) - C(u_i)| \qquad S^a_i = \sum_{j=1}^{a} L^j_{\text{seg}} \tag{11}$$

where n is the index of each segment, N_n is the number of interpolation points within the mth segment, and a is the cumulative segment number of a NURBS curve.

4.3 Acc-jerk-continuous control method

When the feed rate planning module is finished, the value of (u_i, V_i) can be obtained and stored. In the stage of acc-jerk-continuous module, the task is to construct an acc-jerk-continuous feed rate profile shown in Fig. 3 since it is more

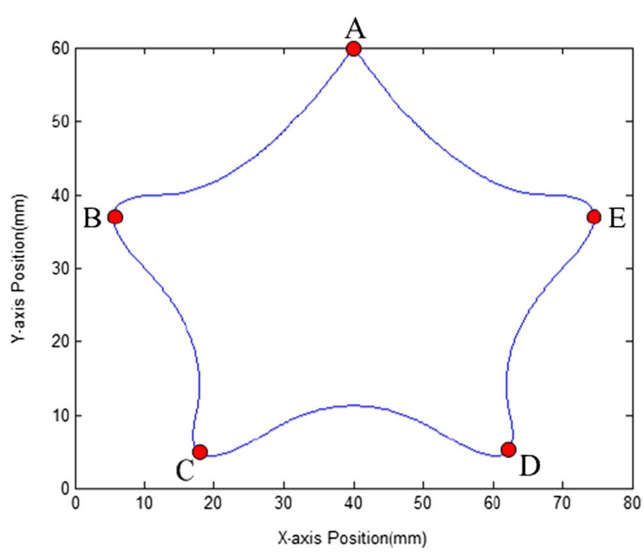

Fig. 5 Starfish curve and block

continuous than the typical S-shaped velocity profile. In this paper, the acc-jerk-continuous feed rate profile is chosen to generate the acceleration profile, and its acceleration equation can be given as formula 12.

$$A(t)=\begin{cases}\frac{A_{ref}}{2}\left(1-\cos\frac{\pi}{T_1}t\right) & 0\le t<t_1\\ \frac{A_{ref}}{2}\left(1+\cos\frac{\pi}{T_2}(t-t_1)\right) & t_1\le t<t_2\\ 0 & t_2\le t<t_3\\ -\frac{A_{ref}}{2}\left(1-\cos\frac{\pi}{T_4}(t-t_3)\right) & t_3\le t<t_4\\ -\frac{A_{ref}}{2}\left(1+\cos\frac{\pi}{T_5}(t-t_4)\right) & t_4\le t<t_5\end{cases}\tag{12}$$

where $T_1=T_2$ and $T_4=T_5$. Integrating Eq. 13 yields the velocity equation.

$$V(t)=\begin{cases}V_s+\frac{A_{ref}}{2}\left(t-\frac{T_1}{\pi}\sin\left(\frac{\pi}{T_1}t\right)\right) & 0\le t<t_1\\ V_s+\frac{A_{ref}T_1}{2}+\frac{A_{ref}}{2}\left((t-t_1)+\frac{T_2}{\pi}\sin\left(\frac{\pi}{T_2}(t-t_1)\right)\right) & t_1\le t<t_2\\ V_s+A_{ref}T_1 & t_2\le t<t_3\\ V_s+A_{ref}T_1-\frac{A_{ref}}{2}\left((t-t_3)-\frac{T_4}{\pi}\sin\left(\frac{\pi}{T_4}(t-t_3)\right)\right) & t_3\le t<t_4\\ V_s+A_{ref}T_1-\frac{A_{ref}T_4}{2}-\frac{A_{ref}}{2}\left((t-t_4)+\frac{T_5}{\pi}\sin\left(\frac{\pi}{T_5}(t-t_4)\right)\right) & t_4\le t<t_5\end{cases}\tag{13}$$

where V_s, V_e, V_f, and A_{ref} show the start, end, and maximum feed rates and the actual acceleration, respectively. Differentiating Eq. 13, one obtains the jerk equation.

$$J(t)=\begin{cases}\frac{\pi}{2T_1}A_{ref}\sin\frac{\pi}{T_1}t & 0\le t<t_1\\ -\frac{\pi}{2T_2}A_{ref}\sin\frac{\pi}{T_2}(t-t_1) & t_1\le t<t_2\\ 0 & t_2\le t<t_3\\ \frac{\pi}{2T_4}A_{ref}\sin\frac{\pi}{T_4}(t-t_3) & t_3\le t<t_4\\ -\frac{\pi}{2T_5}A_{ref}\sin\frac{\pi}{T_5}(t-t_4) & t_4\le t<t_5\end{cases}\tag{14}$$

According to Eq. 14, one equation can be obtained.

$$A_{ref}=\frac{\left(V_f-V_s\right)}{T_1}\tag{15}$$

where V_f is the allowable feed rate.

When Eq. 15 is substituted into Eqs. 12, 13, and 14, the new acceleration equation, velocity equation, and jerk equation can be expressed as

$$A(t)=\begin{cases}\frac{(V_f-V_s)}{2T_1}\left(1-\cos\frac{\pi}{T_1}t\right) & 0\le t<t_1\\ \frac{(V_f-V_s)}{2T_1}\left(1+\cos\frac{\pi}{T_1}(t-t_1)\right) & t_1\le t<t_2\\ 0 & t_2\le t<t_3\\ -\frac{(V_f-V_s)}{2T_4}\left(1-\cos\frac{\pi}{T_4}(t-t_3)\right) & t_3\le t<t_4\\ -\frac{(V_f-V_s)}{2T_4}\left(1+\cos\frac{\pi}{T_4}(t-t_4)\right) & t_4\le t<t_5\end{cases}\tag{16}$$

$$V(t)=\begin{cases}V_s+\frac{(V_f-V_s)}{2T_1}\left(t-\frac{T_1}{\pi}\sin\left(\frac{\pi}{T_1}t\right)\right) & 0\le t<t_1\\ V_s+\frac{A_{ref}T_1}{2}+\frac{(V_f-V_s)}{2T_1}\left((t-t_1)+\frac{T_1}{\pi}\sin\left(\frac{\pi}{T_1}(t-t_1)\right)\right) & t_1\le t<t_2\\ V_s+A_{ref}T_1 & t_2\le t<t_3\\ V_s+A_{ref}T_1-\frac{(V_f-V_s)}{2T_4}\left((t-t_3)-\frac{T_4}{\pi}\sin\left(\frac{\pi}{T_4}(t-t_3)\right)\right) & t_3\le t<t_4\\ V_s+A_{ref}T_1-\frac{A_{ref}T_4}{2}-\frac{(V_f-V_s)}{2T_4}\left((t-t_4)+\frac{T_4}{\pi}\sin\left(\frac{\pi}{T_4}(t-t_4)\right)\right) & t_4\le t<t_5\end{cases}\tag{17}$$

$$J(t)=\begin{cases}\frac{\pi(V_f-V_s)}{2T_1^2}\sin\frac{\pi}{T_1}t & 0\le t<t_1\\ -\frac{\pi(V_f-V_s)}{2T_1^2}\sin\frac{\pi}{T_1}(t-t_1) & t_1\le t<t_2\\ 0 & t_2\le t<t_3\\ \frac{\pi(V_f-V_s)}{2T_4^2}\sin\frac{\pi}{T_4}(t-t_3) & t_3\le t<t_4\\ -\frac{\pi(V_f-V_s)}{2T_4^2}\sin\frac{\pi}{T_4}(t-t_4) & t_4\le t<t_5\end{cases}\tag{18}$$

During the whole ACC+CF+DEC (Acceleration+Constant-Feedrate+Deceleration), according to the limits of velocity, acceleration, and jerk, the following equation can be obtained.

$$T_1=\max\left\{\frac{\left(V_f-V_s\right)}{A_{ref}},\sqrt{\frac{\pi\left(V_f-V_s\right)}{2J_{max}}}\right\}\tag{19}$$

The process of calculating T_4 is similar to that of calculating T_1. The areas under the ACC/DEC sections of the acc-jerk-continuous feed rate profile are derived as

$$S_{ACC} = \int_0^{t_1} V_1 dt + \int_{t_1}^{t_2} V_2 dt = \frac{(V_f + V_s) T_1}{4}$$
$$S_{DEC} = \int_{t_3}^{t_4} V_4 dt + \int_{t_4}^{t_5} V_5 dt = \frac{(V_f + V_e) T_4}{4} \quad (20)$$

Therefore, time in the CF section of the acc-jerk-continuous feed rate profile is obtained.

$$T_3 = \frac{S_i - S_{ACC} - S_{DEC}}{V_f} \quad (21)$$

where S_{ACC}, S_{DEC}, and S_i are the areas under the ACC and DEC and the length of the NURBS block.

5 Simulation results and interpolation analysis

In this section, a NURBS curve named starfish, shown in Fig. 4, is used as an example to test the feasibility of the developed interpolation scheme. The command generator programs are written by Matlab and are executed on a personal computer with Intel(R) Core(TM) i5-2450M 2.50-GHz CPU.

According to Fig. 4, the starfish curve has the following parameters:

$P_i = \{[40\ 60], [25\ 40], [0\ 40], [20\ 20], [15\ 0], [40\ 15], [65\ 0], [60\ 20], [80\ 40], [55\ 40], [40\ 60]\}$,

$w_i = \{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\}$,

$U = \{0, 0, 0, 0.1110, 0.2220, 0.3330, 0.4440, 0.5550, 0.6660, 0.7770, 0.8880, 1, 1, 1\}$,

where P_i, w_i, and U are the control point, weight, and knot vectors, respectively.

The interpolation conditions for the starfish curve are listed in Table 1. Based on the interpolation conditions for the starfish curve, the starfish curve and block are shown in Fig. 5.

From Fig. 6c, d, we know that the acceleration and jerk value of the S-shape-based interpolation algorithm are both beyond the maximum acceleration and jerk. However, the acceleration and jerk value of the proposed IAL interpolation algorithm are controlled within the maximum acceleration and jerk in Fig. 7c, d, respectively.

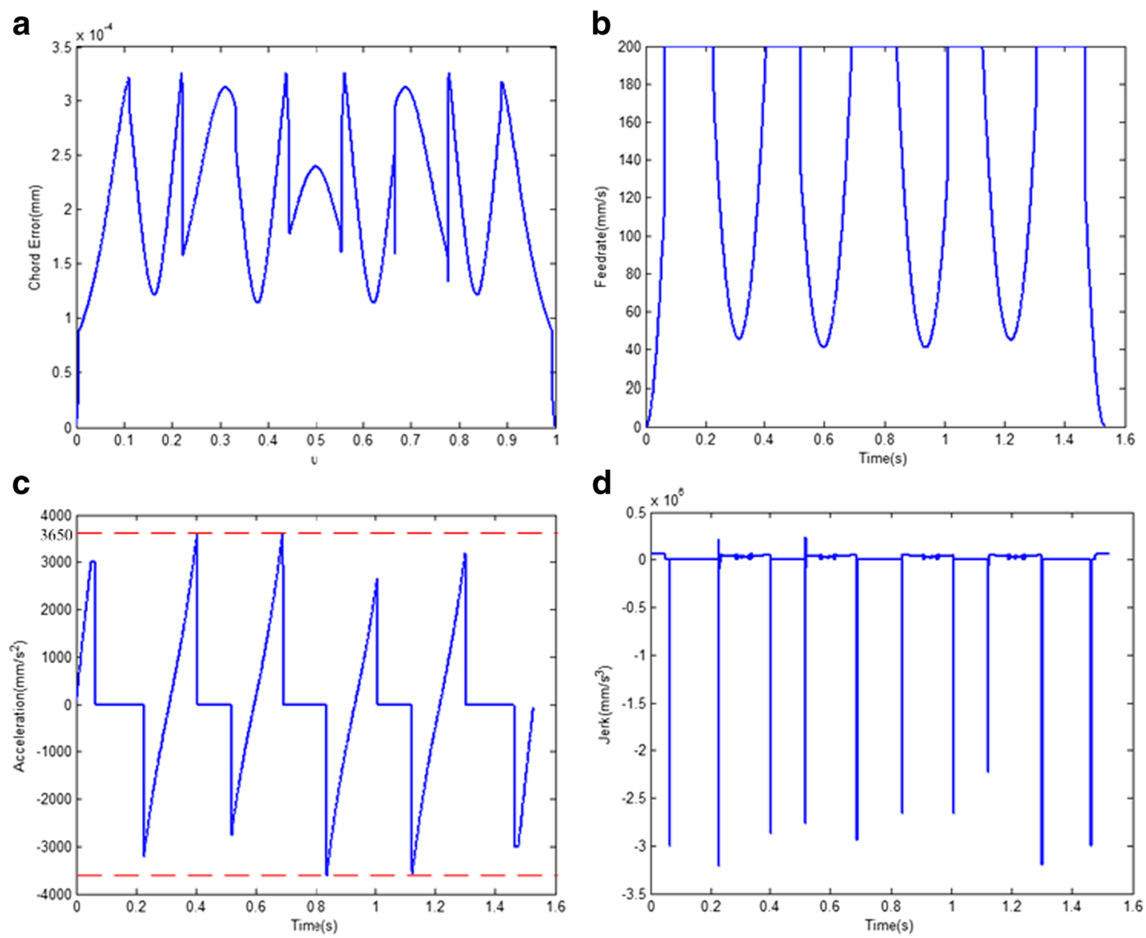

Fig. 6 Chord error (**a**), free date profile (**b**), acceleration profile (**c**), and jerk profile (**d**) by the S-shaped algorithm

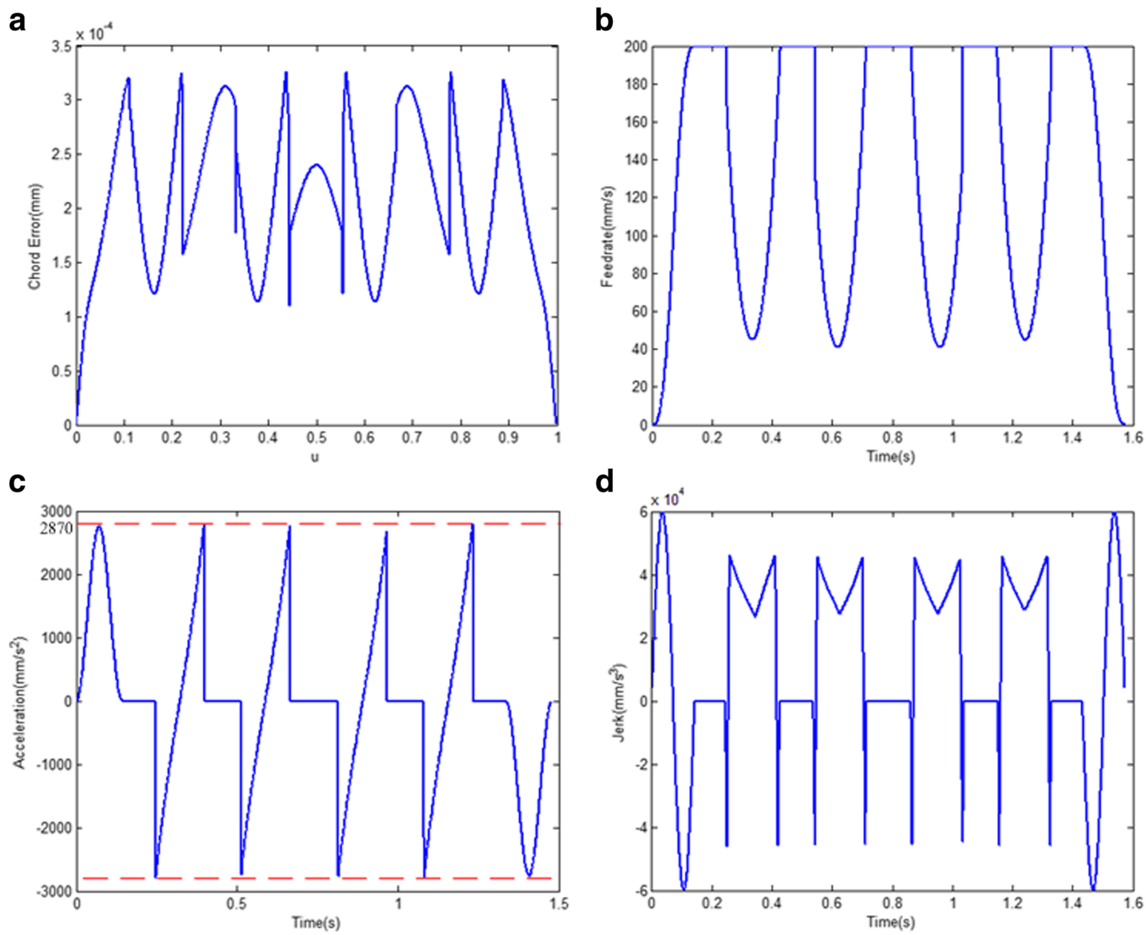

Fig. 7 Chord error (**a**), free date profile (**b**), acceleration profile (**c**), and jerk profile (**d**) by the IAL interpolation algorithm

From Table 2, the proposed IAL interpolation algorithm and the S-shape-based interpolation algorithm reduce the maximum chord error by 67.44 and 66.46 %, respectively. The former algorithm can obtain less chord error than the latter algorithm. Although the machining time of the proposed IAL interpolation algorithm is 0.027 s more than the S-shape-based interpolation algorithm, the gap is very small. From the above, the proposed IAL interpolation algorithm in this paper has better flexibility than the S-shape-based interpolation algorithm.

Table 2 Statistical comparisons of chord error, interpolating points, and machining time

Interpolation algorithm	Chord error (μm)		Interpolating points	Machining time (s)
	MAX	RMS		
S-shaped	0.3354	0.0934	1534	1.534
IAL	0.3256	0.0852	1561	1.561

6 Conclusions and future work

This paper proposed an integrated approach based on the acc-jerk-continuous-based control method and look-ahead algorithm for NURBS interpolation. The NURBS interpolation algorithm is described and the key modules of the proposed approach, such as the feed rate planning module and the acc-jerk-continuous control module, are presented. Finally, an industry application example is provided and several simulation experiments were carried out. Simulation with one free-form NURBS curve is conducted to verify the feasibility and applicability of the proposed algorithm.

In future studies, the authors plan to set up the laboratory table and integrate the proposed algorithm with real-time interpolation to further verify its capability.

Acknowledgments The support of the National Natural Science Foundation of Jiangsu Province (no. BK20151144), Discipline Frontier Research Project (no. 2015XKQY10), and Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) in carrying out this research are gratefully acknowledged.

References

1. Jahanpour J, Alizadeh MR (2014) A novel acc-jerk-limited NURBS interpolation enhanced with an optimized S-shaped quintic feed-rate scheduling scheme. Int J Adv Manuf Technol 77(9–12):1889–1905
2. Wang Y, Yang D, Gai R, Wang S, Sun S (2015) Design of trigonometric velocity scheduling algorithm based on pre-interpolation and look-ahead interpolation. Int J Mach Tools Manuf 96:94–105
3. Nam SH, Yang MY (2004) A study on a generalized parametric interpolator with real-time jerk-limited acceleration. Comput Aided Des 36(1):27–36
4. Tsai MC, Cheng MY, Lin KF, Tsai NC (2005) On acceleration/deceleration before interpolation for CNC motion control. In: ICM'05. IEEE International Conference on Mechatronics, 2005. IEEE, pp 382–387
5. Liu X, Ahmad F, Yamazaki K, Mori M (2005) Adaptive interpolation scheme for NURBS curves with the integration of machining dynamics. Int J Mach Tools Manuf 45(4):433–444
6. Yau HT, Wang JB, Chen WC (2005) Development and implementation for real-time look-ahead interpolator by using Bezier curve to fit CNC continuous short blocks. In: ICM'05. IEEE International Conference on Mechatronics, 2005. IEEE, pp 78–83
7. Lin MT, Tsai MS, Yau HT (2007) Development of a dynamics-based NURBS interpolator with real-time look-ahead algorithm. Int J Mach Tools Manuf 47(15):2246–2262
8. Ye P, Shi C, Yang K, Lv Q (2008) Interpolation of continuous micro line segment trajectories based on look-ahead algorithm in high-speed machining. Int J Adv Manuf Technol 37(9–10):881–897
9. Tsai MS, Nien HW, Yau HT (2008) Development of an integrated look-ahead dynamics-based NURBS interpolator for high precision machinery. Comput Aided Des 40(5):554–66
10. Wang H, Zhao D (2009) Research and implementation of NURBS real-time and look-ahead interpolation algorithm. In: ICMTMA'09. International Conference on Measuring Technology and Mechatronics Automation, 2009. IEEE 3:273–276
11. Emami MM, Arezoo B (2010) A look-ahead command generator with control over trajectory and chord error for NURBS curve with unknown arc length. Comput Aided Des 42(7):625–32
12. Zhang LX, Sun RY, Gao XS, Li HB (2011) High speed interpolation for micro-line trajectory and adaptive real-time look-ahead scheme in CNC machining. Sci China Technol Sci 54(6):1481–1495
13. Wang L, Cao J (2012) A look-ahead and adaptive speed control algorithm for high-speed CNC equipment. Int J Adv Manuf Technol 63(5–8):705–717
14. 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
15. Wang Y, Yang D, Liu Y (2014) A real-time look-ahead interpolation algorithm based on Akima curve fitting. Int J Mach Tools Manuf 85:122–130
16. Jin Y, He Y, Fu J, Lin Z, Gan W (2014) A fine-interpolation-based parametric interpolation method with a novel real-time look-ahead algorithm. Comput Aided Des 55:37–48
17. Qiao Z, Wang H, Liu Z, Wang T (2015) Nanoscale trajectory planning with flexible Acc/Dec and look-ahead method. Int J Adv Manuf Technol 79:1377–1387
18. Yong T, Narayanaswami R (2003) A parametric interpolator with confined chord errors, acceleration and deceleration for NC machining. Comput Aided Des 35(13):1249–1259
19. Park J, Nam S, Yang M (2005) Development of a real-time trajectory generator for NURBS interpolation based on the two-stage interpolation method. Int J Adv Manuf Technol 26(4):359–365
20. Xu RZ, Xie L, Li CX, Du DS (2008) Adaptive parametric interpolation scheme with limited acceleration and jerk values for NC machining. Int J Adv Manuf Technol 36(3–4):343–354
21. Sekar M, Narayanan VN, Yang SH (2008) Design of jerk bounded feed-rate with ripple effect for adaptive NURBS interpolator. Int J Adv Manuf Technol 37(5–6):545–552
22. Wang JB, Yau HT (2009) Real-time NURBS interpolator: application to short linear segments. Int J Adv Manuf Technol 41(11–12):1169–1185
23. Shen H, Fu J, Fan Y (2011) A new adaptive interpolation scheme of NURBS based on axis dynamics. Int J Adv Manuf Technol 56(1–4):215–221
24. Du D, Liu Y, Guo X, Yamazaki K, Fujishima M (2010) An accurate adaptive NURBS curve interpolator with real-time flexible acceleration/deceleration control. Robot Comput Integr Manuf 26(4):273–281
25. Wang X, Wang J, Rao Z (2010) An adaptive parametric interpolator for trajectory planning. Adv Eng Softw 41(2):180–187
26. 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
27. Wang Y, Liu H, Yu S (2012) Curvature-based real-time NURBS surface interpolator with look-ahead ACC/DEC control. Math Comput Sci 6(3):315–326
28. Lee AC, Lin MT, Pan YR, Lin WY (2011) The feed-rate scheduling of NURBS interpolator for CNC machine tools. Comput Aided Des 43(6):612–628
29. Piegl L, Tiller W (1997) The NURBS books, 2nd edn. Springer, Berlin