Int J Adv Manuf Technol (2009) 41:1169–1185
DOI 10.1007/s00170-008-1564-8

ORIGINAL ARTICLE

Real-time NURBS interpolator: application to short linear segments

Jun-Bin Wang · Hong-Tzong Yau

Received: 29 August 2007 / Accepted: 10 May 2008 / Published online: 25 June 2008

Abstract This study proposes the use of a real-time non-uniform rational B-spline (NURBS) interpolator with a look-ahead function to handle numerous short linear segments. The short linear segments conforming to the continuous short block (CSB) criterion can be fitted into NURBS curves in real time. A modified maximum feedrate equation based on the geometric characteristics of the fitting curves and the dynamics of the servo control system has been derived in this paper. Taking advantage of the multi-thread design and the look-ahead function, the real-time NURBS interpolator can process enough G01 block information and complete feedrate planning before interpolation. In addition, the S-shaped jerk-limited acceleration method is adopted for smoother feedrate profiles. Two part shapes, which possess more than 1,000 short linear segments, are tested on our PC-based real-time control system. Both simulation and experimental results verify the feasibility and precision of the proposed interpolation algorithm.

Keywords NURBS · Interpolator · Real-time · Look-ahead · Jerk-limited · Short block

1 Introduction

For high-speed and high-accuracy machining, parametric interpolations are replacing conventional linear and circular interpolations. The commonly used parametric forms include Bezier, B-spline, and non-uniform rational B-spline (NURBS) curves. NURBS curves have two advantages: (1) weighting control can locally change contour profiles, and (2) complicated contour representation does not need to increase the order of equation. NURBS curves have presently become a common and useful parametric form used by computer-aided design (CAD)/computer-aided manufacturing (CAM) systems.

There have been many studies on NURBS interpolation [1–17] (see Fig. 1). The first issue with respect to NURBS interpolation is how to compute successive reference-point parameter values. Farouki and Tsai [1] derived correct co-efficients up to the third order for the Taylor series expansion. Cheng et al. [2] compared several numerical algorithms and suggested that Taylor's second-order approximation was a good choice for real-time NURBS command generators.

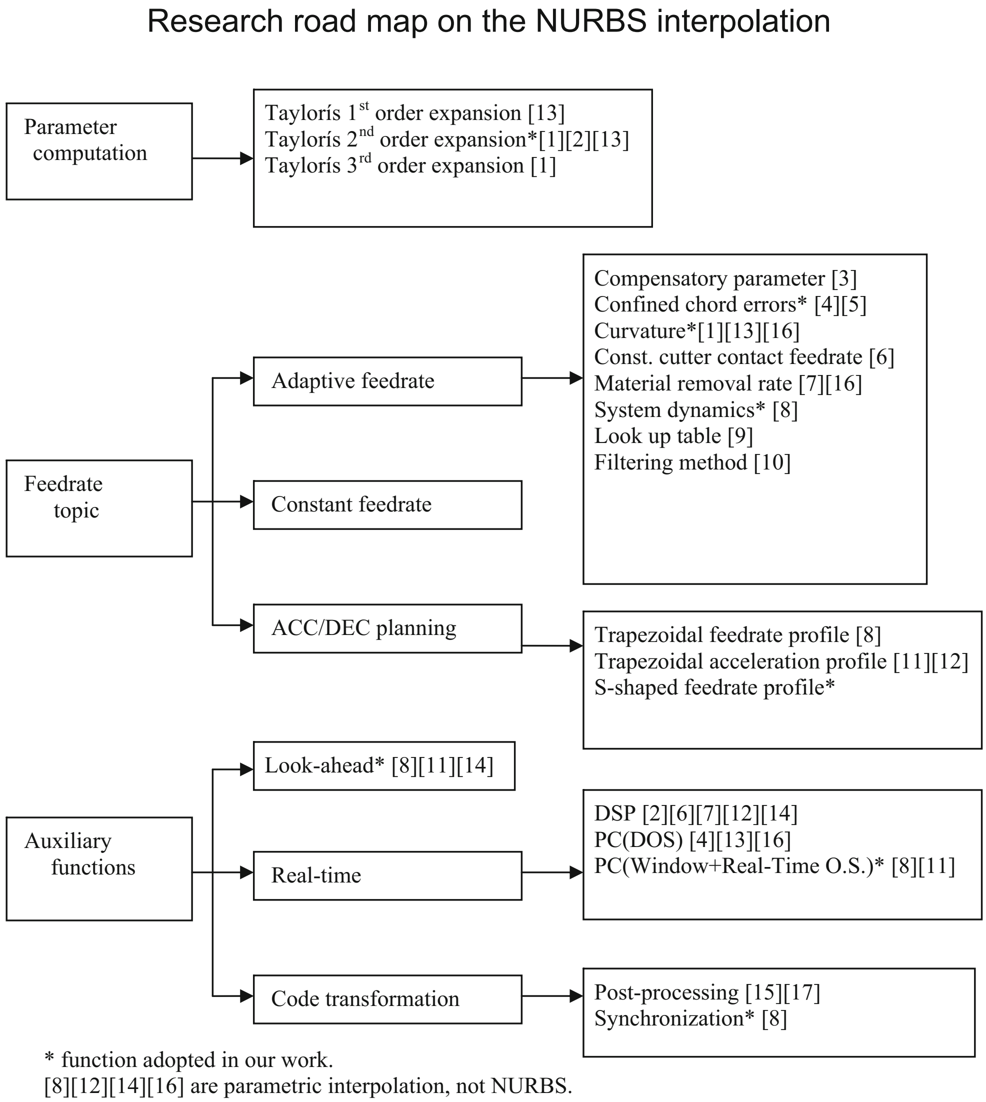
Machining with a constant feedrate can reduce machining profile errors that may result from cutter wear and vibrations. However, over-cutting may happen and lead to loss of accuracy when the cutter moves along high curvature profiles. This brought into consideration the use of adaptive feedrate. The related published works discussed compensatory parameters [3], confined chord errors [4–5], constant cutter contact velocity [6], constant material removal rate [7, 16], and system dynamics [8]. In addition, Yong and Narayanaswami [9] presented a speed-error controlled interpolator, which employed an off-line detection of feedrate sensitive corners. Liu et al. [10] employed a filtering method to smooth the feedrate profile that can reduce feedrate and acceleration fluctuations.

On the other hand, imperfect feedrate profiles reduce the quality of machining because of great jerk amplitudes. Furthermore, exciting the natural modes of the mechanical structure or servo control system incurs serious resonance.

J.-B. Wang (✉) · H.-T. Yau
Department of Mechanical Engineering,
National Chung Cheng University,
168, University Rd., Ming-Hsiung,
Chia-Yi 621 Taiwan, Republic of China
e-mail: jbwang@ms11.hinet.net

H.-T. Yau
e-mail: imehty@ccu.edu.tw

Fig. 1 Previous research issues on NURBS interpolation



Therefore, jerk-limited consideration was adopted in previous studies. The jerk-limited profile generation method by Nam and Yang [11] and the trapezoidal acceleration-based feedrate generation by Erkorkmaz and Altintas [12] are two examples of the use of the jerk-limited acceleration method.

The implementation of the NURBS interpolator relies on a real-time environment, for example, on a DSP [2, 6, 7, 12, 14], or a PC with a real-time operation system [4, 8, 11, 13, 16]. Zhiming et al. [13] showed the CPU time required for one interpolating operation by using Taylor's first- and second-order approximation methods. If the sampling interval of the servo system is 2 ms, it should be feasible to implement this interpolator on-line. Tikhon et al. [7] compared CPU times for NURBS interpolators with different algorithms. The result showed that 4 ms per one interpolation was the upper bound. The same conclusion was given in [6], as well. Therefore, to reduce the interpolation period, a multi-thread design with a look-ahead function was employed in [8] and [11].

In recent years, due to the rapid development of 3D scanning technology, there has been a trend toward creating digital models using point-based or triangulated surface models. Linear numerically controlled (NC) segments are perhaps still the best NC representation form that is used to generate NC tool paths for digital CAD models due to tool path generation speed and accuracy considerations. Consequently, to deal with large amounts of short linear segments, Han et al. [14] presented a high-speed machining algorithm based on the look-ahead interpolation technique for the machining of 3D surfaces using a CAD/CAM system. The proposed algorithm improves the machining speed using a program consisting of short linear segments without any hardware support.

Yau and Kuo [15] took the lead in proposing a post-processing approach to convert G01 NC codes from most

CAD/CAM systems to NURBS NC paths for high-speed contour machining. A NURBS interpolation strategy that took into account an optimized cutting feedrate based on machine dynamic response and curvature of the NURBS curve was also developed. Li et al. [17] proposed a NURBS pre-interpolator for five-axis machining. Its feasibility has been evaluated only from simulation. Ye et al. [18] presented an interpolation of continuous microline segment trajectories based on the look-ahead algorithm. Ye's proposed method, strictly speaking, uses linear interpolation. Due to the superiority of cubic Bezier interpolation for fast computation, Yau et al. [8] proposed a real-time cubic Bezier interpolator with a look-ahead function to deal with a large amount of short linear segments. But the cubic Bezier interpolator still has a C^1 discontinuity property at the interface of the fitted G01 blocks.

Therefore, this paper proposes a real-time NURBS interpolator with a look-ahead function using a PC-based control architecture. To deal with numerous G01 short segments that are created in digital CAD models or that are produced by the linearization of curves, the proposed approach allows short linear segments to be fitted into smoother NURBS curves. During the NC code interpreting stage, the continuous short block (CSB) criterion [8] is employed to identify the CSBs. The CSBs are valid for NURBS interpolation, and the fitted blocks then have the characteristics of C^0, C^1, and C^2 continuity. On the other hand, the S-shaped jerk-limited acceleration method is applied to feedrate planning. It is conducive to restraining excessive jerks and helpful to achieving smoother feedrate profiles. Two NC programs possessing a large number of short G01 blocks are tested to verify that the proposed real-time NURBS interpolator is able to provide satisfactory performance.

2 CSB criterion and look-ahead function

Based on our previous work [8], the CSB criterion is used in this study to check whether short linear segments can be fitted by NURBS curves. The CSB criterion includes both the critical corner angle test and the bi-chord error test. During the look-ahead process, the critical corner angle test picks out short G01 blocks, which have corner angles that are greater than the critical corner angle. Then, the bi-chord error test is taken into consideration before NURBS curves are used to fit such short segments. These continuous short G01 blocks that satisfy the CSB criterion are called CSBs and are valid for NURBS fitting and curve interpolation. Figure 2 shows the flowchart of the CSB criterion [8]. A brief introduction to the CSB criterion is given below.

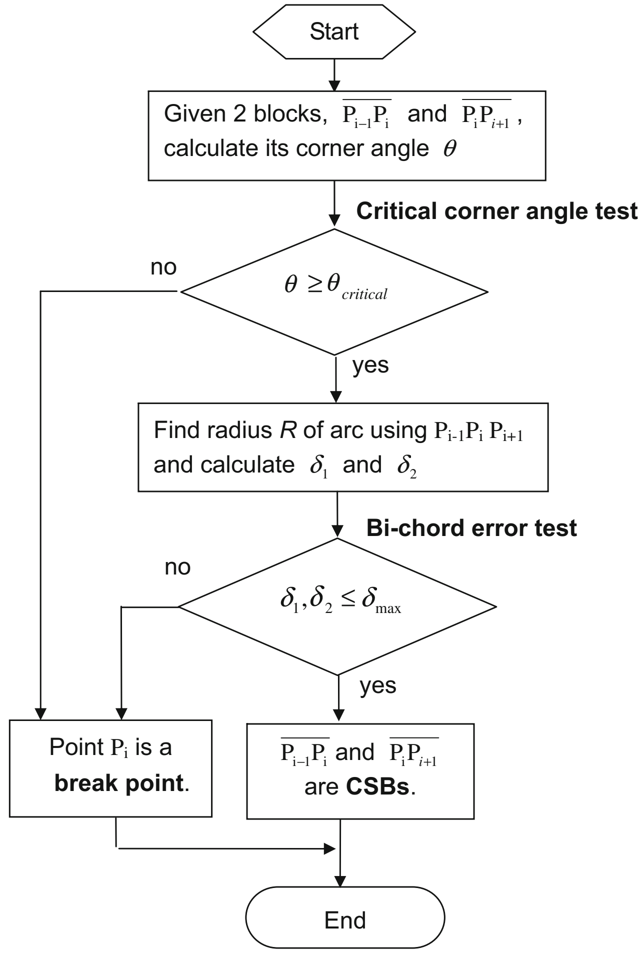


Fig. 2 Flowchart of the CSB criterion [8]

2.1 CSB criterion

A first-order approximation of a feedforward servo system is adopted as the system model shown in Fig. 3. When a tool moves at a feedrate V_{NC} along a G01 block's corner under a permissible corner error ε_{max}, the smallest corner angle, that is the critical corner angle $\theta_{critical}$, can be obtained from the following expression:

$$\theta_{critical} = 2\cos^{-1}\left(\frac{K_p}{1-K_f}\cdot\frac{\varepsilon_{max}}{V_{NC}}\right) \tag{1}$$

where K_f is the feedforward gain and K_p is the position gain. Any G01 blocks forming corner angles that are greater than $\theta_{critical}$ can be treated as blocks that could be fitted by NURBS curves.

To obtain good accuracy, the bi-chord error test must be taken into consideration before NURBS curves are used to fit such short blocks that satisfy the critical corner angle test. As shown in Fig. 4, if both δ_1 and δ_2 are smaller than

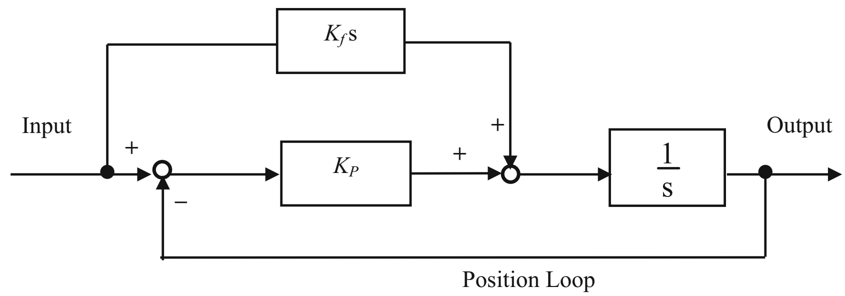


Fig. 3 Block diagram of a first-order approximation of a feed-forward servo system

the designated maximum chord error $\delta_{\max}$, the contour error must be restricted within $\delta_{\max}$ after fitting points P_{i-1}, P_i, and P_{i+1} to a NURBS curve. In this work, the critical corner angle θ_{critical} is 172.8° (using a feedrate of 3,000 mm/min), the maximum corner error $\varepsilon_{\max}$ is 0.01 mm, and the maximum chord error $\delta_{\max}$ is 0.01 mm, as listed in Table 2.

2.2 Maximum feedrate determination of NURBS-fitted blocks

To maintain contour accuracy, the feedrate must be decreased when the cutter moves toward local high-curvature profiles. The critical value of curvature requires significant axis deceleration. Therefore, the first stage in modifying the maximum feedrate is to determine the critical curvature κ_{critical} before feedrate planning. The critical curvature κ_{critical} can be obtained by using

$$\kappa_{\text{critical}} = \frac{a}{V_{\text{NC}}^2} \tag{2}$$

where V_{NC} is the desired feedrate in NC codes and a is the maximum acceleration of the system.

To derive the modified maximum feedrate equation of the NURBS-fitted blocks, whose curvature is larger than the critical curvature κ_{critical}, the first-order approximation of a feed-forward servo system is again adopted as a simplified model. The corresponding circular contouring error equation based on the dynamics of the servo control systems is expressed by

$$\kappa\,\varepsilon_{\text{ss}} = 1 - \sqrt{\frac{K_{\text{f}}^2\omega^2 + K_{\text{p}}^2}{\omega^2 + K_{\text{p}}^2}} \tag{3}$$

where ε_{ss} is the steady state circular contouring error, κ is the curvature, and ω is the angular velocity. Rearranging Eq. 3 gives the formula

$$\omega = K_{\text{p}}\sqrt{\frac{1-(1-\kappa\,\varepsilon_{\text{ss}})^2}{(1-\kappa\,\varepsilon_{\text{ss}})^2 - K_{\text{f}}^2}} \tag{4}$$

The modified maximum feedrate V_k for the NURBS-fitted blocks having high curvature is then determined by

$$V_\kappa = \frac{\omega}{\kappa} \tag{5}$$

By setting ε_{ss} as the tolerance value of the circular contouring error, V_κ is thus obtained as the modified maximum feedrate. Thus, the following equation is proposed to determine the maximum feedrate $V_{\max}$ according to the curvature of the NURBS-fitted blocks.

$$V_{\max} = \begin{cases} V_{\text{NC}}\,, & \text{if } \kappa < \kappa_{\text{critical}} \\ V_\kappa\,, & \text{if } \kappa \geq \kappa_{\text{critical}} \end{cases} \tag{6}$$

2.3 Look-ahead function

To execute G-code interpretation and NURBS interpolation synchronously, a real-time PC-based motion controller is used in this study. Because the functionality of the Windows operating system is not targeted for "hard" real-time applications, VenturCom's RTX product is adopted to satisfy a full range of real-time application requirements. The RTX adds a real-time subsystem, known as RTSS, to the Windows operating system. RTSS is conceptually similar to other Windows subsystems like Win32. Figure 5 shows the system architecture of our PC-based motion controller, in which the G-code interpreter is designed on the Win32 thread and the NURBS interpolator is placed on the RTSS thread. Communication between the Win32 thread and the RTSS thread relies on the shared memory. As long as an interpreter outputs G-code data to the shared

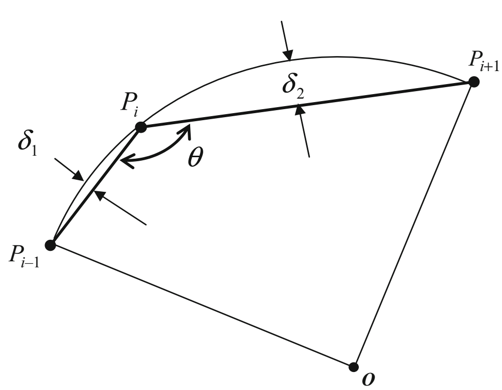


Fig. 4 Schematic diagram of the bi-chord test

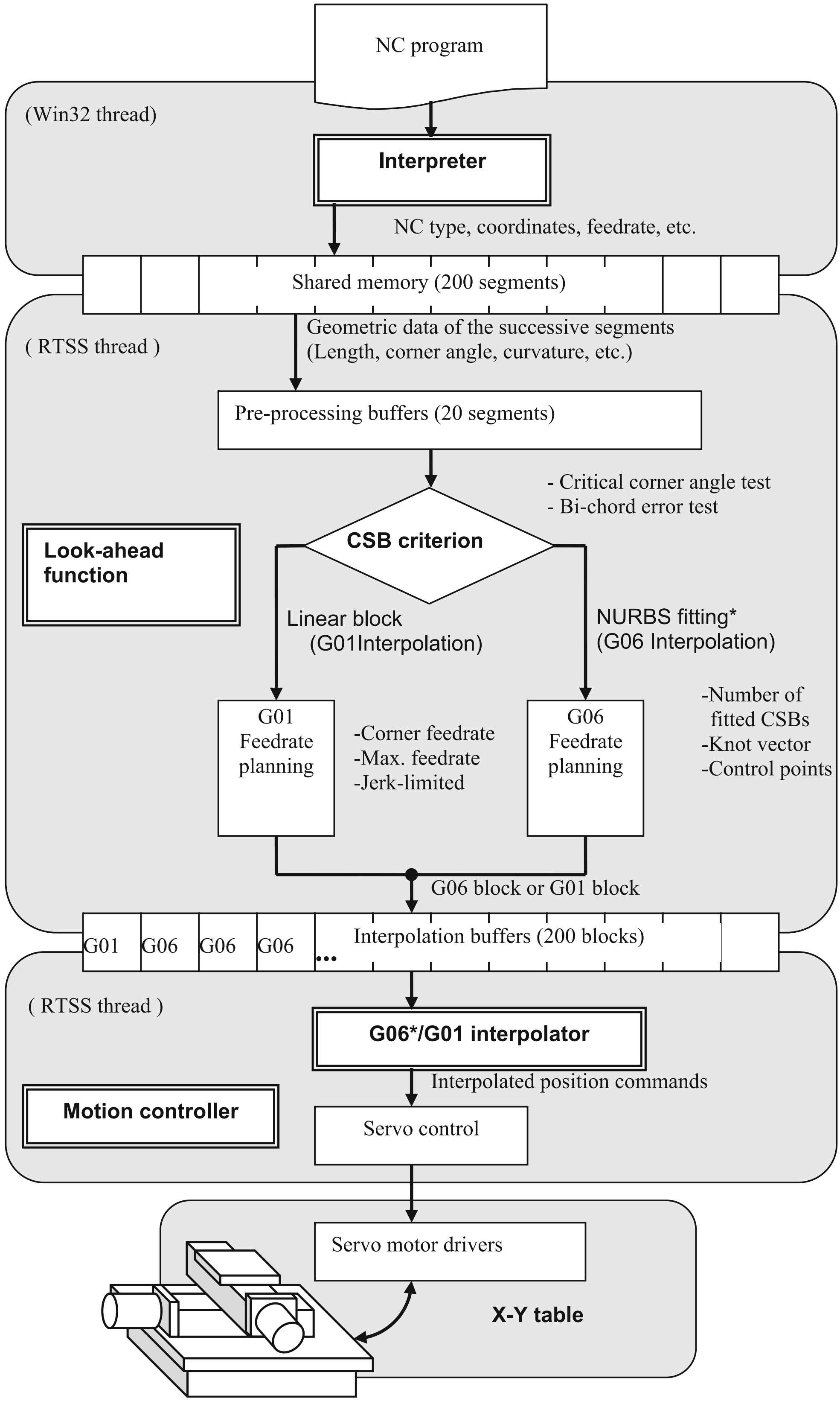


Fig. 5 Architecture of the PC-based real-time NURBS interpolator (a detailed description of the NURBS interpolation is marked with an *asterisk*; see Fig. 10)

memory, the NURBS interpolator moves data from the shared memory and starts to calculate the interpolated position coordinates. In the look-ahead stage, the CSB criterion [8] can quickly determine if the short linear segments belong to G01 interpolation or NURBS interpolation. The results are saved in the interpolation buffers, which have a storage capacity that is designed for up to 200 fitted curves at most.

In this work, NURBS curves with degree 3 are manipulated for curve fitting. The number of fitted segments cannot be less than 3. In addition, the maximum number of fitted segments is 20, taking into consideration PC performance. The flow chart of the NURBS interpolator with the look-ahead function is shown in Fig. 6. The other computations in the look-ahead stage, such as the parameters of NURBS curves and curve feedrate planning, are described in Section 3.

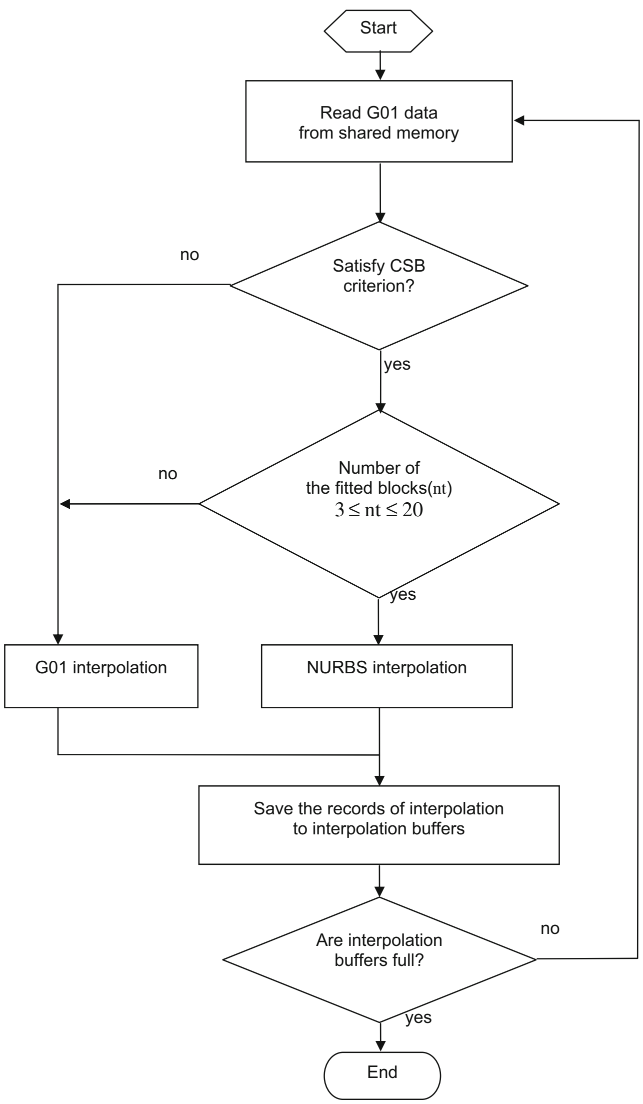


Fig. 6 Flowchart of the look-ahead function that decides between G01 and NURBS interpolation

3 Real-time NURBS interpolator

Relying on manipulating weights, knot vectors, and control points, one can create many kinds of complicated curves and surfaces with NURBS curves. In this research, the CSBs are fitted into NURBS curves and interpolated to produce smoother contours in real-time machining. This section first introduces NURBS curves and then presents a procedure for finding control points. After that, the S-shaped jerk-limited acceleration method is used for feedrate generation. The interpolated feedrate profiles are smoothed with jerk limitation.

3.1 NURBS curves

A pth-degree NURBS curve [19] is defined by

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i P_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} = \sum_{i=0}^{n} R_{i,p}(u) P_i \quad 0 \le u \le 1 \tag{7}$$

where P_i are the control points, w_i are the weights, $N_{i,p}$ is the ith B-spline basis function of the pth-degree defined on the non-periodic knot vector **U**

$$U = \{u_0, u_1, \ldots, u_{n+p+1}\} \tag{8}$$

$N_{i,p}$ is defined by Eqs. 9 and 10 and the $R_{i,p}$, the rational basis functions, are defined by Eq. 11.

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad i = 0, 1, \ldots, n \tag{10}$$

$$R_{i,p}(u) = \frac{N_{i,p}(u) w_i}{\sum_{j=0}^{n} N_{j,p}(u) w_j} \tag{11}$$

Since NURBS curves have to pass through all given CSB coordinates, C_0, C_1,…,C_n, Eq. 7 is re-expressed in matrix form shown in Eq. 12 to determine the control points.

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ R_{0,p}(\bar{u}_1) & R_{1,p}(\bar{u}_1) & R_{2,p}(\bar{u}_1) & \cdots & R_{n,p}(\bar{u}_1) \\ R_{0,p}(\bar{u}_2) & R_{1,p}(\bar{u}_2) & R_{2,p}(\bar{u}_2) & \cdots & R_{n,p}(\bar{u}_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ P_n \end{bmatrix} = \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ \vdots \\ C_n \end{bmatrix} \tag{12}$$

where the parameter $\bar{u}_i$ according to points C_i can be calculated using the chord length parameterization method [19] (see Eq. 13) and the summation of length of the CSBs, L, approximates the length of a NURBS curve (see Eq. 14).

$$\bar{u}_0 = 0,\ \bar{u}_n = 1$$

$$\bar{u}_i = \bar{u}_{i-1} + \frac{\|C_i - C_{i-1}\|}{L} \quad i = 1, \ldots, n-1 \tag{13}$$

$$L = \sum_{i=1}^{n} \|C_i - C_{i-1}\| \tag{14}$$

Equation 12 is a $(n+1)\times(n+1)$ system of linear equations. The following describes in detail how the system of linear equations is solved to give the solution of the control points P_i. First, to reflect the distribution of $\bar{u}_i$, the knots u_i are manipulated using the technique of averaging [19], as follows. Let the knot vector be $\mathbf{U}=\{u_0,u_1,\ldots, u_p, u_{j+p},\ldots, u_{m-p},\ldots,u_m\}$, then

$$u_0 = \ldots = u_p = 0, u_{m-p} = \ldots = u_m = 1 \tag{15}$$

$$u_{j+p} = \frac{1}{p}\sum_{i=j}^{j+p-1} \bar{u}_i \quad ,j = 1, \ldots, n-p$$

After the knot vector $\mathbf{U}$ has been determined, the next step is to derive the rational basis functions $R_{i,p}(\bar{u}_k)$ (see Eq. 11). When the parameter $\bar{u}_k$ is in the knot span $[u_r, u_{r+1})$, at most $p+1$ of the $N_{i,p}(\bar{u}_k)$ is nonzero, namely, the functions $N_{s-p,p}(\bar{u}_k),\ldots, N_{s,p}(\bar{u}_k)$. Here, s is the span index and $s=r-p$. Hence, $R_{i,p}(\bar{u}_k)$ can be easily calculated, and a $(n+1)\times(n+1)$ square matrix can be formed. This process is illustrated by a triangular scheme shown in Fig. 7, with degree $p=3$ and knot span $u_r \le \bar{u}_k < u_{r+1}$.

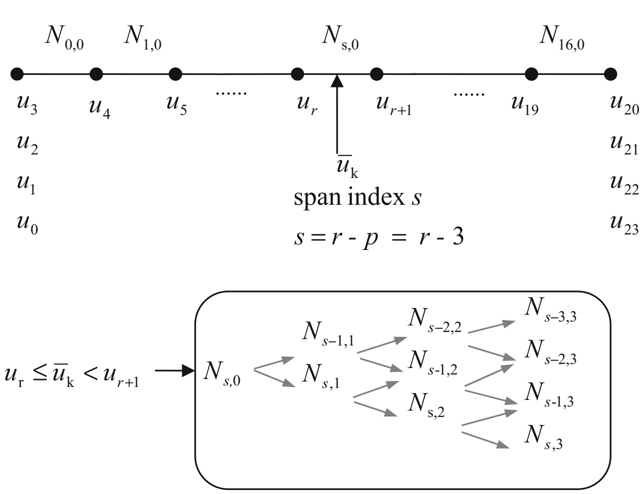


Fig. 7 The triangular scheme in our method: finding nonzero, when n=19 (20 control points) and degree p=3

The matrix formed by the rational basis functions $R_{i,p}(\bar{u}_k)$ is manipulated using the Gauss elimination method. Therefore, the control points P_i can be acquired. So far, one has obtained the control points P and knot vector $\mathbf{U}$ of the NURBS curves.

3.2 S-shaped feedrate profile for ACC/DEC planning

The major function of the look-ahead algorithm is the generation of feedrate profiles. Some important data, like initial and final feedrates, must be decided before feedrate planning. Yau et al. [8] adopted Eq. 1 to re-express the corner feedrate V as Eq. 16, from which the corner feedrate V for different corner angles θ under an acceptable corner error $\varepsilon_{\max}$ is obtained.

$$V = \frac{K_p}{1 - K_f} \cdot \frac{\varepsilon_{\max}}{\cos\left(\frac{\theta}{2}\right)} \tag{16}$$

In other words, the look-ahead algorithm records each block's initial and final feedrates (including linear blocks and NURBS blocks) in the look-ahead buffers before beginning to generate feedrate profiles. Three successive blocks in the look-ahead buffers are used to display the NURBS fitting, as illustrated in Fig. 8, where blocks 1 and 3 are NURBS blocks and block 2 is a linear block. That is to say, NURBS curves are employed to fit the CSBs P_1~P_7 and P_8~P_n, and the line segment P_7P_8 adopts linear G01 interpolation. The feedrate profiles of these three blocks, after the use of the S-shaped jerk-limited acceleration method, are as shown in Fig. 9. The following equations, Eqs. 17, 18, 19, 20, and 21, represent the feedrate profiles in each region in block 1.

$$V(t) = V_s + \frac{1}{2}J_{\max}t^2, \qquad (t_0 \le t \le t_1) \tag{17}$$

$$V(t) = V_1 + A_{\text{ref}}(t - t_1) - \frac{1}{2}J_{\max}(t - t_1)^2, \quad (t_1 < t \le t_2) \tag{18}$$

$$V(t) = V_2 = V_{\text{NC}}, \qquad (t_2 < t \le t_3) \tag{19}$$

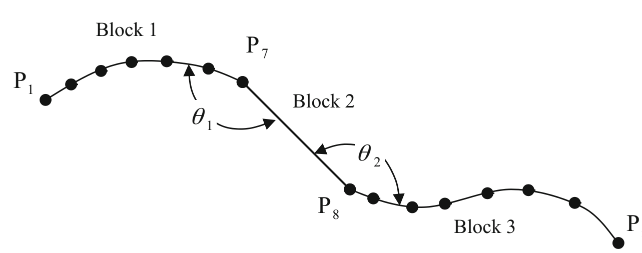


Fig. 8 Three successive blocks: Block1 and block 3 are NURBS blocks. Block2 is a G01 block

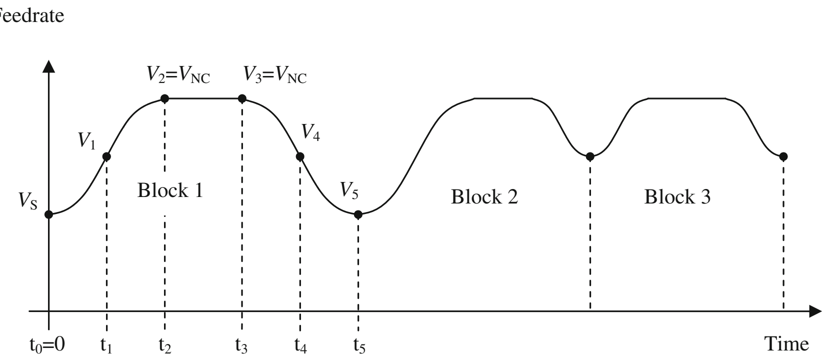


Fig. 9 S-shaped jerk-limited acceleration method for feedrate planning

$$V(t) = V_{\mathrm{NC}} - \frac{1}{2}J_{\max}(t-t_3)^2, \qquad (t_3 < t \le t_4) \tag{20}$$

$$V(t) = V_4 - D_{\mathrm{ref}}(t-t_4) + \frac{1}{2}J_{\max}(t-t_4)^2, \quad (t_4 < t \le t_5) \tag{21}$$

where V_s is the initial feedrate of the block, V_{NC} is the assigned maximum feedrate, $J_{\max}$ is the maximum jerk, and A_{ref} and D_{ref} are the system maximum acceleration and deceleration, respectively.

3.3 Real-time NURBS interpolation scheme

The real-time NURBS interpolation scheme proposed in this paper is schematized in Fig. 10. There are two major processes in the look-ahead function and motion controller. The first process determines the geometrical parameters of the NURBS curves, including the knot vector **U** and the control points P. This is introduced in Section 3.1. We now show the NURBS interpolation process in the motion controller to accomplish the aim of this work. First, the definition of NURBS curve is rearranged to that given in Eq. 22:

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i P_i}{\sum_{j=0}^{n} N_{j,p}(u) w_j} = \frac{A(u)}{w(u)} \tag{22}$$

The value of the parameter u_{i+1} can be obtained using the Taylor's second-order expansion [1] for interpolating the (i+1)th point on NURBS curves. The following second-order approximation is used to calculate the parameter u_{i+1}.

$$u_{i+1} = u_i + \frac{V(u_i)}{|C'(u_i)|}\cdot T_s + \frac{1}{|C'(u_i)|} \cdot \left(A(u_i) - \frac{C'(u_i)\cdot C''(u_i)}{|C'(u_i)|^3}\cdot V(u_i)^2 \right) \cdot \frac{T_s^2}{2} \tag{23}$$

where the mth derivative of a NURBS curve is given as

$$C^{(m)}(u) = \frac{d^{(m)}C(u)}{du^{(m)}} = \frac{A^{(m)}(u) - \sum_{i=1}^{m} \frac{m!}{i!(m-i)!}\cdot w^{(i)}(u)\cdot C^{(m-i)}(u)}{w(u)} \tag{24}$$

and the first and second derivatives of the NURBS curves are as follows:

$$C'(u) = \frac{A'(u) - w'(u)C(u)}{w(u)} \tag{25}$$

$$C''(u) = \frac{A''(u) - 2w'(u)C'(u) - w''(u)C(u)}{w(u)} \tag{26}$$

In Eqs. 25 and 26, $A'(u)$, $w'(u)$, $A''(u)$, and $w''(u)$ can be treated as the derivatives of the B-spline curves (see Eq. 22). Thus, let $B^{(m)}(u)$ denote the mth derivative of a B-spline curve and let $B^{(m)}(u)$ have the form

$$B^{(m)}(u) = \sum_{i=0}^{n-m} N_{i,p-m}(u) P_i^{(m)} = \sum_{i=s-p+m}^{s} N_{i,p-m}(u) P_i^{(m)} \tag{27}$$

where s is the knot span index where u is located, and the values of $P_i^{(m)}$ are recursively calculated as follows:

$$P_i^{(m)} = \begin{cases} P_i, & m = 0 \\ \frac{p-m+1}{u_{i+p+1}-u_{i+m}}\left(P_{i+1}^{(m-1)} - P_i^{(m-1)}\right), & m \ge 1 \end{cases} \tag{28}$$

As shown in Fig. 11, let m=1 and 2. If the derived $P_i^{(m)}$ are put into the Cox–de Boor algorithm, the first and second derivatives of the B-spline curves, $B'(u)$ and $B''(u)$, will be acquired. Using the Cox–de Boor algorithm, we can quickly calculate the derivatives of the NURBS curves in

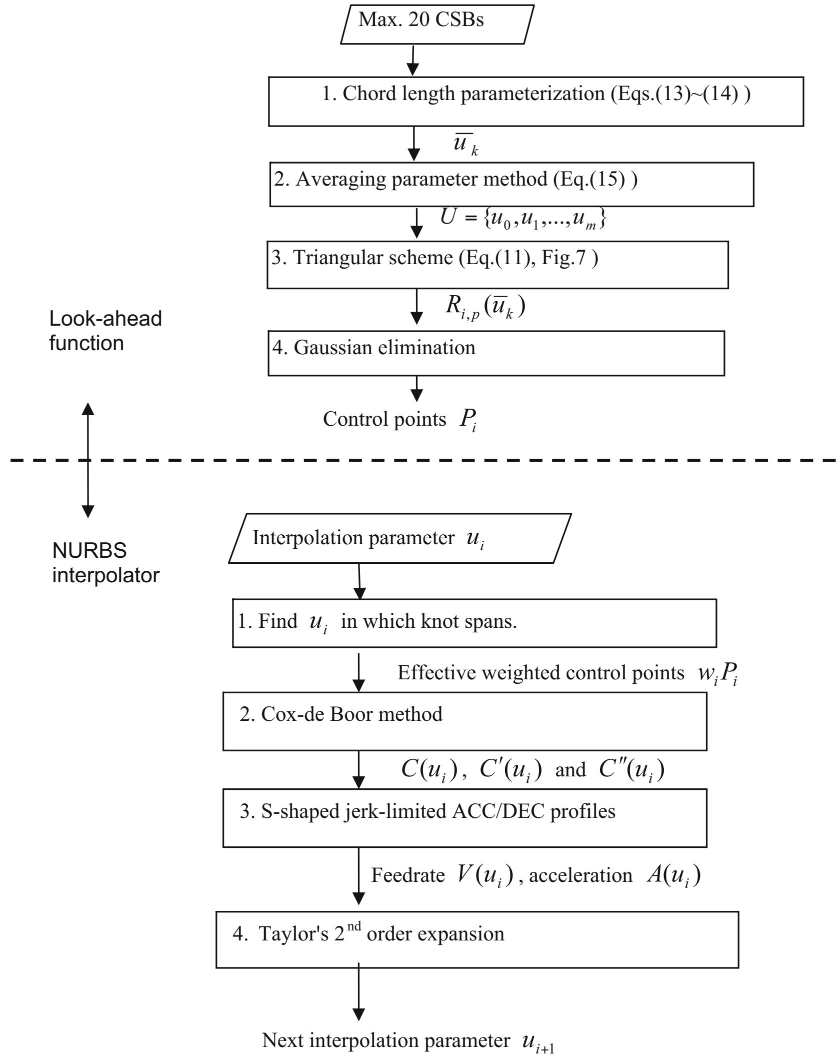


Fig. 10 Real-time NURBS interpolation scheme

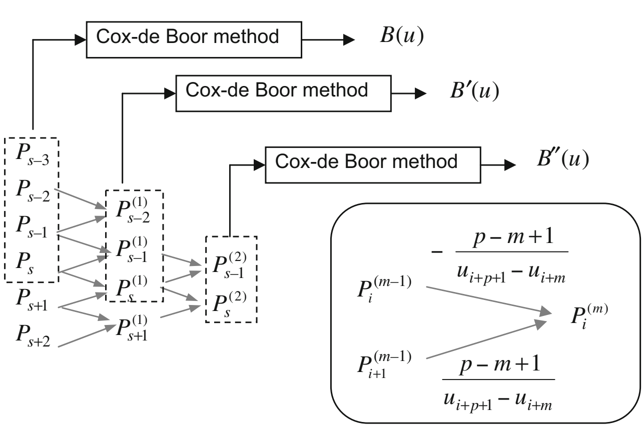


Fig. 11 Use the Cox–de Boor method to find the derivatives of the B-spline curves (u in the sth span and degree p=3)

Eqs. 25 and 26. The procedure of the NURBS interpolator algorithm is summarized as follows:

1. Determine the parameter u_i and the location which spans $u_r \leq u_i < u_{r+1}$
2. Calculate the interpolated point $C(u_i)$ and its derivatives, $C'(u_i)$ and $C''(u_i)$, using the Cox–de Boor method
3. Determine the feedrate $V(u_i)$ and acceleration $A(u_i)$ from S-shaped jerk-limited ACC/DEC profiles.
4. Update the parameter u_{i+1} using Taylor's second-order expansion.

Table 1 Data statistics of the NC programs in the simulations and experiments

Cases	Number of G01 blocks	Max. length (mm)	Min. length (mm)	Mean length (mm)
Oxalis	1,006	0.580	0.091	0.262
Dolphin	1,221	0.743	0.106	0.329

Table 2 Parameters values for the simulations and experiments

Field	Velocity planning				CSB criterion		
Item	Feedrate (mm/min)	A_{max} (mm/s^2)	J_{max} (mm/s^3)	Circular contouring error(mm)	Corner error (mm)	Critical corner angle (degree)	Chord error (mm)
Value	3,000	2,450	50,000	0.005	0.01	172.8	0.01

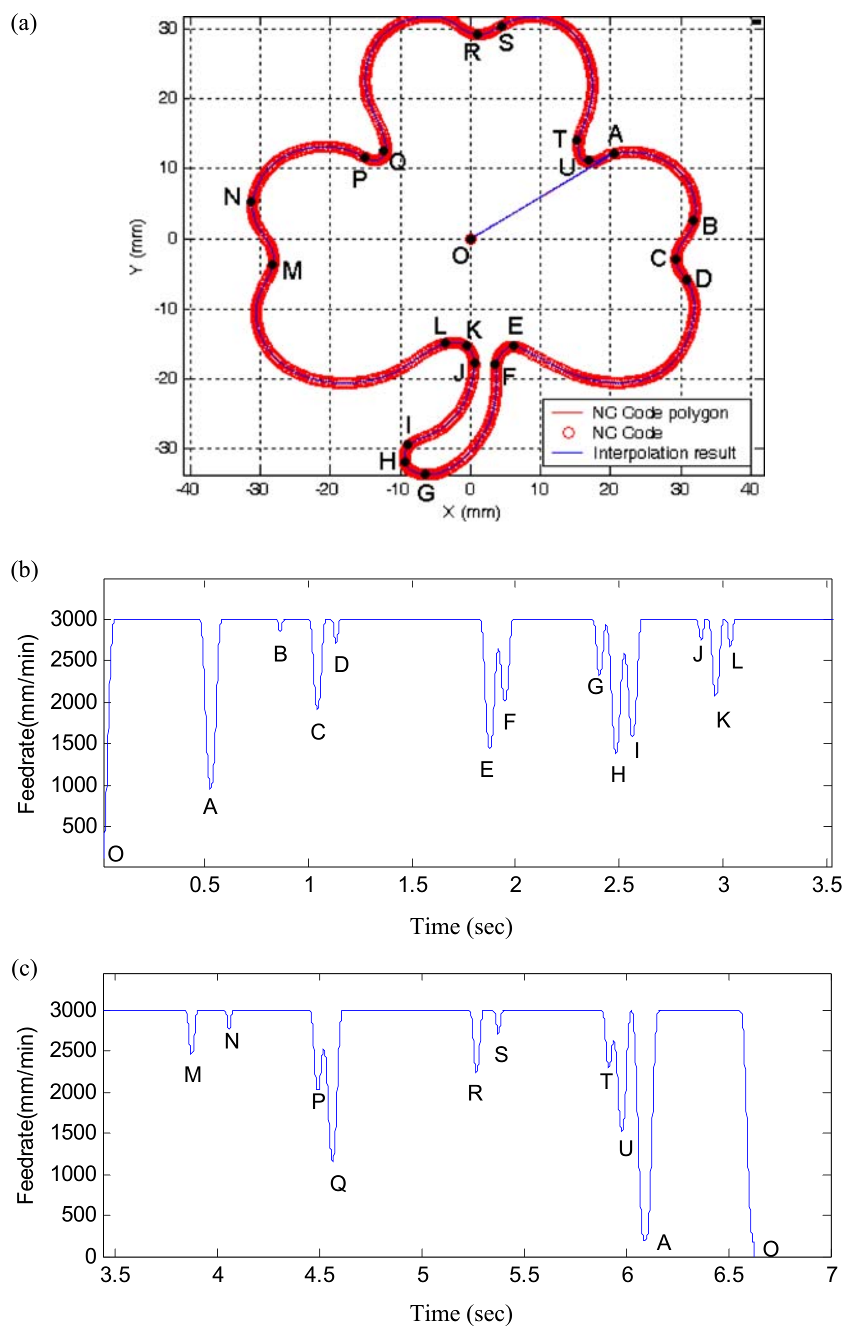


Fig. 12 Oxalis: **a** contour profiles, **b** and **c** feedrate profiles

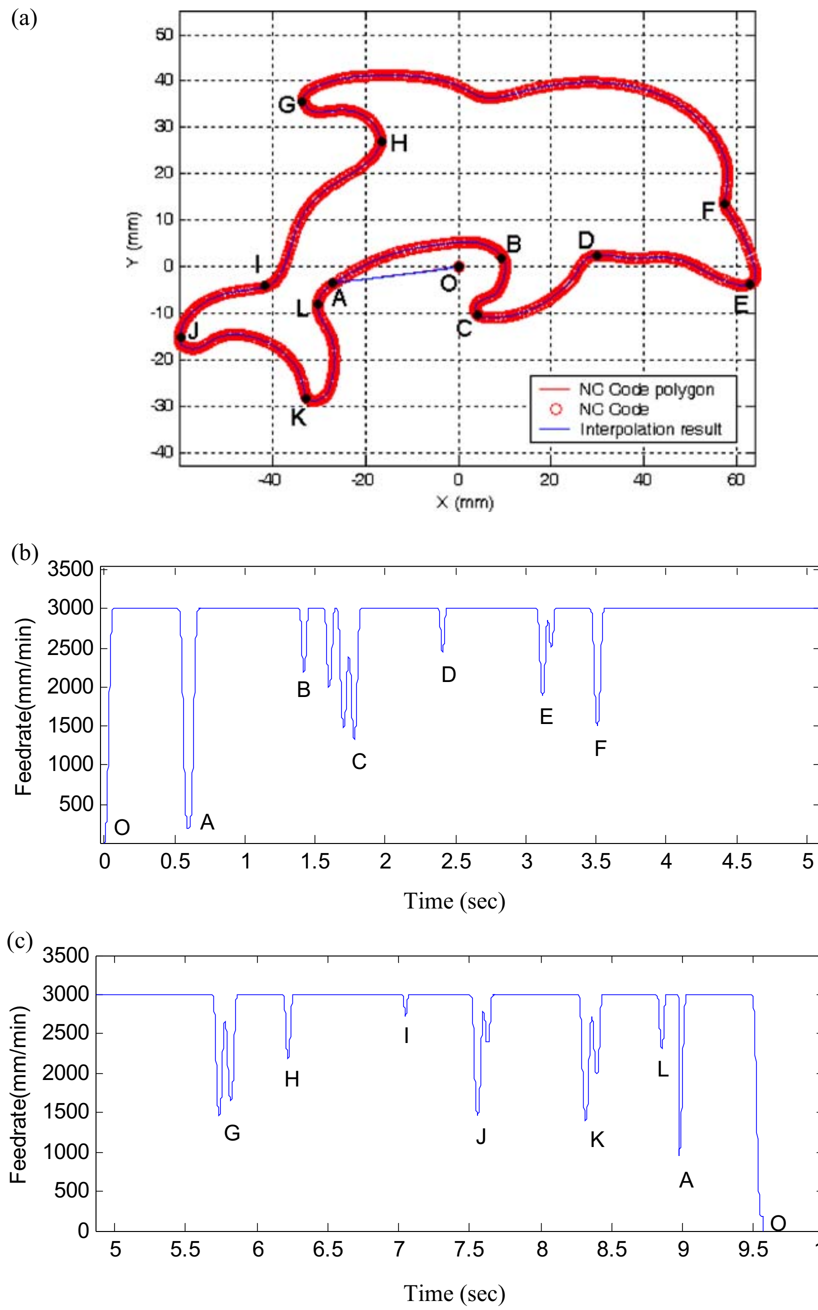


Fig. 13 Dolphin: **a** contour profiles, **b** and **c** feedrate profiles

Table 3 Simulation results: compression factor and contour errors using a feedrate of 3,000 mm/min

Classification	NURBS-fitting			Contour Errors	
	Before fitting (blocks)	After fitting (blocks)	Compression factor	Maximum (μm)	RMS (μm)
Oxalis	1,006	53	18.98	3.66	0.66
Dolphin	1,221	65	18.78	6.84	0.65

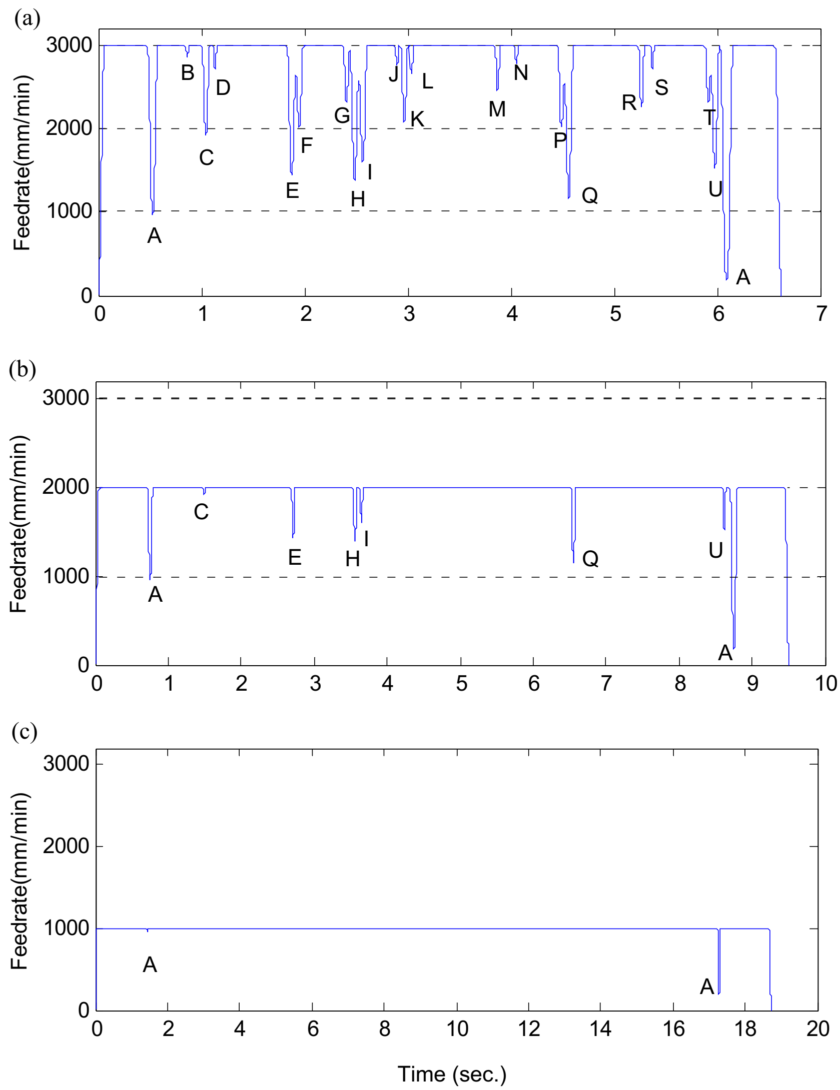


Fig. 14 Oxalis feedrate profiles: **a** 3,000 mm/min, **b** 2,000 mm/min, and **c** 1,000 mm/min

Repeat steps 1 to 4 until $u_{i+1} \geq u_{\max}$, where $u_{\max} = \max\{u_j, j=0,1,\ldots,n+p+1\}$

4 Simulation and experimental results

In our research, the real-time NURBS interpolator is implemented using Visual C and is executed on a real-time Windows operating system (Windows 2000+RTX). The system's hardware includes a Pentium 4 desktop PC with a built-in Adventech PCI-1716 AD/DA card, a PCI-1784 decoder card, and a XY-Table that is driven by Yaskawa servo motors and drivers. Two-part shapes, an oxalis-shaped object and a dolphin-shaped object, which possess above 1,000 of G01 codes, are chosen for simulations and experiments. Further data statistics are listed in Table 1. Simulation parameters, such as fitting feedrate, acceleration, jerk-limited acceleration, and conditions of the CSB criterion, are given in Table 2.

Three observations on the simulations are made: (1) the performance of the real-time NURBS interpolator fitting a large number of CSBs, (2) the output feedrate profiles, and (3) the accuracy of the interpolated position commands. Figures 12 and 13 show the simulation results for the NC programs given in Table 1. The fitting performance of the real-time NURBS interpolator is listed in Table 3. After the NURBS-fitting process, the number of blocks of the oxalis-shaped object is reduced from 1,006 to 53 and that of the dolphin-shaped object from 1,221 to 65. We define the compression factor as the ratio of the number of blocks before NURBS-fitting to the number of blocks after NURBS-fitting. Both compression factors of these two

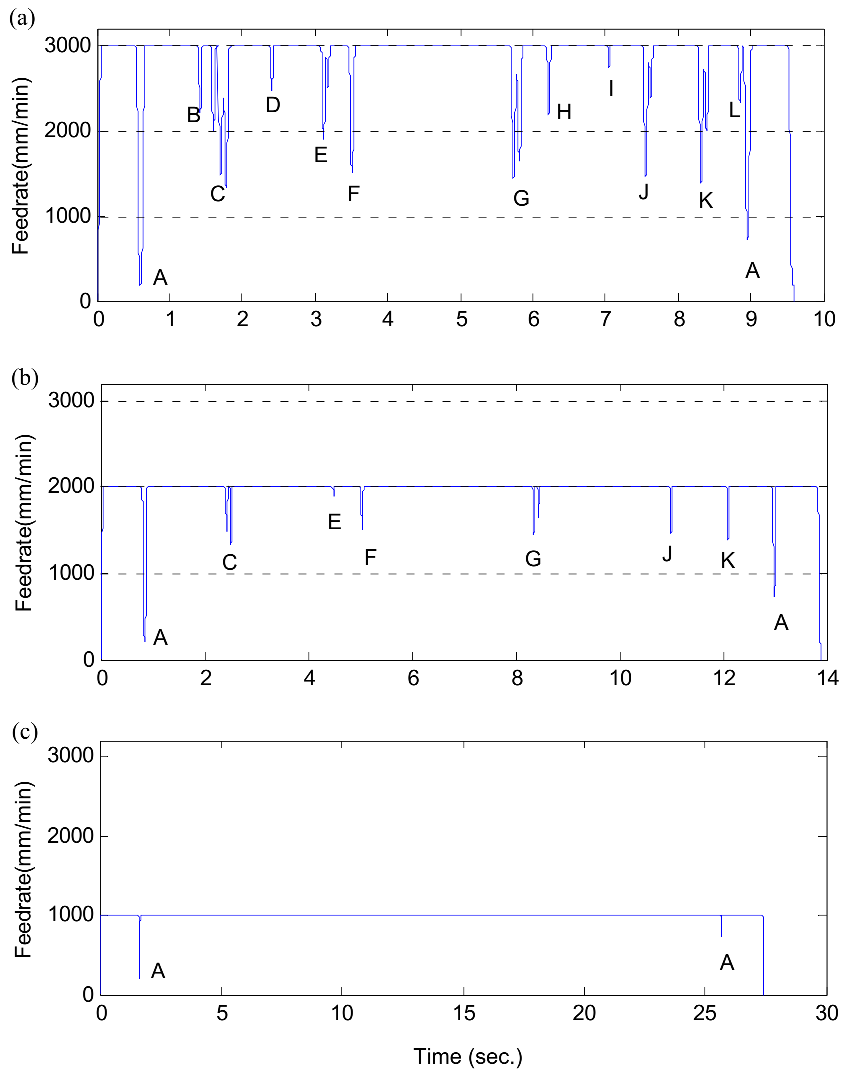


Fig. 15 Dolphin feedrate profiles: **a** 3,000 mm/min, **b** 2000 mm/min, and **c** 1,000 mm/min

cases are about 19. The results verify the effectiveness of the real-time NURBS interpolator on handling short linear segments, which results in smoother contours of the parts.

Figures 12 and 13 also show feedrate sensitive corners at the feedrate of 3,000 mm/min. The English alphabet marks the feedrate sensitive corners in sequence and shows the motion direction of the cutter. It can be seen that the real-time NURBS interpolator does well on these high curvature regions; that is, it decreases feedrate to improve contour precision. By magnifying the portion of violent changes in the feedrate profile, the S-sharped jerk-limited acceleration method makes the feedrate profiles smoother.

The contour error indicating the deviation between the interpolated position commands and the NC code polygon is also listed in Table 3. This indicates the performance of the NURBS interpolator in terms of accuracy. Uncertain factors like mechanism design and control algorithm are eliminated. The maximum contour error of the oxalis-shaped object is 3.66 μm and that of the dolphin-shaped object is 6.84 μm when the feedrate is 3,000 mm/min. The RMS of the oxalis-shaped object is 0.66 μm and that of the dolphin-shaped object is 0.65 μm. This verifies that the contour error is constrained within the desired 10 μm tolerance that is assigned to the CSB criterion. The simulation results show that the proposed NURBS algorithm has feasibility for NURBS interpolation in terms of high-accuracy machining.

Finally, a servo XY-Table system is used to evaluate the feasibility of the implementation of the real-time NURBS interpolator. Two different 2D models used previously in simulations are adopted again in this study (see Figs. 12 and 13). In both cases, the XY-Table is driven using three feedrate commands of 1,000, 2,000, and 3,000 (see Figs. 14 and 15). Experimental results on contour errors and tracking errors are shown in Figs. 16 and 17, respectively.

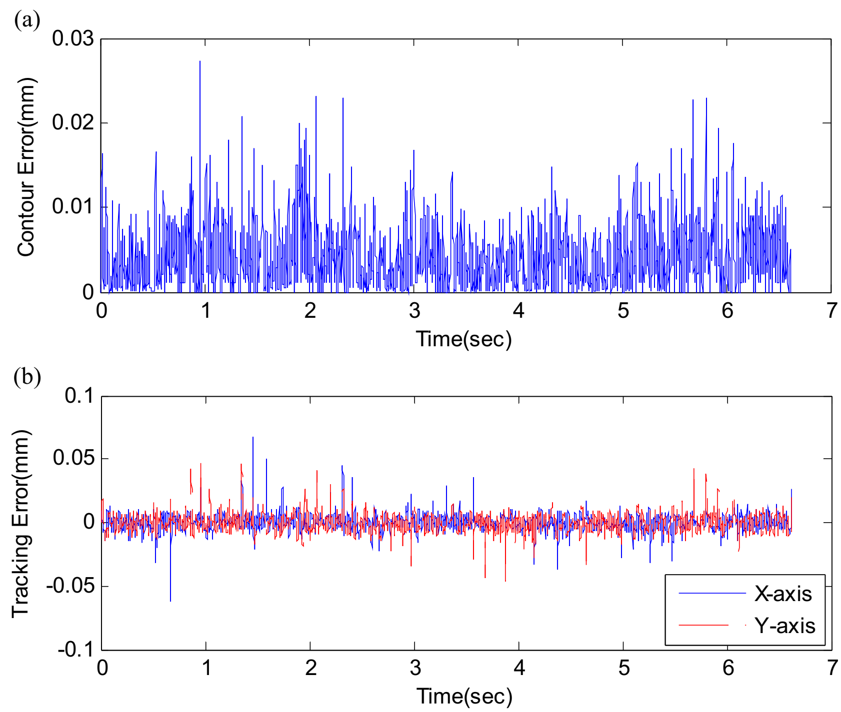


Fig. 16 Oxalis experimental results: **a** contour errors and **b** tracking errors (feedrate= 3,000 mm/min)

The details, including the maximum value and RMS, are summarized in Tables 4 and 5. It is reasonable to anticipate that precision is decreased when the feedrate is 3,000 mm/min. However, both the maximum contour errors are merely within 27.3 μm, and the RMS is smaller than 6.0 μm.

To demonstrate the performance of the proposed algorithm in dealing with the short G01 blocks, a contrast experiment is performed to show the differences between real-time NURBS interpolation and Fast Bezier interpolation [8]. For certain conditions, like the examples used in

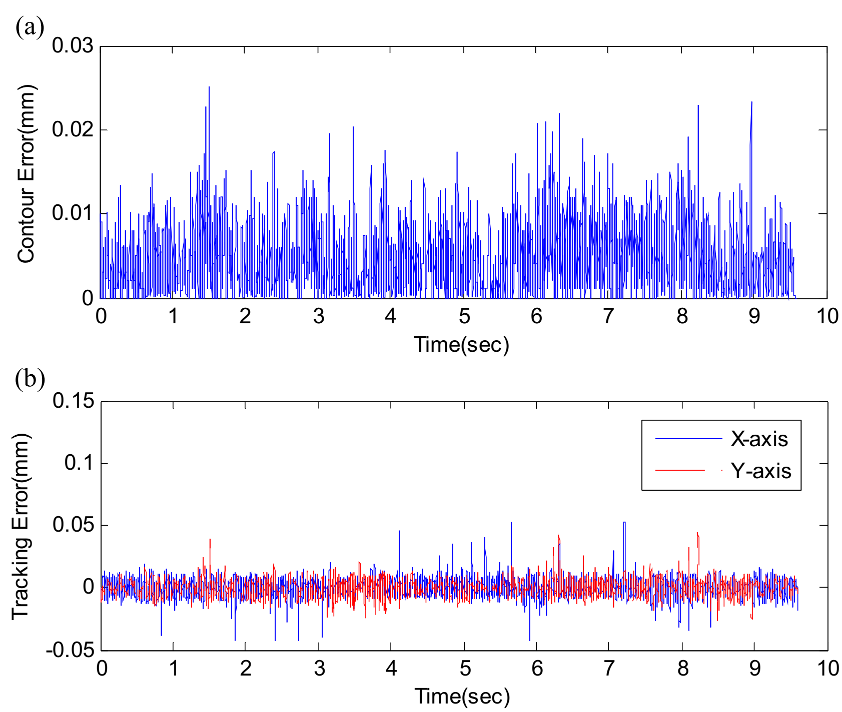


Fig. 17 Dolphin experimental results: **a** contour errors and **b** tracking errors (feedrate= 3,000 mm/min)

Table 4 Experimental results of the Oxalis

	Feedrate (mm/min)		Contour errors (μm)	Tracking errors(μm)		Machining time (s)
				X-axis	*Y*-axis	
NURBS Interpolator	1,000	Max.	15.8	25.0	45.0	18.715
		RMS	4.7	4.6	5.6	
	2,000	Max.	19.0	35.7	51.9	9.509
		RMS	5.4	5.1	6.5	
	3,000	Max.	27.3	68.0	47.4	6.624
		RMS	5.6	6.1	7.4	
Fast Bezier Interpolator	3,000	Max.	38.0	77.0	51.7	6.303
		RMS	6.1	6.0	6.8	

this study, the bulk of the short linear blocks satisfy the CSB criterion. However, for high curvature profiles, Fast Bezier interpolation, which does not modify the maximum feedrate, still maintains the NC assigned feedrate through the Bezier-fitted blocks. This causes larger contour errors than when real-time NURBS interpolation is used (see Figs. 18 and 19 for the use of a feedrate of 3,000 mm/min). Detailed comparisons are also given in Tables 4 and 5.

5 Conclusions

Using parametric forms to describe the contours of workpieces in the fields of CAD/CAM and CNC machining is becoming a trend. However, due to the rapid development of 3D scanning technology that employs short linear segments to generate NC tool paths for digital CAD models, processing a large amount of short linear segments is still regarded as a challenging issue.

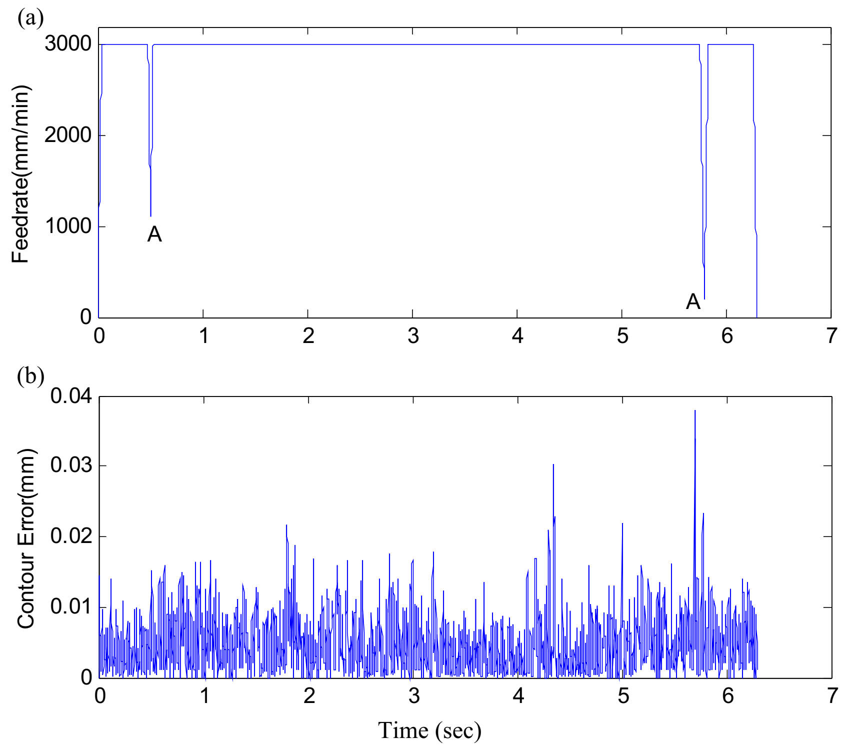
This paper proposed a convincing solution, a real-time NURBS interpolator with a look-ahead function, to solve the problem of processing short linear segments for NC interpolation. Short linear segments conforming to the CSB criterion can be fitted into NURBS curves in real-time. Taking advantage of the multi-threads design and the look-ahead function, the NURBS interpolator can process enough NC block information to complete feedrate planning before interpolation. Consequently, the computational load is controlled successfully in an interpolation period and unnecessary feedrate acceleration and deceleration can be avoided. The proposed approach allows the NURBS-fitted segments to have C^2 continuity. As a result, smoother machining and better surface quality can be obtained.

Two different 2D NC programs possessing thousands of short G01 blocks were tested on an in-house developed XY-Table with a PC-based motion controller. The simulation and experimental results show that the feedrate profiles reflect the geometric characteristics of the models and that the output contour profiles approach the input NC program models when compared with Fast Bezier interpolation. This indicates that the proposed real-time NURBS interpolator is able to provide satisfactory performance.

Table 5 Experimental results of the Dolphin

	Feedrate (mm/min)		Contour errors (μm)	Tracking errors(μm)		Machining time (s)
				X-axis	*Y*-axis	
NURBS Interpolator	1,000	Max.	16.0	23.0	25.0	27.416
		RMS	4.2	4.7	5.0	
	2,000	Max.	20.4	42.0	36.9	13.880
		RMS	5.5	6.2	6.1	
	3,000	Max.	25.1	53.0	44.7	9.592
		RMS	6.0	7.1	6.9	
Fast Bezier Interpolator	3,000	Max.	30.0	68.0	35.5	9.201
		RMS	5.5	5.8	5.9	

Fig. 18 Oxalis experimental results from using Fast Bezier Interpolator: **a** feedrate commands and **b** contour errors (feedrate=3,000 mm/min)



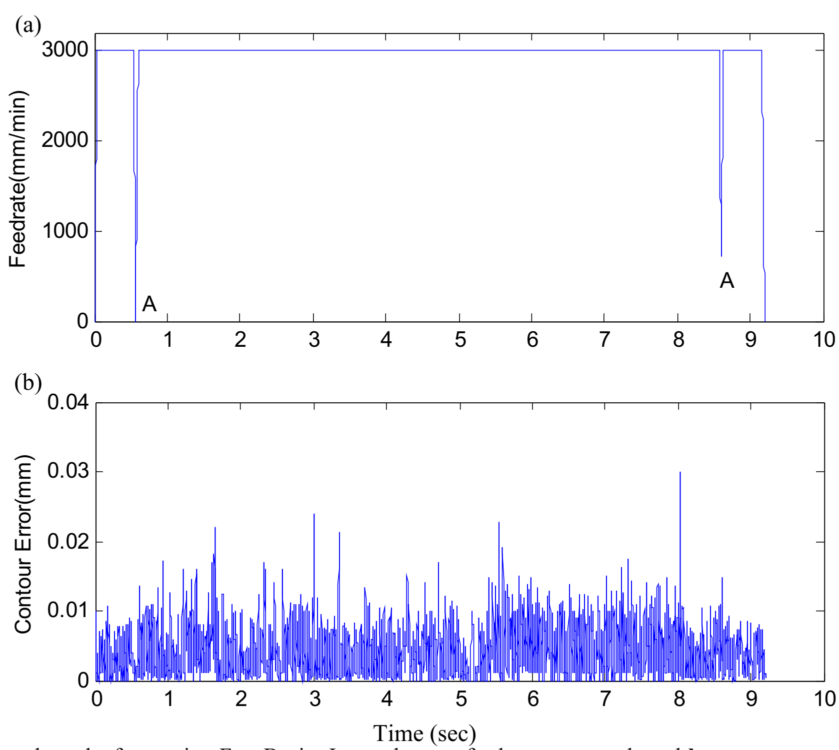


Fig. 19 Dolphin experimental results from using Fast Bezier Interpolator: **a** feedrate commands and **b** contour errors (feedrate=3,000 mm/min)

References

1. Farouki RT, Tsai Y-E (2001) Exact Taylor series coefficients for variable-feedrate CNC curve interpolators. Comput-Aided Des 33:155–165
2. Cheng M-Y, Tsai M-C, Kuo J-C (2002) Real-time NURBS command generators for CNC servo controllers. Int J Mach Tools Manuf 42:801–813
3. Yeh S-S, Hsu P-L (1999) The speed-controlled interpolator for machining parametric curves. Comput-Aided Des 31:349–357
4. Yeh S-S, Hsu P-L (2002) Adaptive-feedrate interpolation for parametric curves with a confined chord error. Comput-Aided Des 34:229–237
5. Du D, Liu Y, Yan C, Li C (2007) An accurate adaptive parametric curve interpolator for NURBS curve interpolation. Int J Adv Manuf Technol 32:999–1008
6. Tsai M-C, Cheng C-W, Cheng M-Y (2003) A real-time NURBS surface interpolator for precision three-axis CNC machining. Int J Mach Tools Manuf 43:1217–1227
7. Tikhon M, Ko TJ, Lee SH, Kim HS (2004) NURBS interpolator for constant material removal rate in open NC machine tools. Int J Mach Tools Manuf 44:237–245
8. Yau H-T, Wang J-B (2007) Fast bezier interpolator with real-time lookahead function for high-accuracy machining. Int J Mach Tools Manuf 47:1518–1529
9. Yong T, Narayanaswami R (2003) A parametric interpolator with confined chord errors, acceleration and deceleration for NC machining. Comput-Aided Des 35:1249–1259
10. 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 45:433–444
11. Nam S-H, Yang M-Y (2004) A study on a generalized parametric interpolator with real-time jerk-limited acceleration. Comput-Aided Des 36:27–36
12. 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 41:1323–1345
13. Zhiming X, Jincheng C, Zhengjin F (2002) Performance evaluation of a real-time interpolation algorithm for NURBS curves. Int J Adv Manuf Technol 20:270–276
14. Han G-C, Kim D-I, Kim H-G, Nam K, Choi B-K, Kim S-K (1999) A high speed machining algorithm for CNC machine tools. In: IECON'99 Conference Proceedings. 25th Annual Conference of the IEEE Industrial Electronics Society, San Jose, United States, Nov 1999, pp. 1493–1497
15. Yau H-T, Kuo M-J (2001) NURBS machining and feed rate adjustment for high-speed cutting of complex sculptured surfaces. Int J Prod Res 39(1):21–41
16. Farouki RT, Manjunathaiah J, Nicholas D, Yuan GF, Jee S (1998) Variable-feedrate CNC interpolators for constant material removal rates along Pythagorean-hodograph curves. Comput-Aided Des 30 (8):631–640
17. 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 36:927–935
18. 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 37:881–897. doi:10.1007/s00170-007-1041-9
19. Piegl L, Tiller W (1997) The NURBS books. Springer, Berlin