Int J Adv Manuf Technol (2012) 62:1179–1189
DOI 10.1007/s00170-011-3877-2

ORIGINAL ARTICLE

# Correcting and compressing interpolation algorithm for free-form surface machining

**Xiaohui Zhang · Dong Yu · Taotao Song**

Received: 31 March 2010 / Accepted: 21 December 2011 / Published online: 15 January 2012


**Abstract** The existing interpolation algorithm cannot meet the need of high-speed and high-accuracy machining of a free-form surface. So this paper proposed a *correcting and compressing* interpolation algorithm. Depending on the distance and angle evaluated from the adjacent command points, the machining path of free form can be divided into two machining types. For those regions where the accurate figure is critical such as corners, the convention linear interpolation is performed exactly between the adjacent command points. For those regions having a large radius of curvature where the smooth figure is critical, firstly, the interior point selection method based on circle transition is derived to reduce the tolerance between the machining path and the original surface; secondly, the interior point correction method based on the least-square method is proposed to reduce the calculation error and round-off error in the interior point and estimate the first- and second-order derivative vectors of the interior point; thirdly, the shape-defining point is selected by the bend direction of the machining path and fitted to a quintic spline curve which has the C2 continuity; fourthly, the fitting accuracy controlling method is proposed to ensure the machining accuracy; lastly, the curve interpolation is performed on the fitted smooth curve. Machining tests carried out on a vertical machining center show that the proposed algorithm can improve the machining efficiency and machining quality of a free-form surface.

**Keywords** High-speed machining · Interpolation algorithm · Free-form surface

X. Zhang (✉)
Numerical Control Research Institute,
Shanghai Sany Precision Machinery Co., LTD,
Shanghai 201200, People's Republic of China
e-mail: zhangxiaohui@sict.ac.cn

X. Zhang · D. Yu · T. Song
National Engineering Research Center for High-End CNC,
Shenyang Institute of Computing Technology,
Chinese Academy of Sciences,
Shenyang 110171, People's Republic of China

## 1 Introduction

Free-form surface is extensively applied to a wide range of industries such as automotive, aerospace, and dies/molds, but free-form surface computer numerical control (CNC) machining is still difficult. In order to mill an entire free-form surface and check collision in real time, the computer-aided manufacturing (CAM) system generally converts the free-form surface generated by computer-aided design (CAD) system into a polyhedron as shown in Fig. 1 and overlays this polyhedron with machining paths within the specified tolerances, finally generating CNC codes composed of many command points [1–3]. At this time, if the linear interpolation is performed between the adjacent command points, it can lead to many small planes mapping on the workpiece surface as shown in Fig. 13a and acceleration jumping in the machining axis, which in turn can cause resonance in machining elements [4, 5]. To describe the above algorithm easily, this paper defines the linear interpolation algorithm as the convention interpolation algorithm.

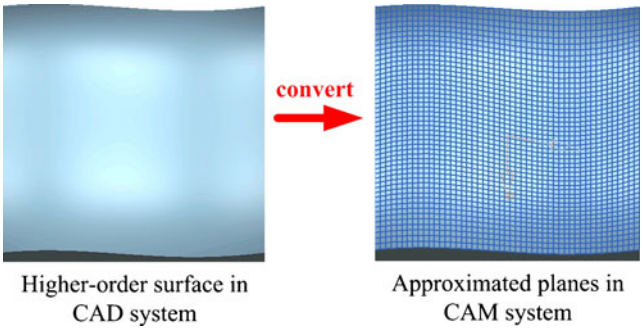



**Fig. 1** The original plane and approximated plane

To overcome the above disadvantages, many interpolation algorithms have been proposed by investigators. In technology, FANUC [5] proposed that for those portions of free-form surface having a large radius of curvature, spline interpolation is performed on the smooth curve which is calculated from the polygonal lines specified by command points (defining G-code as G05.1). In theory, Li et al. [6] proposed a non-uniform rational B-spline (NURBS) pre-interpolator for five-axis machining. Its feasibility has been evaluated only by simulation. Ye et al. [7, 8] presented an interpolation of continuous micro-line blocks based on the look-ahead algorithm, but strictly speaking, this algorithm still uses linear interpolation. Yau et al. [9] developed a real-time cubic Bezier interpolator with look-ahead function to deal with continuous micro-line blocks, but the slope discontinuity might still occur at the junction of the fitted curve blocks. Lin et al. [10] applied NURBS curve fitting technique to convert the continuous micro-line blocks to the smooth curve with NURBS format. Yeh et al. [11–13] developed a NURBS curve fitting and interpolation with a look-ahead function to handle continuous micro-line blocks, but NURBS curve fitting and interpolation require a large computational time; it cannot meet the real-time requirements of interpolation. To describe above algorithm easily, this paper defines the above interpolation algorithms as the smooth interpolation algorithm.

The existing works all assume the command points from the CAM system are just on the desired curve as shown in Fig. 2, but practically, the command points from the CAM system are placed on the boundary of tolerance [14, 15] as shown in Fig. 3. For this reason, if generating a curve with the existing algorithms capable of C2 continuity, the generated curve would depart from the desired curve and the machining accuracy and quality would be reduced.

Based on the above research and analysis, this paper proposed a correcting and compressing interpolation algorithm. By using the bi-chord error test method, the machining path was divided into two machining regions. For those regions needing the accuracy, the linear interpolation is performed between the adjacent command points. For those regions needing the smoothness, it can create multiple interior points between adjacent command points, which are closer to the original curve than the command point from the CAM system. And the position of each interior point was corrected in a unit smaller than the least input increment of CNC within tolerance; the impact of rounding error is reduced. So the machining accuracy of the free-form surface can be improved compared to the existing method.

This paper is organized in details as follows. Section 2 presents the design of the machining path analysis method. A region criterion method, interior point *selecting* method, and interior point *correcting* method are discussed in Sections 2.1, 2.2, and 2.3, respectively. Section 3 describes the design of the smooth curve generation and interpolation method. The shape-defining point selection method, shape-defining point fitting method, fitting accuracy controlling method, and smooth curve interpolation method are discussed in Sections 3.1, 3.2, 3.3, 3.4, respectively. Section 4 shows the machining tests on the vertical machining center. Section 5 concludes this paper.

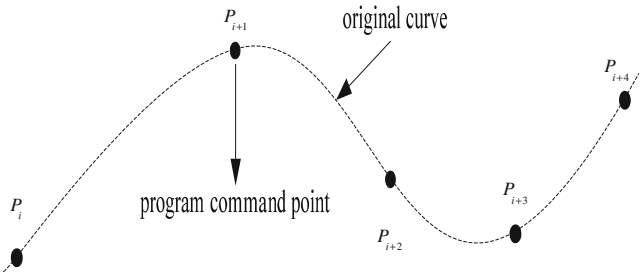



**Fig. 2** The desired curve and command point

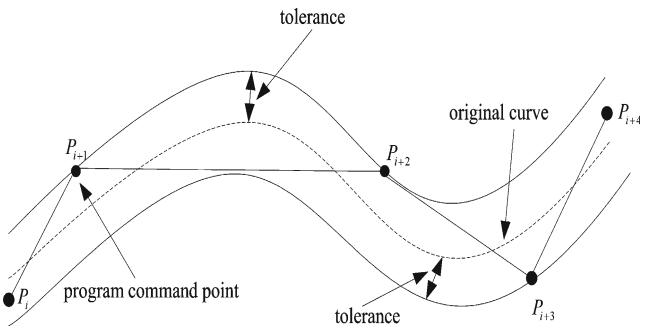



**Fig. 3** The desired curve and command point in the actual CAM system

## 2 Analysis of the machining path

### 2.1 Criterion of the machining region

When a desired sculptured free-form surface is approximated by numerous command points, the distance between adjacent command points and the angle between adjacent segments are different between those regions that need accuracy machining and those regions that need smooth machining. For those regions that need accuracy machining, the distance is longer or the angle is greater; for those regions that need smooth machining, the distance and angle are both smaller. So the machining regions can be determined by distance and angle between continuous three adjacent command points. The bi-chord error criterion method [9] can quickly determine if the machining regions need the linear interpolation or smooth interpolation.

As shown in Fig. 4, if the distance $L_{i-1i}$, $L_{ii+1}$ between three adjacent command points $P_{i\text{-}1}$, $P_i$, and $P_{i+1}$, the angle $\theta$ between two line segments $P_{i\text{-}1}P_i$ *and* $P_iP_{i+1}$, and the bi-chord error $\delta_1$, $\delta_2$ which can be calculated by formula 1 are all smaller than the critical length $L_{\max}$, critical angle $\theta_{\max}$, and critical contour error $\delta_{c\max}$ which are as CNC system input parameter, it means that the machining region consisting of three command points needs smooth interpolation. In this work, the critical length $L_{\max}$ is 0.01 mm, the critical angle $\theta_{\max}$ is 45°, and the critical contour error $\delta_{c\max}$ is 0.01 mm.

$$\begin{cases} \delta_1 = R(1-\cos\varphi_1) \\ \delta_2 = R[1-\cos\varphi_2] = R[1-\cos(\theta-\varphi_1)] \\ R = \frac{L_{i-1i}}{2\sin\varphi_1} \\ \varphi_1 = \tan^{-1}\left(\frac{L_{i-1i}\sin(\theta)}{L_{i-1i}\cos(\theta)+L_{ii+1}}\right) \end{cases} \tag{1}$$

where $R$ is the circular radius consisting of three adjacent command points $P_{i-1}$, $P_i$ and $P_{i+1}$, $\varphi_1$ and $\varphi_2$ are half of the angle $\angle P_{i-1}OP_i$ and $\angle P_iOP_{i+1}$. By applying the existing bi-chord error criterion method [9] for all command points, the machining path can be divided into two regions.

### 2.2 Selection of the interior point

For every region needing smooth machining, the method based on circle transition is proposed to calculate the interior point as shown in Fig. 5, where $P_{i-1}$, $P_i$, and $P_{i+1}$ are the three adjacent command points, and $Q_iQ_{i+1}$ is a circle which is inserted in a corner as the transition circle between the line block $P_{i-1}P_i$ and $P_iP_{i+1}$. Supposing the inserted circle $Q_iQ_{i+1}$ is tangent to the line block $P_{i-1}P_i$ and $P_iP_{i+1}$, $Q_i$ and $Q_{i+1}$ are the tangency points, and $\delta_t$ is the tolerance between the corner point $P_i$ and the circle $Q_iQ_{i+1}$. The geometrical relationship between the two line blocks$P_{i-1}P_i$, $P_iP_{i+1}$ and the circle $Q_iQ_{i+1}$ can be expressed as:

$$\begin{cases} L = \frac{\sin\frac{\theta}{2}}{1-\cos\frac{\theta}{2}}\delta_t \\ R = \frac{\cos\frac{\theta}{2}}{1-\cos\frac{\theta}{2}}\delta_t \end{cases} \tag{2}$$

where $L$ is the transition distance from the tangency point $Q_i$ or $Q_{i+1}$ to the corner point $P_i$, $\theta$ is the included angle between the two movement vectors $\overrightarrow{P_{i-1}P_i}$ and $\overrightarrow{P_iP_{i+1}}$, and $R$ is the radius of the inserted circle $Q_iQ_{i+1}$. Comparing Fig. 5 with Fig. 3, it is easy to see that if the $\delta_t$ is equal to the tolerance in the CAM system, the tangency points $Q_i$ and $Q_{i+1}$ are much closer to the original curve than the command point $P_i$. So this paper takes the two points

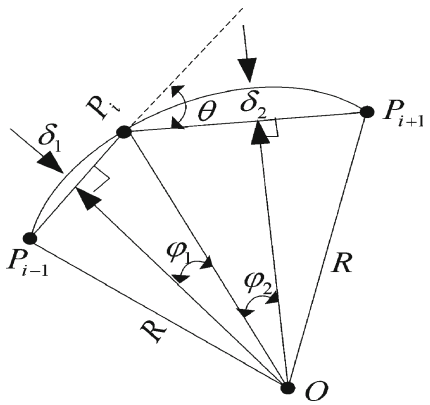



**Fig. 4** Schematic diagram of the bi-chord test

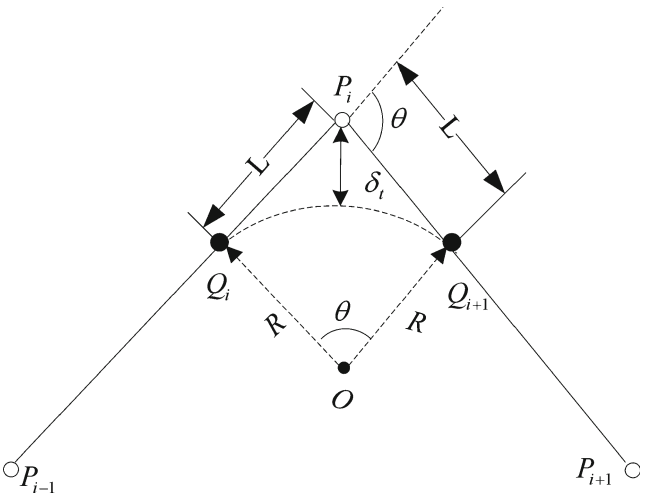



**Fig. 5** The circle transition

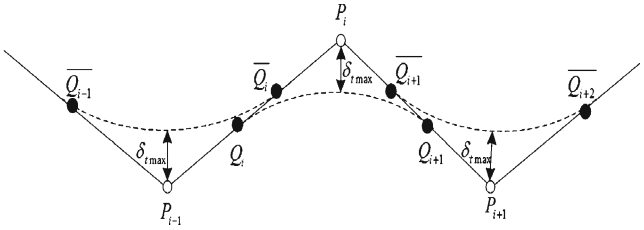



**Fig. 6** The crossing of two adjacent interior points

$Q_i$ and $Q_{i+1}$ as the interior points corresponding to command point $P_i$. At this time, the coordinate of $Q_i$ and $Q_{i+1}$ can be calculated as:

$$\begin{cases} Q_i = \frac{L}{L_{i-1i}} P_{i-1} + \frac{L_{i-1i}-L}{L_{i-1i}} P_i \\ Q_{i+1} = \frac{L_{ii+1}-L}{L_{ii+1}} P_i + \frac{L}{L_{ii+1}} P_{i+1} \end{cases} \tag{3}$$

where $L_{i-1i}$ and $L_{ii+1}$ are the length of line blocks $P_{i-1}P_i$ and $P_iP_{i+1}$. The coordinate of $P_{i-1}$, $P_i$, and $P_{i+1}$ can be acquired from the CNC code. In order to avoid the cross of two interior points as shown in Fig. 6, where $\overline{Q_{i-1}}$and $\overline{Q_i}$ are the two interior points corresponding to the command point $P_{i-1}$, $Q_i$ and $Q_{i+1}$ are the two interior points corresponding to the command point $P_i$, $\overline{Q_{i+1}}$and $\overline{Q_{i+2}}$ are the two interior points corresponding to command point $P_{i+1}$. The transition distance $L$ is also confined by the length of two adjacent line blocks, as shown in Eq. 4:

$$L \le \min\left(\frac{L_{i-1i}}{2}, \frac{L_{ii+1}}{2}\right) \tag{4}$$

For every region needing smooth machining, all interior points except the start and end can be calculated by the above circle transition method. The start and end command points can be directly adopted as interior points.

### 2.3 Correction of the interior point

In order to reduce the calculation error and round-off error in the interior point, the interior point correction method is proposed. Before correcting the interior point, the parametric value must firstly be *assigned* to each interior point. The centripetal method [16] is adopted in this paper. Equation 5 describes centripetal method to calculate the parametric value for each interior point.

$$\begin{cases} u_1 = 0 \\ u_i = u_{i-1} + \dfrac{\sqrt{(|Q_{i-1}Q_i|)}}{\sum_{j=2}^{n} \sqrt{(|Q_{j-1}Q_j|)}} \quad (i = 2, 3, \cdots n) \end{cases} \tag{5}$$

where $u_i$ is a parametric value corresponding to interior point $Q_i$, $|Q_{i-1}Q_i|$ is the distance between the interior point $Q_{i-1}$ and $Q_i$, and $|Q_{j-1}Q_j|$ is the distance between the interior point $Q_{j-1}$ and $Q_j$. When all interior points have their own parametric value, the orthogonal basis $u^3$, $u^2$, $u$, 1 is used to approximate continuous five interior points. For arbitrary continuous five interior points $Q_i$, $Q_{i+1}$, $Q_{i+2}$, $Q_{i+3}$, and $Q_{i+4}$, the approximation curve can be written as:

$$Q_i(u) = a_i u^3 + b_i u^2 + c_i u + d_i \quad (u_i \le u \le u_{i+4}) \tag{6}$$

where $a_i$, $b_i$, $c_i$, and $d_i$ are coefficient vectors of $Q_i(u)$ (The dimension of those coefficient vectors are the same as the number of movement axis), $u_i$ and $u_{i+4}$ are parameter values corresponding to interior points $Q_i$ and $Q_{i+4}$. The square sum of distances $J_i$ from the interior point $Q_i$, $Q_{i+1}$, $Q_{i+2}$,$Q_{i+3}$, $Q_{i+4}$ to the respective curve point which is on the curve $Q_i(u)$ and corresponds to the parameter value $u_i$, $u_{i+1}$, $u_{i+2}$, $u_{i+3}$, $u_{i+4}$ can be written as:

$$J_i = \sum_{k=i}^{i+4} |Q_i(u_k) - Q_k|^2 = J_{ix} + J_{iy} + J_{iz} \tag{7}$$

where

$$\begin{cases} J_{ix} = \sum_{k=i}^{i+4} |Q_{ix}(u_k) - x_k|^2 \\ J_{iy} = \sum_{k=i}^{i+4} |Q_{iy}(u_k) - y_k|^2 \\ J_{iz} = \sum_{k=i}^{i+4} |Q_{iz}(u_k) - z_k|^2 \end{cases} \tag{8}$$

$Q_{ix}(u_k)$, $Q_{iy}(u_k)$, and $Q_{iz}(u_k)$ are the coordinate values of point $Q_i(u_k)$ at $x$, $y$, and $z$ directions; $x_k$, $y_k$, and $z_k$ are the coordinate values of interior point $Q_k$ at $x$, $y$, and $z$ directions; and $J_x$, $J_y$, and $J_z$ are square sum of distances at $x$, $y$, and $z$ directions. In order to minimize the square sum of distances $J_i$, the partial derivative of $J_x$, $J_y$, and $J_z$ must meet the following requirements:

$$\begin{cases} \frac{\partial J_{ix}}{\partial a_{ix}} = \frac{\partial J_{ix}}{\partial b_{ix}} = \frac{\partial J_{ix}}{\partial c_{ix}} = \frac{\partial J_{ix}}{\partial d_{ix}} = 0 \\ \frac{\partial J_{iy}}{\partial a_{iy}} = \frac{\partial J_{iy}}{\partial b_{iy}} = \frac{\partial J_{iy}}{\partial c_{iy}} = \frac{\partial J_{iy}}{\partial d_{iy}} = 0 \\ \frac{\partial J_{iz}}{\partial a_{iz}} = \frac{\partial J_{iz}}{\partial b_{iz}} = \frac{\partial J_{iz}}{\partial c_{iz}} = \frac{\partial J_{iz}}{\partial d_{iz}} = 0 \end{cases} \tag{9}$$

By substituting the coordinate value and parametric value of the five interior points $Q_i$, $Q_{i+1}$, $Q_{i+2}$, $Q_{i+3}$, and $Q_{i+4}$ into

Eq. 9, the coefficient vectors of curve $Q_i(u)$ at $x$, $y$, and $z$ directions can be represented as :

$$\begin{bmatrix} a_{ix} & a_{iy} & a_{iz} \\ b_{ix} & b_{iy} & b_{iz} \\ c_{ix} & c_{iy} & c_{iz} \\ d_{ix} & d_{iy} & d_{iz} \end{bmatrix} = \begin{bmatrix} \sum_{k=i}^{i+4} u_k^3 & \sum_{k=i}^{i+4} u_k^2 & \sum_{k=i}^{i+4} u_k & \sum_{k=i}^{i+4} 1 \\ \sum_{k=i}^{i+4} u_k^4 & \sum_{k=i}^{i+4} u_k^3 & \sum_{k=i}^{i+4} u_k^2 & \sum_{k=i}^{i+4} u_k \\ \sum_{k=i}^{i+4} u_k^5 & \sum_{k=i}^{i+4} u_k^4 & \sum_{k=i}^{i+4} u_k^3 & \sum_{k=i}^{i+4} u_k^2 \\ \sum_{k=i}^{i+4} u_k^6 & \sum_{k=i}^{i+4} u_k^5 & \sum_{k=i}^{i+4} u_k^4 & \sum_{k=i}^{i+4} u_k^3 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{k=i}^{i+4} x_k & \sum_{k=i}^{i+4} y_k & \sum_{k=i}^{i+4} z_k \\ \sum_{k=i}^{i+4} u_k x_k & \sum_{k=i}^{i+4} u_k y_k & \sum_{k=i}^{i+4} u_k z_k \\ \sum_{k=i}^{i+4} u_k^2 x_k & \sum_{k=i}^{i+4} u_k^2 y_k & \sum_{k=i}^{i+4} u_k^2 z_k \\ \sum_{k=i}^{i+4} u_k^3 x_k & \sum_{k=i}^{i+4} u_k^3 y_k & \sum_{k=i}^{i+4} u_k^3 z_k \end{bmatrix} \tag{10}$$

Once the coefficient vectors of $Q_i(u)$ are calculated, the coordinate, first- and second-order derivative vectors on the curve $Q_i(u)$ at parametric value $u_i$, $u_{i+1}$, $u_{i+2}$, $u_{i+3}$, $u_{i+4}$ can be calculated as:

$$\begin{cases} Q_i(u_k) = a_i u_k^3 + b_i u_k^2 + c_i u_k + d_i \\ Q'_i(u_k) = 3a_i u_k^2 + 2b_i u_k + c_i \\ Q''_i(u_k) = 6a_i u_k + 2b_i \end{cases} \quad (i \le k \le i+4) \tag{11}$$

For all interior points except for the first three and last three interior points such as $Q_1$, $Q_2$, $Q_3$ and $Q_{n-2}$, $Q_{n-1}$, $Q_n$, the corrected interior point, first- and second-order derivative vectors can be estimated as the following:

$$\begin{cases} O_k = \frac{1}{3}(Q_i(u_k) + Q_{i+1}(u_k) + Q_{i+2}(u_k)) \\ O'_k = \frac{1}{3}(Q'_i(u_k) + Q'_{i+1}(u_k) + Q'_{i+2}(u_k)) \\ O''_k = \frac{1}{3}(Q'_i(u_k) + Q''_{i+1}(u_k) + Q''_{i+2}(u_k)) \end{cases} \begin{matrix} \\ (1 \le i \le n-4) \\ (4 \le k \le n-4) \end{matrix} \tag{12}$$

where $O_k$ is the corrected interior point of the interior point $Q_k$, and $O'_k$ and $O''_k$ are the first- and second-order derivative vectors at the point $O_k$. Due to the lack of the extra neighboring interior point, the first three interior points can be corrected as follows:

$$\begin{cases} O_1 = Q_1(u_1), O_2 = \frac{1}{2}(Q_1(u_2) + Q_2(u_2)), O_3 = \frac{1}{3}(Q_1(u_3) + Q_2(u_3) + Q_3(u_3)) \\ O'_1 = Q'_1(u_1), O'_2 = \frac{1}{2}(Q'_1(u_2) + Q'_2(u_2)), O'_3 = \frac{1}{3}(Q'_1(u_3) + Q'_2(u_3) + Q'_3(u_3)) \\ O''_1 = Q''_1(u_1), O''_2 = \frac{1}{2}(Q''_1(u_2) + Q''_2(u_2)), O''_3 = \frac{1}{3}(Q''_1(u_3) + Q''_2(u_3) + Q''_3(u_3)) \end{cases} \tag{13}$$

The last three interior points can also be corrected by replacing $Q_1$ with $Q_n$, $Q_2$ with $Q_{n-1}$, and $Q_3$ with $Q_{n-2}$. With the above method, the interior point can be corrected, and the derivative vector can be estimated.

## 3 Generation and interpolation of the smooth curve

### 3.1 Selection of the shape-defining point

Considering the estimated first- and second-order derivative victors, the quintic spline is chosen to fit the corrected interior point. But if each two corrected interior points are fitted to a curve, much more computation time and memory space are needed. In order to reduce the computation time and compress the interior point, the interior point where the machining path bends to the different direction is defined as the shape-defining point. Because in practical application, if the machining path specified by more than three interior points bends to the same direction, the fitted quintic spline curve with first and end points will meet the accuracy requirement. For the machining path specified by corrected interior points$O_1$, $O_2$…, $O_{n-1}$ and $O_n$ as shown in Fig. 7, the shape-defining points can be selected as the following steps:

1. Mark the start corrected interior point as the shape-defining point, and calculate the normal vector $v_{12}$ with the first-order derivative vector $O'_1$ and$O'_2$ at points $O_1$ and $O_2$ as follows:

$$v_{12} = O'_1 \times O'_2 \tag{14}$$

2. Judge whether the values of $j$ and n are the same ($j$ starting from 3). If $j<n$, calculate the normal vector $v_{j-1j}$ as Eq. 14 and jump to 3; else, mark the point $O_j$ as the shape-defining point and jump to 6.

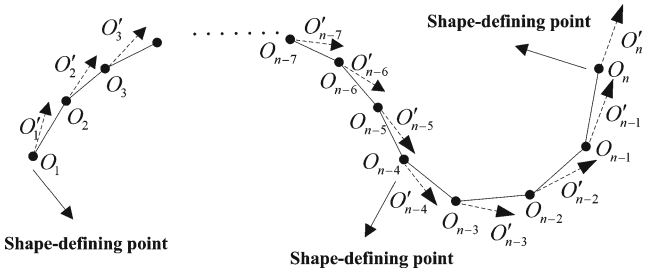



**Fig. 7** Schematic diagram of the shape-defining point

3. Calculate the intersection angle $\alpha_j$ between the vector $v_{j-2j-1}$ and $v_{j-1j}$, and judge whether the value of $\alpha_j$ is greater than 90°. If $\alpha_j$<90°, jump to 4; else, jump to 5.
4. If $\alpha_j$<90°, it shows the machining path at points $O_j$ and $O_1$ bends to the same direction. Jump to 2 with $j$+ +.
5. If $\alpha_j$<90°, it shows the machining path at points $O_j$ and $O_{j-1}$ bends to the different direction. Mark the point $O_j$ as the shape-defining point, use $O_{j-1}$ as the new starting point, and jump to 1 with $j$+ +;
6. Exit this shape-defining point selection.

### 3.2 Fitting of the shape-defining point

In the shape-defining point fitting, the objective is to connect $n$ shape-defining points with $n-1$ fifth-order spline curves $O_1$,…, $O_{n-1}$.These spline curves are fitted in such a way that continuity up to the second derivative is preserved along the overall composite curve. For the arbitrarily adjacent shape-defining points $O_i$ and $O_j$, the polygonal line machining path defined by corrected interior points $O_i$,… $O_j$ and the smooth curve machining path $O_k(u)$ fitted by geometry information of shape-defining points $O_i$ and $O_j$ are shown in Fig. 8. The curve $O_k(u)$ can be expressed as:

$$O_k(u) = A_k u^5 + B_k u^4 + C_k u^3 + D_k u^2 + E_k u + F_k \quad (u \in [u_i, u_j]) \tag{15}$$

where $A_k$, $B_k$, $C_k$, $D_k$, $E_k$, and $F_k$ are coefficient vectors of curve $O_k(u)$ (the dimension of those coefficient vectors are the same as the number of movement axis), $u_i$ and $u_j$ are parameter values corresponding to shape-defining points $O_i$ and $O_j$. In order to make the curve $O_k(u)$ pass through the shape-defining point and ensure second-order continuity, the coordinate, first- and second-order derivative vectors at the shape-defining points $O_i$ and $O_j$ are used as boundary conditions. The boundary conditions can be written as:

$$\begin{bmatrix} u_i^5 & u_i^4 & u_i^3 & u_i^2 & u_i & 1 \\ 5u_i^4 & 4u_i^3 & 3u_i^2 & 2u_i & 1 & 0 \\ 20u_i^3 & 12u_i^2 & 6u_i & 1 & 0 & 0 \\ u_j^5 & u_j^4 & u_j^3 & u_j^2 & u_j & 1 \\ 5u_j^4 & 4u_j^3 & 3u_j^2 & 2u_j & 1 & 0 \\ 20u_j^3 & 12u_j^2 & 6u_j & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} A_k \\ B_k \\ C_k \\ D_k \\ E_k \\ F_k \end{bmatrix} = \begin{bmatrix} O_i \\ O_i' \\ O_i'' \\ O_j \\ O_j' \\ O_j'' \end{bmatrix} \tag{16}$$

where $O_i$, $O_i'$, $O_i''$, $O_j$, $O_j'$, and $O_j''$ are the coordinates, first- and second-order derivative vectors at the shape-defining points $O_i$ and $O_j$, which have been calculated in Section 2.3. By substituting these values into Eq. 16, the $A_k$, $B_k$, $C_k$, $D_k$, $E_k$, and $F_k$ coefficients vector can be calculated.

### 3.3 Controlling of fitting accuracy

Although the fitted curve passes through the shape-defining points and has C2 continuity along the overall composite curve, it does not ensure the accuracy at all corrected interior points between the two adjacent shape-defining points. In order to solve this problem, the distance from the corrected interior point to the fitted spline curve is calculated to judge whether the fitted spline curve meets the accuracy requirement at the corrected interior point. Considering the real time of the CNC system, the above distance is replaced by the distance between the interior point and the fitted curve point which is on the fitted curve and corresponds to the parameter value of the interior point. If the fitted spline curve does not meet the accuracy requirement, the shape of the fitted curve is amended by increasing the shape-defining point. For the fitted curve $O_k(u)$ and corrected interior points $O_{i+1}$ … $O_{j-1}$ between the shape-defining points $O_i$ and $O_j$, the fitting accuracy controlling can be carried out by the following steps as shown in Fig. 9.

1. Calculate the distance $L_m$ from the corrected interior point $O_m$ ($m$ starting from $i$+1) to the fitted curve point $O_k(u_m)$, where $u_m$ is the parameter value corresponding to the interior point $O_m$. Judge whether the value of $L_m$ is larger than $e_{cmax}$, where $e_{cmax}$ is the maximum allowable

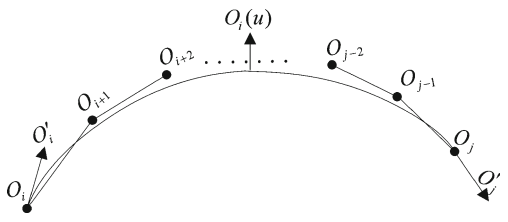


**Fig. 8** The fitted machining path

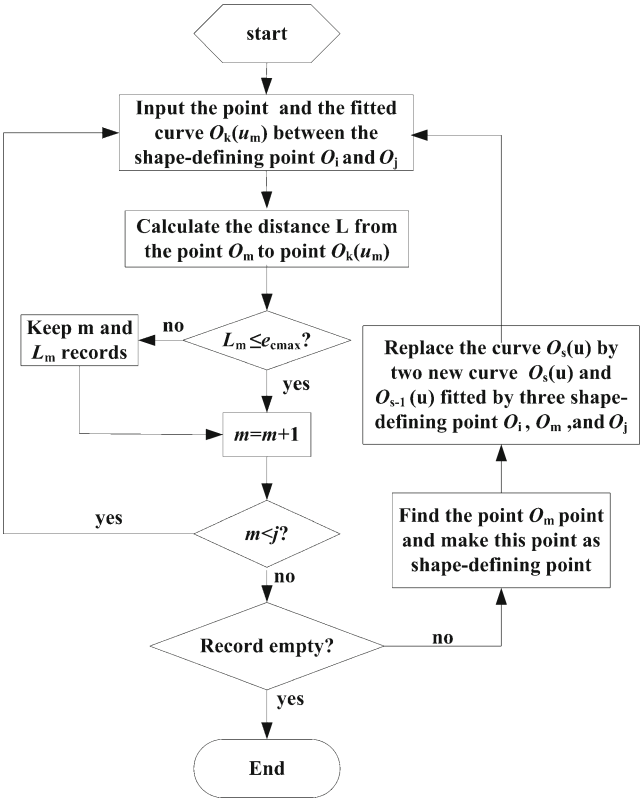



**Fig. 9** Flowchart of the fitting accuracy controlling method

contour error in CNC machining. If $L_m \le e_{cmax}$, jump to 2; else, jump to 3.

2. If $L_m \le e_{cmax}$, it shows the fitted curve $O_k(u)$ meets the accuracy requirement at the interior point $O_m$. Judge whether the values $m$ and $j$ are the same. If $m<j$, jump to 1 with $m++$; else jump to 4.


**Fig. 10** SMTCL VM650 CNC machining center

**Fig. 11** The actual machining process

3. If $L_m > e_{cmax}$, it shows the fitted curve $O_k(u)$ does not meet the accuracy requirement at interior point $O_m$. Keep the current $m$ and $L_m$ values as record, and judge whether the values of $m$ and $j$ are the same. If $m<j$, jump to 1 with $m++$; else jump to (4).
4. Jude whether there are $m$ and $L_m$ values in the record. If not, it shows the fitted curve $O_k(u)$ meets the accuracy requirement at all the interior points and jump to 6; else, it shows the fitted curve $O_k(u)$ does not meet the accuracy requirement at some interior point; find the interior point which is the farthest from the curve $O_k(u)$ according to the values of $m$ and $L_m$, and mark it as new shape-defining point; jump to 5.
5. According new and old shape-defining points between the shape-defining points $O_i$ and $O_j$, generate two new curves $O_{s-1}(u)$ and $O_s(u)$ to replace curve

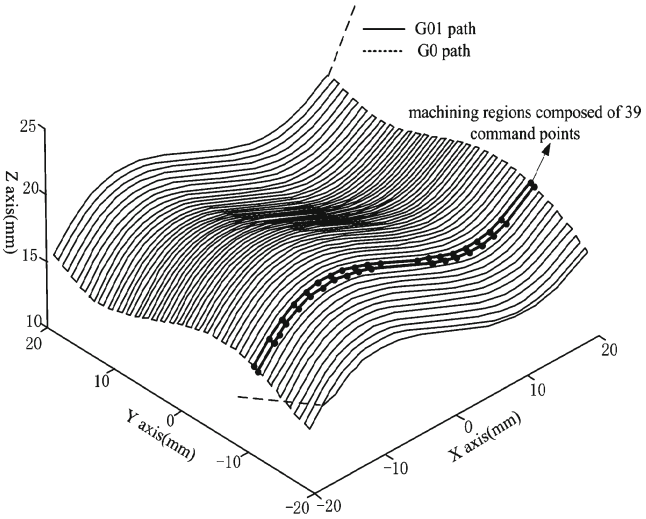



**Fig. 12** The actual machining path

$O_k(u)$ as the new fitted curve between the shape-defining points $O_i$ and $O_j$. Repeat the above process, until the fitted curve meets the accuracy requirement at all the interior points between the shape-defining points $O_i$ and $O_j$.

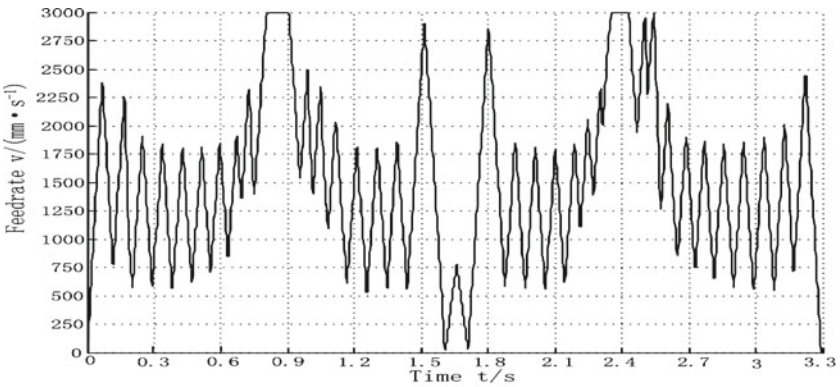
6. Exit fitting accuracy controlling.

### 3.4 Interpolation of the smooth spline curve

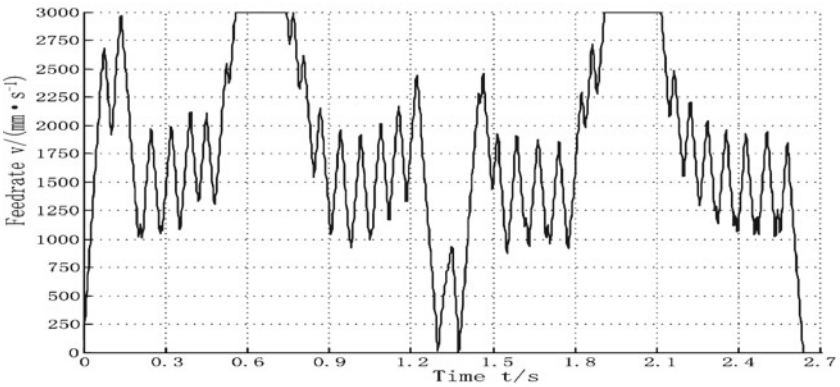
In this paper, the spline curve interpolation algorithm based on the second-order Taylor expansion [17] is adopted to calculate the interpolation point on the fitted spline curve. At this time, the value of parameter $u$ at $i$ interpolation cycle can be expressed as:

$$u_i = u_{i-1} + \frac{V_{i-1} \bullet T}{|O'(u_{i-1})|} + \frac{T^2}{2} \times \left( \frac{A_{i-1}}{2|O'(u_{i-1})|} - \frac{V_{i-1}^2 |O'(u_{i-1})| \bullet |O''(u_{i-1})|}{2|O'(u_{i-1})|^2} \right) \quad (17)$$

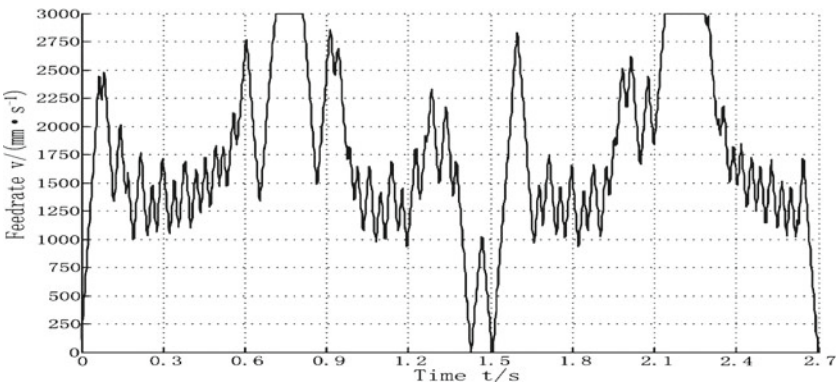
where $T$ is the interpolation cycle, $V_{i-1}$ and $A_{i-1}$ are feedrate and acceleration at $i$ interpolation cycle, and $O'(u_{i-1})$ and $O''$




(a) The conventional interpolation

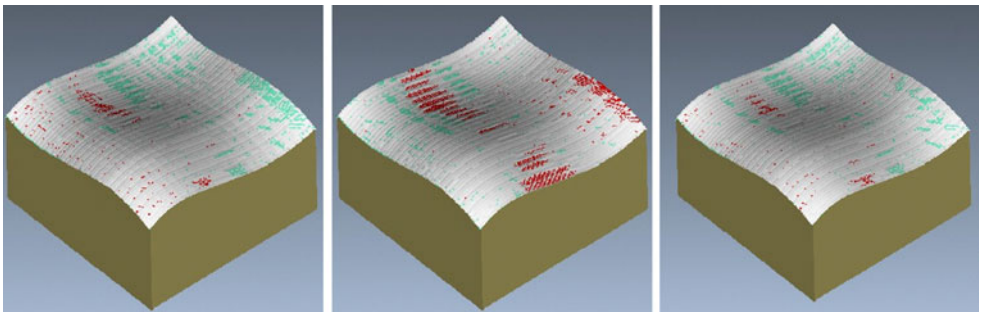



(b) The smooth interpolation




(c) The proposed interpolation

**Fig. 13** Comparison diagram of feedrate. **a** The conventional interpolation, **b** the smooth interpolation, **c** the proposed interpolation



(a) The conventional interpolation (b) The smooth interpolation (c) The proposed interpolation

**Fig. 14** Comparison results with the original surface. **a** The conventional interpolation, **b** the smooth interpolation, **c** the proposed interpolation

($u_{i-1}$) are the first- and second-order derivative vectors of the spline curve $O(u)$ at $u_i$. Based on our previous work [18], the feedrate $V_{i-1}$ and acceleration $A_{i-1}$ can be planned. By substituting the planned parameter value into Eq. 17, the interpolation point which is as reference command of the servo motion control system can be calculated.

## 4 Machining tests and discussion

To evaluate the proposed interpolation algorithm, the algorithm has been implemented in SMTCL VM650 CNC machining center as shown in Fig. 10, which comprises an industrial PC, a Mechatrolink-III communication card, Mechatrolink-III filed bus, and a mechanical system with three 3-phase Yaskawa AC servo packs. The industrial PC with a Pentium M 1.6-GHz CPU and 512-M RAM is used to run the open CNC system based on RTLinux (Real-Time Linux) operating system which is developed by us [19]. The tasks in the open CNC system are divided into the HMI task for human and machine operations, the interpreter task for interpreting CNC codes and generating smooth curve, the motion controller task for interpolating trajectory point and generating motion command, the PLC task for controlling I/O status, and the field bus driver task for receiving and sending communication card date. In terms of task cycle time, HMI is 20 ms, interpreter 5 ms, motion controller 2 ms, PLC 4 ms, and field bus driver 2 ms. The Mechatroline-III communication card is used to control the three AC servo packs at a sampling period of 2 ms.

The free-form surface as shown in Fig. 1 is used as a machining example. Figure 12 shows the machining path which is generated by CAD/CAM soft UG with tolerance of 0.01 mm. The number of CNC blocks is 12,416. The machining parameters used in this example are as follows: the interpolation cycle $T$ is 2 ms, the feedrate limit $F$ is 3,000 mm/min, the acceleration limit $A_{\max}$ is 500 mm/s$^2$, the tolerance limit $\delta_{n\max}$ is 0.01 mm, the contour error limit $\delta_{c\max}$ is 0.01 mm, and the tool is ball end mill with diameter 6 mm; the machining process is shown in Fig. 11.

In order to show the feedrate change for the conventional interpolation algorithm, the smooth interpolation algorithm, and the proposed algorithm expressly, an arbitrary machining path composed of 39 line blocks is adopted as shown in Fig. 12. The feedrate profiles for different algorithms are shown in Fig. 13. For the conventional interpolation algorithm, the machining time is 3.294 s. Comparing with the conventional interpolation algorithm, the machining times of the smooth interpolation algorithm and proposed algorithm are respectively reduced to 2.634 and 2.689 s. This is mainly because for smooth interpolation algorithm and proposed interpolation

AUTO-DIFF Solid Report

Comparison Type: Gouge and Excess
Gouge Tolerance: 0.01
Excess Tolerance: 0.01

DESIGN COMPONENT(S):

Component Name: Design
Component Type: Design
Model Type: POLYGON
SUMMARY:
Maximum gouge of 0.518336 occurred at record 4550
Number of Gouges: 273
Maximum excess of 17.472983 occurred at record 12401
Number of Excesses: 1021

*(a) The conventional interpolation*

AUTO-DIFF Solid Report

Comparison Type: Gouge and Excess
Gouge Tolerance: 0.01
Excess Tolerance: 0.01

DESIGN COMPONENT(S):

Component Name: Design
Component Type: Design
Model Type: POLYGON

SUMMARY:
Maximum gouge of 0.91569 occurred at record 9373
Number of Gouges: 480
Maximum excess of 17.513253 occurred at record 11964
Number of Excesses: 786

*(b) The smooth interpolation*

AUTO-DIFF Solid Report

Comparison Type: Gouge and Excess
Gouge Tolerance: 0.01
Excess Tolerance: 0.01

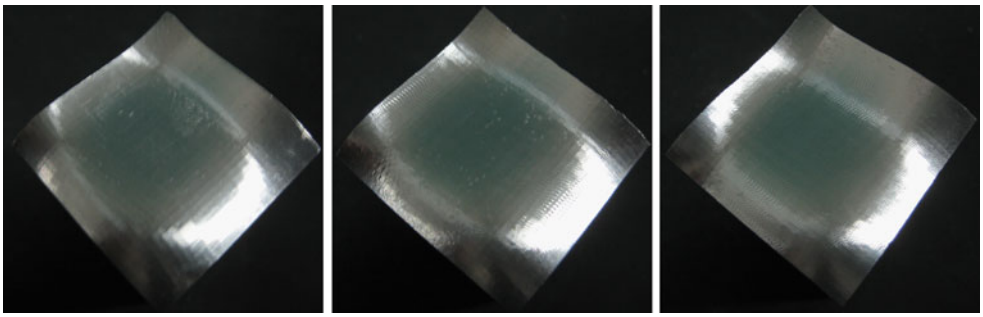
DESIGN COMPONENT(S):

Component Name: Design
Component Type: Design
Model Type: POLYGON

SUMMARY:
Maximum gouge of 0.734206 occurred at record 6866
Number of Gouges: 124
Maximum excess of 17.450889 occurred at record 11965
Number of Excesses: 703

*(c) The proposed interpolation*

**Fig. 15** The auto-diff solid report. **a** The conventional interpolation, **b** the smooth interpolation, **c** the proposed interpolation



(a) The conventional interpolation (b) The smooth interpolation (c) The proposed interpolation

**Fig. 16** The workpiece actual machining results. **a** The conventional interpolation, **b** the smooth interpolation, **c** the proposed interpolation

algorithm, the interpolation is performed on the smooth spline curve; the transition feedrate between the adjacent blocks is much increased.

Besides interpolation algorithm, there are many outside factors affecting actual machining accuracy and quality such as machine rigidity, servo system, and so on. Therefore, the actual workpiece machining results cannot directly distinguish interpolation algorithm good or bad. In order to rule out other outside factors, the interpolation points in the actual machining process were sampled to compare with the designed surface from the UG software in VERICUT software within tolerance of 0.01 mm. The experimental results for different interpolation algorithm were shown in Fig. 14, and the compared reports of VERICUT software were shown in Fig. 15.

For the conventional interpolation algorithm, the numbers of over-cut and owe-cut points are, respectively, 273 and 1,021 as shown in Fig. 15a, and there are a large number of uneven corners and small planes mapping on the machining surface as shown in Figs. 14a and 16a which is the actual machining result. This is due to the fact that (1) the command points from the CAM system are not just located on the target original curve, and (2) the tool movement precisely follows the line segment which is used to approximate a target original curve. Although this effect is caused by high-accuracy machining which precisely follows the command point, the uneven corners that result will be judged unsatisfactory when smooth surface is required.

After applying the smooth interpolation algorithm which is the existing algorithm capable of C2 continuity, the numbers of over-cut and owe-cut points are, respectively, 480 and 786 as shown in Fig. 15b. Comparing with the conventional interpolation algorithm, the over-cut points were reduced, but the total numbers of over-cut and owe-cut points are not reduced. And it is easy to see that the number of small planes mapped on the machining surface is reduced, and the machining surface is much smoother as shown in Figs. 14b and 16b. This is mainly because, for the smooth interpolation algorithm, the interpolation is performed on the smooth curve fitted by the command point, but the tolerance, calculation error, and round-off error in the command point are not corrected.

After applying the proposed algorithm, the numbers of over-cut and owe-cut points are all reduced as shown in Fig. 15c, and the numbers of small place mapped on the machining surface are reduced as shown in Figs. 14c and 16c. This is mainly because the tolerance, calculation error, and rounding error in the command point are corrected.

## 5 Conclusions

This paper proposed a *correcting and compressing* interpolation algorithm for high-speed free-form surface machining. According to the command point, the algorithm can automatically determine whether the accurate figure or smooth figure is required for the current machining region. For the machining region needing the accuracy, the traditional linear interpolation is performed exactly as specified by the command points. For the machining region needing the smoothness, the curve interpolation is performed on the smooth curve which is calculated from the polygonal lines specified by the command point. The proposed algorithm can ensure that the fitted curve has the C2 continuity and meets the requirements of machining accuracy. As a result, smoother machining and better surface quality can be obtained. The machining tests are performed on a three-axis vertical machining center for testing the proposed algorithm. The machining results show that the proposed algorithm can reduce the machining time and increase the machining quality.

**Acknowledgments** This research is supported by the Knowledge Innovation Program of the Chinese Academy of Sciences (KGCX2-YW-119) and Important National Science & Technology Specific Projects Concerning High-End CNC Machine Tools and Basic Manufacturing Equipment (2009ZX04009-013).

## References


1. Vickers GW, Bradley C (1992) Curved surface machining through circular arc interpolation. Computer in Industry 19 (33):329–37
2. Erkormaz K, Altinas Y (2001) High speed CNC system design part II: modeling and identification of feed drives. Int J Mach Tool Manuf 41(8):1487–1509
3. Bedi S, Ali I, Quan N (1993) Advanced interpolation techniques for CNC machines. ASME Trans J End Ind 115(8):329–336
4. SIEMENS (2004) SINUMERIK 810D/840D tool and mold making, 4
5. FANUC (2004) Series 30i/31i/32i—a user's manual for MC, GFZ-63944 EN-2/03, 9
6. Li W, Liu YD, Yamazaki K, Fujisima M, Mori M (2008) The design of a NURBS pre-interpolator for five-axis machining. Intern J Advanced Manufacture Tech 36(5):927–938
7. Ye P, Shi C, Yang K et al (2008) Interpolation of continuous micro line segment trajectories based on look-ahead algorithm in high-speed machining. Inter J Manufacture Tech 37:881–897
8. Leng HB, Wu YJ, Pan XH (2009) Adaptive prospective interpolation method for high speed machining of micro line blocks based on the cubic polynomial model. Chinese J Mechanical Engineering 45(6):73–79
9. Yau HT, Wang JB (2007) Fast Bezier interpolator with real-time look-ahead function for high accuracy machining. Int J Mach Tool Manuf 47(10):1518–1529
10. Lin KY, Ueng WD, Lai JY (2008) CNC codes conversion from linear and circular paths to NURBS curves. Int J Adv Manuf Technol 39(7–8):760–773
11. Yeh SS, Su HC (2009) Implementation of online NURBS curve fitting process on CNC machines. Int J Adv Manuf Technol 40 (5):531–540
12. Wang JB, Yau HT (2009) Real-time NURBS interpolator: application to short linear segments. Int J Adv Manuf Technol 41 (11):1169–1185
13. Tsai MS, Nien HW, Yau HT (2010) Development of a real-time look-ahead interpolation methodology with spline-fitting technique for high-speed machining. Int J Adv Manuf Technol 47(5–8):621–638
14. Hino TO, Fujiyoshida SI, Yamanashi HS (2007) Curve interpolation method. United States Patent 7274969 B2, 2007, 09, 25
15. Yutkowitz S J, OH WC (2004) Motion control system and method utilizing spline interpolation. United States Patent 6782306 B2, 2004, 08, 24
16. Lee ETY (1989) Choosing nodes in parametric curve interpolation. Computer Aided Design 21(6):363–370
17. Yang DCH, Kong T (1994) Parametric interpolator versus linear interpolator for precision surface machining. Computer-Aided Design 26(3):225–234
18. Zhang XH, Yu D, Hu Y, Hong HT, Sun WT (2009) Development of NURBS curve interpolator with look-ahead control and federate filtering for CNC system. Industrial Electronics and Applications (ICIEA), 25-27 May: 2755–2759
19. Yu D, Hu Y, Xu XW et al (2009) An open CNC system based on component technology. IEEE Transactions On Automation Sci Engineering 6(2):302–310