ORIGINAL ARTICLE

Curve Fitting of NURBS Curve for CNC Motion Controller

Chin-Sheng Chen[1,*] and Cheng-Yin Tsai[2]

[1]*National Taipei University of Technology, Graduate Institute of Automation Technology, 1, Sec. 3, Zhongxiao E. Rd, Taipei, Taiwan, 106*
[2]*Airoha Technology Co., 5F., No.6-5, Dusing Road, Hsinchu Science Park, Hsinchu, Taiwan, 300*
(**Received** 9 May 2017; **Accepted** 14 July 2017; **Published on line** 1 September 2017)
*Corresponding author: saint@mail.ntut.edu.tw
DOI: 10.5875/ausmt.v7i3.1437

Abstract: This paper proposes a real-time coplanar Non-Uniform Rational B-Spline (NURBS) curve fitting in a CAD/CAM system to improve fitting quality and reduce calculation effort. Curve segmentation is applied to filter the line segments and reshape the different NURBS curve segments. Both position errors and the first derivative are considered to avoid over fitting of the NURBS curve. After the fitted NURBS curve is obtained, the corresponding interpolation for each interpolated time is further processed to obtain the motion command for each axis. The real-time NURBS curve fitting process consists of (1) discrete points import, (2) linear segment filter, (3) curve reshaping, (4) curve fitting considering both the position errors and the first derivative. Simulation results show the proposed approach effectively improves the fitting quality and calculation efficiency.

Keywords: NURBS curve, curve fitting, interpolation, curve reshaping

Introduction

Most CAD/CAM systems provide design tools to create 2D and 3D curves and surfaces using parametric forms. However, the traditional motion controller only supports linear and circular interpolation, as shown in Fig. 1, which results in a huge number of linear or circular segments in CAD/CAM system output. In contrast, the modern motion controller, shown in Fig. 2, not only provides linear or circular interpolation but also offers parametric interpolation for curves such as Bezier, B-spline, and NURBS curves [1][2]. The aforementioned 2D and 3D complex curve segments can be directly transferred to the motion controller, thus dramatically reducing the number of segments. In the motion controller, the CNC part programs are first interpreted into different interpolation functions. The interpolated commands are then fed into the servo control loop to drive the machine tools.

The NURBS curve is a parametric form in mathematics that is usually used in CAD/CAM systems and offers good flexibility for representing freeform curves. Figure 2 illustrates the appropriate use of the NURBS curve in modern CAD/CAM and motion systems. In traditional CAD/CAM systems, motion system performance is limited in applications since the tolerance and discontinuity of the linked segments deteriorates machining quality during curve or surface representation. As shown in Fig. 3, in CAD/CAM systems, these steps usually interpolate the designed curves use small segments to reduce these errors [3-8], at the cost of more complex calculations.

To solve abovementioned problems in traditional CAD/CAM systems, several studies have suggested that the character of parametric curves, NURBS, directly generates a path without segmentation curve processing [9-14]. Hence, the linear and arc segments are fitted into a NURBS curve in real-time for the motion controller. NURBS curve fitting processes [15–18] have nearly all been implemented in non-real-time systems and thus computation time is generally not a critical issue. However, for real-time systems, the computation efficiency of the NURBS curves is a critical concern. Therefore, Yeh and Su [19] proposed an on-line NURBS curve fitting process to improve the quality and efficiency of CNC machining. Nevertheless, the parameters of the order of NURBS and the amount of corresponding control points are used to globally fit the NURBS curve, because the NURBS curve is generated by a knot vector *U*, an order of NURBS *p*, basis functions *N*, and control points *P*, providing good flexibility for representing freeform curves. However, this still cannot dramatically improve the fitting efficiency and quality in the local area.

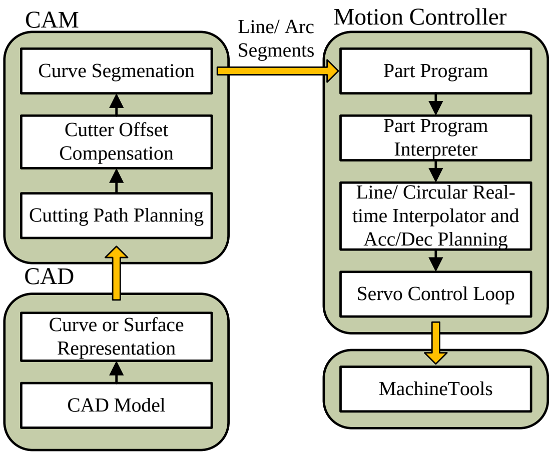

Figure 1. Traditional CAD/CAM and motion system.

To solve the problems confronting traditional motion controllers, many small segments are fitted into the NURBS curve, and the servo command corresponding to an NURBS curve is generated by parametric curve interpolations which rely mainly on parameter approximation methods using Taylor's second-order expansions. It interpolates the received motion commands through its mechanical properties and generates reference commands for servo control loop.

This paper proposes a fast scheme for coplanar NURBS curve fitting in a motion controller for real-time applications. The NURBS curves fitting process includes four stages: (1) reading discrete points, (2) linear segment filter, (3) curve reshaping, (4) and curve fitting considering both the errors of position and first derivative. Firstly, many ordered discrete points from the CAD/CAM system are read. The linear motion segments are then filtered out to execute linear interpolation. Suitable curve segments are separated and discrete points are adjusted to reshape the curve. In the final stage, both the position errors and their first derivative are considered to avoid over fitting of the NURBS curve. Finally, the simulations tested the NURBS curve fitting process performance.

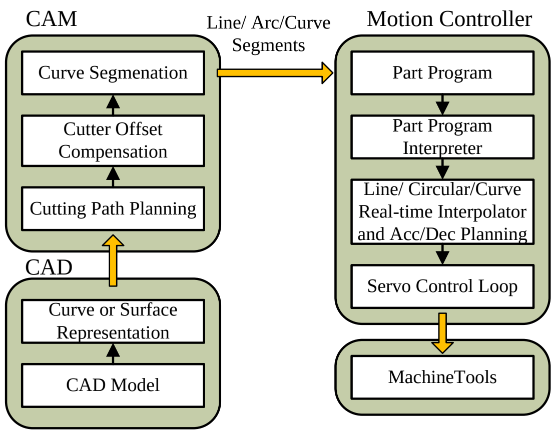

Figure 2. Modern CAD/CAM and motion system.

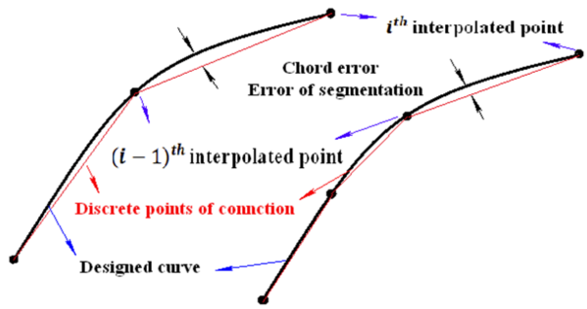

Figure 3. Segmentation process in CAD systems.

NURBS Curve

The NURBS curve is a parametric curve which is easy to program and to present, so NURBS is commonly used in engineering problems [16]. NURBS curves are calculated by

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

Here

$$R_{i,p}(u)=\frac{N_{i,p}(u)W_i}{\sum_{i=0}^{n} N_{i,p}(u)W_i}$$

where *i* is the index-of-sum, *n* is the upper index-of-sum, *(n+1)* is the number of control points, P_i are the control points for shaping NURBS curves, W_i are the

Chin-Sheng Chen received the Ph. D. degrees in mechanical engineering from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., in 1999. He was a researcher of Sintec Technology Co. Ltd. during 1999 to 2000 and a R&D manager of TECO Electric & Machinery Co. Ltd. from 2000 to 2002. In 2002, he joined the Gradulate Institute of Automation Technology, National Taipei University of Technology, Taipei, Taiwan, R.O.C., as an Assistant Professor. Presently he is a professor and director of Graduate Institute of Automation Technology at National Taipei University of Technology. His research interests include motion control, mechantronics and machine vision. He received the Outstanding Research Award from the College of Mechanical and Electrical Engineering, Taipei TECH in 2013 and 2014. Prof. Chen has published over 150 journal and conference papers and book chapters on the research.

Cheng-Yin Tsai received the master degree in the Graduate Institute of Automation Technology in 2011 at National Taipei University of Technology. His thesis mainly focused on NURBS curve fitting and interpolation of CNC motion controller. He obtained the HIWIN THESIS AWARD in 2012. He has been employed in the MIRLE Automation Co. from 2011 to 2014 and in the Premtek Co. from 2014 to 2016. Now, he is an enginner in the Airoha Technology Co.

corresponding weights of P_i, $N(u)_{i,p}$ are the basis functions or blending functions of the NURBS curve, p is the degree of $N(u)_{i,p}$, and $R(u)_{i,p}$ are the rational basis functions of the NURBS curve. The recurrence formulas for computing $N(u)_{i,p}$ can be defined as

$$N_{i,0}(u)=\begin{cases}1, & for \quad u_i \le u \le u_{i+1} \\ 0, & else\end{cases}$$
$$N_{i,p}(u)=\frac{u-u_i}{u_{i+p-1}-u_i}N_{i,p-1}(u)+\frac{u_{i+p}-u}{u_{i+p}-u_{i+1}}N_{i+1,p-1}(u) \quad (2)$$

where u is the NURBS knot. The knot vector U formed by a series of knots is given by Eq. (3).

$$U=\{u_0,u_1,\ldots\ldots,u_{i+p+1}\} \quad (3)$$

Equation (4) describes a chord length method to compute the parameter values $\overline{u_i}$ for a given set of data points Q_i, $i=0,1,\ldots\ldots,m$. Let d be the total chord length

$$d=\sum_{i=1}^{m}|Q_i-Q_{i-1}|$$
$$\overline{u_0}=0, \overline{u_m}=1$$
$$\overline{u_i}=\overline{u}_{i-1}+\frac{|Q_i-Q_{i-1}|}{d}, i=1,2,\ldots,m-1 \quad (4)$$

The parameters $\{\overline{u_i}\}$ can be computed using Eq. (4). The placement of the knots should reflect the distribution of the $\{\overline{u_i}\}$. Let $g=(m+1)/(n-p+1)$; if g is a positive real number, denote by $i=int(g)$, the largest integer such that $i \leqq g$. We need a total of $(n+p+2)$ knots; there are $(n-p)$ internal knots, and $(n-p+1)$ internal knot spans [16]. Therefore, all the knots are given by

$$g=\frac{m+1}{n-p+1}$$

$$u_0=\ldots=u_p=0, \quad u_{m-p}=\ldots=u_m=1$$
$$i=\text{int}(jg) \ , \ \alpha=jg-i \quad (5)$$
$$u_{p+j}=(1-\alpha)\overline{u}_{i-1}+\alpha\overline{u}_i, \ j=1,2,\cdots n-p$$

NURBS interpolation distributes the NURBS curve into the motion command of each axis. The command considers the sampling time, feed rate and the current geometric status of any parametric curve.

Proposed NURBS Curve Fitting

Figure 4 shows the flow chart of proposed NURBS curve fitting. After reading the discrete points, the linear segment filter is used to filter out the linear trajectory. The discrete points among the curve segment could then be adjusted according to the error boundary to obtain a smooth curve. Finally, both the position errors and their first derivative are considered to avoid over fitting of the NURBS curve. After the fitted NURBS curve is obtained, the corresponding interpolation for each interpolated time is further processed to obtain the motion command for each axis.

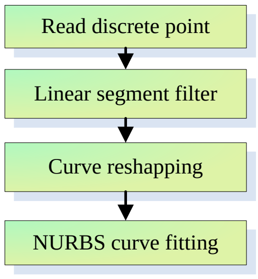

Figure 4. NURBS Curve Fitting.

Linear segment filter

As shown in Figure 5, whether the discrete points from A to F are grouped into a line segment can be determined by the tolerance E_c. The detailed procedures are described as follows:

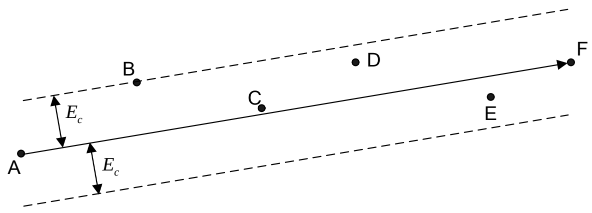

Figure 5. Discrete points located at the neighbor of line $\overline{AF}$.

Step 1: Chose points A, B, and C, as shown in Fig. 6(a). If the condition $\overline{BM_1} \le E_c$ is met, the point B is located on line $\overline{AC}$. We then choose the next point D.
Step 2: Chose points A, B, and D, as shown in Fig. 6(b). If the conditions $\overline{BM_2} \le E_c$ and $\overline{BM_3} \le E_c$ are met, the points B and C are located on line $\overline{AD}$. We then choose the next point E.
Step 3: Continuously move to the next discrete point. In this case, when we choose the final point F, and all the distances from points B, C, D, and E to line $\overline{AF}$ are smaller than E_c, then all the discrete points can be clustered as one line segment $\overline{AF}$, as shown in Fig. 6(c).

Curve reshaping

Following the linear segment filter, points not belonging to the line segment are classified as the curve segment. This study proposes a reshaping method to adjust the corresponding discrete points to obtain smooth NURBS curve fitting. Figure 7 shows discrete point adjustment for a curve segment. As shown in Fig. 8, three points are extracted for adjustment. The distance from point B to line $\overline{AC}$ is calculated. If the condition $\overline{BM} \leq E_c$ is met, the point B is adjusted to be the middle M of line $\overline{AC}$. Otherwise, B point is adjusted to point M^* along the $\overrightarrow{BM}$ with distance E_c.

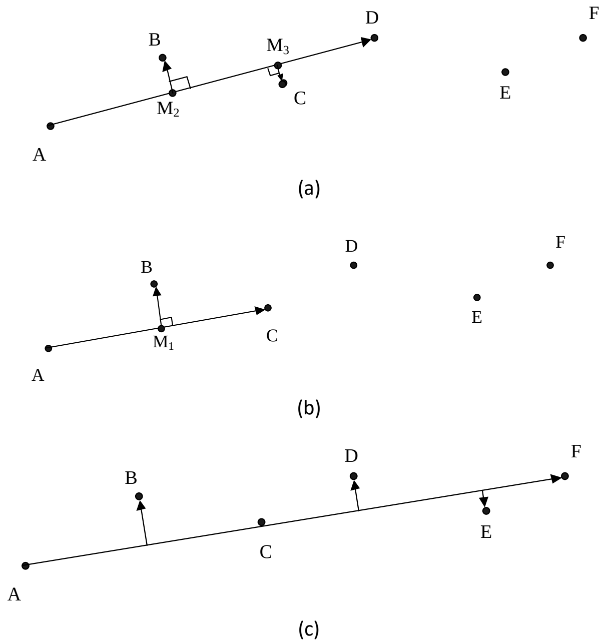

Figure 6. Linear segment filter verification.

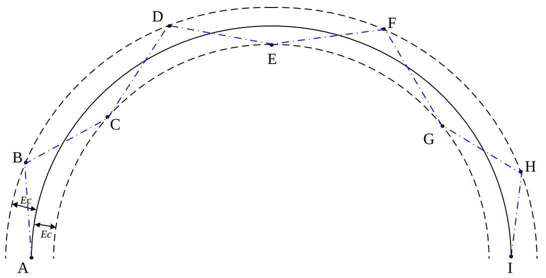

Figure 7. Curve reshaping for a curve segment.

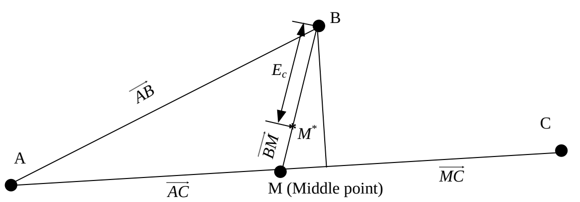

Figure 8. Curve reshaping.

NURBS curve fitting

The least-squares approach is typically used for fitting a set of data points to a NURBS curve. However, as shown in Eqs. (1)~(3), the approximate distributions u_i and W_i must be applied prior to least-squares fitting. The chord length method is used to obtrain the parameter values $\bar{u}_j$ by Eq. (4), and j=0,1,......,m and the knot vector are selected by Eq. (5). Equation (6) is the fitted form of the NURBS curve by applying the least-squares fitting.

$$Q_k \approx C(\bar{u}_j) = \frac{\sum_{i=0}^{n} N_{i,p}(\bar{u}_k) W_i P_i}{\sum_{i=0}^{n} N_{i,p}(\bar{u}_k) W_i} = \sum_{i=0}^{n} R_{i,p}(\bar{u}_k) P_i \quad (6)$$

$$k = 0,1,.....,m$$

Here, Q_k represents the given discrete points, $\bar{u}_k$ is the corresponding parameter values of the fitted NURBS curves, and *(m+1)* is the number of discrete points. The NURBS curve fitting method provides an algorithm for obtaining the appropriate distributions of knots, weights, order, upper index-of-sum, and control points such that the errors between the data points and fitted points are confined within the desired range.

To avoid the nonlinear problem, we set $W_i = 1$ for *i=0,1,...,n.* The NURBS curves shown in Eq. (1) are converted to B-spline curves as follows

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u) P_i \quad (7)$$

where $N_{i,p}(u)$ are the p^{th}-degree B-spline basis functions.

Equation (8) summarizes the fitting method by applying the B-spine curves

$$Q_k \approx C(\bar{u}_k) = \sum_{i=0}^{n} N_{i,p}(\bar{u}_k) P_i \quad (8)$$

By applying the B-spline curves, the error e_k between Q_k and fitted points $C(\bar{u}_k)$ is obtained as

$$e_k = \left| Q_k - C(\bar{u}_k) \right| = \left| Q_k - \sum_{i=0}^{n} N_{i,p}(\bar{u}_k) P_i \right| \quad (9)$$

Similarly, the error e_k' between Q_k' and fitted points $C'(\bar{u}_k)$ is obtained as

$$e_k' = \left| Q_k' - C'(\bar{u}_k) \right| = \left| Q_k' - \sum_{i=0}^{n} N_{i,p}'(\bar{u}_k) P_i \right| \quad (10)$$

The least-squares approach is typically used to obtain the optimal distributions of control points for *k=0,1,...,m.* In this study, the least square error of the

B-Spline curve, the discrete points and the corresponding first derivative are considered to avoid over fitting. The new cost function is defined as follows

$$E=\frac{1}{2}\sum_{0}^{m}\left|Q_i-C(u_i)\right|^2+\frac{\lambda}{2}\sum_{0}^{m}\left|Q_i'-C'(u_i)\right|^2 \qquad (11)$$

Let E be an estimation function; the control point number is $(n+1)$. The sets of control points may be obtained by ensuring the partial derivative of each control point is zero and the evaluation function E is minimized.

$$\begin{aligned}
\frac{\partial E}{\partial P_0}&=-\sum_{i=0}^{m}\left|Q_i-C(u_i)\right|\left(N_{0,k}(u_i)\right)-\lambda\sum_{i=0}^{m}\left|Q_i'-C'(u_i)\right|\left(N_{0,k}'(u_i)\right)=0\\
\frac{\partial E}{\partial P_1}&=-\sum_{i=0}^{m}\left|Q_i-C(u_i)\right|\left(N_{1,k}(u_i)\right)-\lambda\sum_{i=0}^{m}\left|Q_i'-C'(u_i)\right|\left(N_{1,k}'(u_i)\right)=0\\
&\vdots \qquad\qquad \vdots\\
\frac{\partial E}{\partial P_n}&=-\sum_{i=0}^{m}\left|Q_i-C(u_i)\right|\left(N_{n,k}(u_i)\right)-\lambda\sum_{i=0}^{m}\left|Q_i'-C'(u_i)\right|\left(N_{n,k}'(u_i)\right)=0
\end{aligned} \qquad (12)$$

Equation (12) can be reorganized as follows:

$$\begin{aligned}
&\sum_{i=0}^{m}C(u_i)\left(N_{0,k}(u_i)\right)+\lambda\sum_{i=0}^{m}C'(u_i)\left(N_{0,k}'(u_i)\right)\\
&=\sum_{i=0}^{m}Q_i\left(N_{0,k}(u_i)\right)+\lambda\sum_{i=0}^{m}Q_i'\left(N_{0,k}'(u_i)\right)\\
&\sum_{i=0}^{m}C(u_i)\left(N_{1,k}(u_i)\right)+\lambda\sum_{i=0}^{m}C'(u_i)\left(N_{1,k}'(u_i)\right)\\
&=\sum_{i=0}^{m}Q_i\left(N_{1,k}(u_i)\right)+\lambda\sum_{i=0}^{m}Q_i'\left(N_{1,k}'(u_i)\right)\\
&\qquad\vdots \qquad\qquad\qquad \vdots\\
&\sum_{i=0}^{m}C(u_i)\left(N_{n,k}(u_i)\right)+\lambda\sum_{i=0}^{m}C'(u_i)\left(N_{n,k}'(u_i)\right)\\
&=\sum_{i=0}^{m}Q_i\left(N_{n,k}(u_i)\right)+\lambda\sum_{i=0}^{m}Q_i'\left(N_{n,k}'(u_i)\right)
\end{aligned} \qquad (13)$$

Eq. (13) can be simplified as a matrix of linear equations

$$\left(\overline{N}^T\,\overline{N}\overline{P}+\lambda\overline{N'}^T\,\overline{N'}\overline{P}\right)=\left(\overline{N}^T\,\overline{Q}+\lambda\overline{N'}^T\,\overline{Q'}\right) \qquad (14)$$

Where

$$\overline{N'}=\begin{bmatrix}N_{0,k}'(u_0) & N_{1,k}'(u_0) & \cdots & N_{n,k}'(u_0)\\ N_{0,k}'(u_1) & N_{1,k}'(u_1) & \cdots & N_{n,k}'(u_1)\\ \vdots & \vdots & \ddots & \vdots\\ N_{0,k}'(u_m) & N_{1,k}'(u_m) & \cdots & N_{n,k}'(u_m)\end{bmatrix} \qquad (15)$$

Through Eq. (14), the optimal distributions of the control points are obtained as

$$\overline{P}=\left(\overline{N}^T\,\overline{N}+\lambda\overline{N'}^T\,\overline{N'}\right)^{-1}\left(\overline{N}^T\,\overline{Q}+\lambda\overline{N'}^T\,\overline{Q'}\right) \qquad (16)$$

where $\overline{N}^T$ is the transpose of $\overline{N}$. Since the coefficient matrix $\overline{N}^T\overline{N}$ is totally positive, Eq. (16) can be solved by Gaussian elimination.

The penalty factor λ in Eq. (11) could be used to penalize the cost with first derivative of position error to avoid over fitting. Figure 9 shows the flow chart of our proposed optimum method of NURBS curve fitting. The process is as follows:

Step 1: Set the parameters of fitting restrictions $n_{max}, p_{max}, \lambda_{max}, e_{max}, e'_{max}$. The parameters n_{max} and p_{max} respectively confine the search range of upper index-of-sum and order of NURBS.

Step 2: Set the initial values of NURBS order, number of control points and penalty factor respectively to n_0, p_0 and λ_0.

Step 3: Obtain the distributions of control points P_i by the least-squares fitting method, as shown in Eq. (16). Thus the NURBS curve is obtained.

Step 4: Compute e'_k by using Eq. (10). If $max\{e'_k\}$ is smaller than e'_{max}, go to Step 7.

Step 5: Increase the penalty factor λ.

Step 6: If λ is smaller than λ_{max}, go to Step 3. Otherwise, go to step 8.

Step 7: Compute e_k by using Eq. (9). If $max\{e_k\}$ is smaller than e_{max}, go to Step 13. Otherwise go to Step 8.

Step 8: Increase the control number n.

Step 9: If n is smaller than n_{max}, go to Step 3. Otherwise, go to step 10.

Step 10: Increase the degree of NURBS p and reset the control number n as n_0.

Step 11: If p is smaller than p_{max}, go to Step 3. Otherwise, go to Step 1.

Step 12: If $max\{e'_k\} \le e'_{max}$ and, $max\{e_k\} \le e_{max}$, the fitting specifications are satisfied by the fitted NURBS curve. Thus, the fitting process will be finished.

Step 13: Output the fitted NURBS curve.

Simulation results

Figure 10 shows a 3D CAD model used to test the curve fitting quality. There are five tool paths shown on the mask 3D surface. Here, one 3D tool path is fitted into one NURBS curve to verify the curve fitting quality for our proposed algorithm.

Here we chose 3 paths (423, 1429 and 4263) to indicate fitting performance. Figures 11, 12 and 13 respectively show the curvature radii of paths 423, 1429 and 4263. The red and blue lines respectively show the results of direct fitting and proposed fitting. The proposed method clearly results in a small curvature for all the curves, implying that the proposed method can obtain a smooth curve and guarantee a small error of

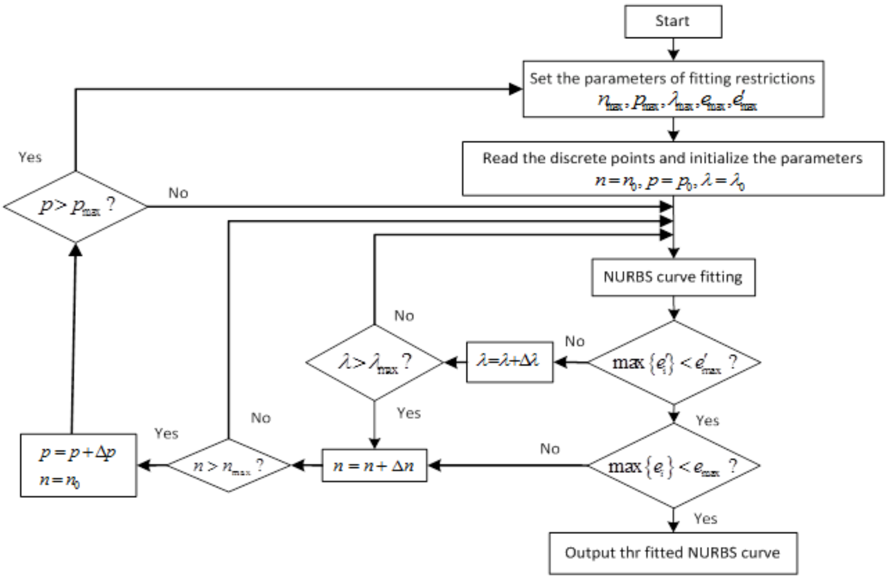

Figure 9. The process of curve fitting.

position and first derivative. The detail fitting results are shown in Tables 1 to 3 for different curves.

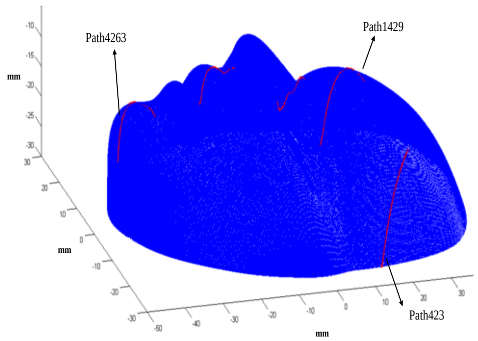

Figure 10. 3D freeform surface and the corresponding tool path.

All simulations were performed on a personal computer with an Intel Core i7 3.4 GHz CPU and 8 GB of memory using Visual Studio 2010. In this study, the whole paths of 423, 1429 and 4263 are respectively fitted as NURBS curves. If NURBS curve fitting is applied in real-time restrictions, the limited discrete points should be segmented as the threshold for each NURBS curve. Computation times are shown in Table 4.

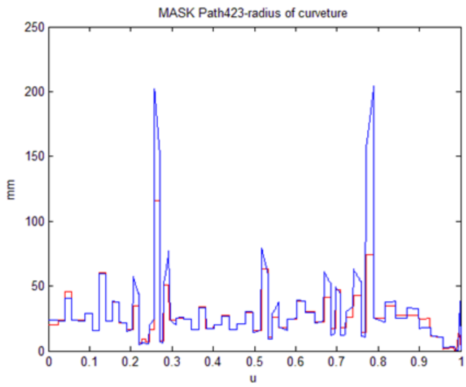

Figure 11. Radius of curvature of path 423.

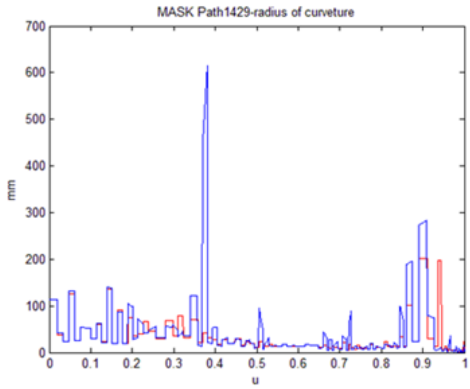

Figure 12. Radius of curvature of path 1429.

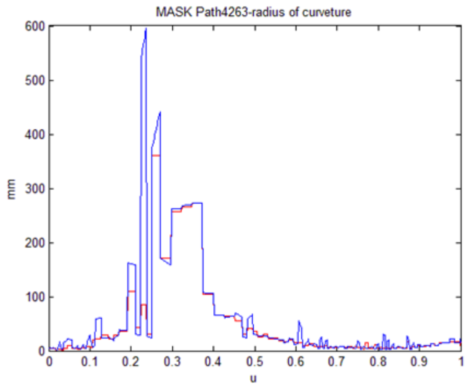

Figure 13. Radius of curvature of path 4263.

Conclusion

The proposed optimum NURBS fitting algorithm simultaneously improves the fitting quality and calculation efficiency. The NURBS curves fitting process includes four stages: (1) reading discrete points, (2) linear segment filter, (3) curve reshaping, (4) curve fitting considering both the errors of position and first derivative. Many ordered discrete points from the CAD/CAM system are read. The linear motion segments are then filtered out for linear interpolation. Suitable curve segments are separated and discrete points are properly adjusted to reshape the curve. In the final stage, both the errors of position and its first derivative are considered to avoid over fitting the NURBS curve. Simulation results indicate that the $max\{e_k'\}$ significantly decreases to 45%, 25% and 50%, respectively, for paths 423, 1429 and 4263 in Tables 1, 2 and 3. The dramatically decreased values indicate that the proposed method simultaneously provides better fitting performance for both the errors of position and first derivative to avoid the over fitting.

Table 1. Fitting results of path 423.

	Direct fitting	Proposed method
Number of control point	58	57
$max\{e_k\}$	$0\mu m$	$1.478\mu m$
$max\{e_k'\}$	21.734964 $mm/\Delta u$	11.901092 $mm/\Delta u$

Table 2. Fitting results of path 1429.

	Direct fitting	Proposed method
Number of control point	101	80
$max\{e_k\}$	$0\mu m$	$1.324\mu m$
$max\{e_k'\}$	5.922947 $mm/\Delta u$	4.463487 $mm/\Delta u$

Table 3. Fitting results of path 4263.

	Direct fitting	Proposed method
Number of control point	101	76
$max\{e_k\}$	$0\ \mu m$	$1.055\ \mu m$
$max\{e_k'\}$	10.538437 $mm/\Delta u$	5.275721 $mm/\Delta u$

Table 4. Computation time of NURBS curve fitting of different paths (msec).

Path 423	Path 1429	Path 4263
1.46	2.46	2.43

References

p[1] G. C. Loney and T. M. Ozsoy, "NC machining of free-form surfaces," *Computer-Aided Design*, vol. 19, no. 2, pp. 85–90, March 1987,. doi:10.1016/S0010-4485(87)80050-7

[2] G. W. Vickers and C. Bradley, "Curved surface machining through circular arc interpolation," *Computers in Industry*, vol. 19, no. 3, pp. 329–337, 1992. doi:10.1016/0166-3615(92)90070-4

[3] S. Bedi, I. Ali, and N. Quan, "Advanced interpolation techniques for CNC machines," *Trans. of the ASME J. of Engineering for Industry*, vol. 115, no. 3, pp. 329–336, Aug. 1993. doi:10.1115/1.2901668

[4] M. Shpitalni, Y. Koren, and C.-C. Lo, "Realtime curve interpolators," *Computer-Aided Design*, vol. 26, no. 11, pp. 832–838, Nov. 1994. doi: 10.1016/0010-4485(94)90097-3

[5] B. Bahr, X. Xiao, and K. Krishnan, "A real-time scheme of cubic parametric curve interpolations for CNC systems," *Computers in Industry*, vol. 45, no. 3, pp. 309-317, July 2001. doi:10.1016/S0166-3615(01)00087-2

[6] D. C. H. Yang and T. Kong, "Parametric interpolator versus linear interpolator for precision CNC machining," *Computer-Aided Design*, vol. 26, no. 3, pp. 225–234, March 1994. doi:10.1016/0010-4485(94)90045-0

[7] S.-S. Yeh and P.-L. Hsu, "Adaptive-feedrate interpolation for parametric curves with a confined chord error," *Computer -Aided Design*, vol. 34, no. 3, pp. 229-237, March 2002. doi:10.1016/S0010-4485(01)00082-3

[8] S.-S. Yeh and P.-L. Hsu, "Speed-controlled interpolator for machining parametric curves," *Computer- Aided Design*, vol. 31, no. 5, pp. 349-357., April 1999 doi:10.1016/S0010-4485(99)00035-4

[9] H. Park, "Error-bounded biarc approximation of planar curves," *Computer-Aided Design* , vol. 36, no. 12, pp. 1241–1251, Oct. 2004. doi:10.1016/j.cad.2004.01.001

[10] L. Piegl and W. Tiller, "Biarc approximation of NURBS curves," *Computer-Aided Design*, vol. 34, no. 11, pp. 807–814, Sept. 2002. doi: 10.1016/S0010-4485(01)00160-9

[11] C.-J. Ong, Y.-S. Wong, H.-T. Loh, and X.-G. Hong, " Optimization approach for biarc curve-fitting of B-spline curves," *Computer-Aided Design*, vol. 28, no. 12, pp. 951–959, Dec. 1996. doi:10.1016/0010-4485(96)00028-0

[12] C.-W. Cheng, M.-C. Tsai, and J. Maciejowski, "Accurate feedrate control of CNC machine tools along NURBS curves," in proceeding of *the 43rd IEEE Conference on Decision and Control*, Nassau, Bahamas, Dec. 14-17, 2004. doi: 10.1109/CDC.2004.1428932

[13] C.-W. Cheng and M.-C. Tsai, "Real-time variable feed rate NURBS curve interpolator for CNC

machining," *International Journal of Advanced Manufacturing Technology*, vol. 23, no. 11-12, pp. 865-873, June 2004. doi:10.1007/s00170-003-1732-9

[14] J. Park, S. Nam, and M. Yang, "Development of a real-time trajectory generator for NURBS interpolation based on the two-stage interpolation method," *The International Journal of Advanced Manufacturing Technology*, vol. 26, no. 4, pp. 359-365, August 2005. doi:10.1007/s00170-003-1998-y

[15] D. S. Meek and D. J. Walton, "Approximating quadratic NURBS curves by arc splines," *Computer-Aided Design*, vol. 25, no. 6, pp. 371–376, June 1993. doi:10.1016/0010-4485(93)90032-J

[16] L. Piegl and W. Tiller, *The NURBS Book.* New York, Springer, 1997.

[17] S.-M. Hu, Y.-F. Li, T. Ju, and X. Zhu, "Modifying the shape of NURBS surfaces with geometric constraints," *Computer-Aided Design*, vol. 33, no. 12, pp. 903–912, Oct. 2001. doi: 10.1016/S0010-4485(00)00115-9

[18] W. Ma and J. P. Kruth, "Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces," *Computer-Aided Design*, vol. 27, no. 9, pp. 663–675, Sept. 1995. doi:10.1016/0010-4485(94)00018-9

[19] S.-S. Yeh and H.-C. Su, "Implementation of online NURBS curve fitting process on CNC machines," *The International Journal of Advanced Manufacturing Technology*, vol. 40, no. 5-6, pp. 531-540, Jan. 2009. doi:10.1007/s00170-007-1361-9