ELSEVIER

Computer-Aided Design 34 (2002) 229–237

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

Adaptive-feedrate interpolation for parametric curves with a confined chord error

Syh-Shiuh Yeh, Pau-Lo Hsu*

Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu, 300 Taiwan

Received 25 January 2000; revised 27 December 2000; accepted 6 January 2001

Abstract

Recently, modern manufacturing systems have been designed which can machine arbitrary parametric curves while greatly reducing data communication between CAD/CAM and CNC systems. However, a constant feedrate and chord accuracy between two interpolated points along parametric curves are generally difficult to achieve due to the non-uniform map between curves and parameters. A speed-controlled interpolation algorithm with an adaptive feedrate is proposed in this paper. Since the chord error in interpolation depends on the curve speed and the radius of curvature, the feedrate in the proposed algorithm is automatically adjusted so that a specified limit on the chord error is met. Both simulation and experimental results for non-uniform rational B-spline (NURBS) examples are provided to verify the feasibility and precision of the proposed interpolation algorithm. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords: Parametric curve; Speed-controlled interpolation; Chord error; NURBS

1. Introduction

In modern CAD/CAM systems, profiles for parts such as dies, vanes, aircraft models, and car models are usually represented in parametric forms. Since conventional CNC machines only provide linear and circular arc interpolators, the CAD/CAM systems have to segment a curve into a huge number of small linearized segments and send them to CNC systems. Such linearized-segmented contours processed on CNC systems are undesirable in real applications for the following reasons:

- the transmission errors between CAD/CAM and CNC systems, i.e. lost data and noise perturbation, may be unavoidable for huge amounts of data;
- the discontinuity of segmentation deteriorates surface accuracy; and
- the motion speed becomes non-uniform and unsmooth.

In modern manufacturing processes, only the parameters of curves or profiles are required to be efficiently transferred among CAD/CAM/CNC systems, as shown in Fig. 1. Shpitalni et al. [1] proposed the transfers of curve segments between CAD and CNC systems, and Bedi et al. [2] proposed a B-spline curve and B-spline surface interpolation algorithm.

There are many different formats for parametric curve representation, such as Bezier, B-spline, cubic spline, and NURBS (non-uniform rational B-spline). The general parameter iteration method used is $u_{i+1} = u_i + \Delta(u_i)$ where u_i is the present parameter, u_{i+1} is the next parameter, and $\Delta(u_i)$ is the incremental value. The interpolated points are calculated by substituting u_i into the corresponding mathematical model to recover the originally designed curves. Since the cutter moves in a straight path between contiguous interpolated points, two position errors may occur during parametric curve motion: (a) radial error; and (b) chord error [3] as shown in Fig. 2. The radial error is the perpendicular distance between the interpolated points and the parametric curve, and the chord error is the maximum distance between the secant $\overline{CD}$ and the secant arc $\overset{\frown}{AB}$. Basically, the radial error is caused by the rounding error of computer systems. With the rapid development of microprocessors for high precision applications, the radial error is no longer a major concern. The chord error is thus the main concern of this paper and we will propose an interpolation algorithm that limits the chord error to a specified tolerance.

Several researchers have developed interpolation algorithms to deal with the parametric curves. Bedi et al. [2] set $\Delta(u_i)$ as a constant in the uniform interpolation algorithm. To reduce speed fluctuation during the interpolation

* Corresponding author. Tel.: +886-3-5712-121; fax: +886-3-5715-998.
E-mail address: plhsu@cc.nctu.edu.tw (P.-L. Hsu).

PII: S0010-4485(01)00082-3

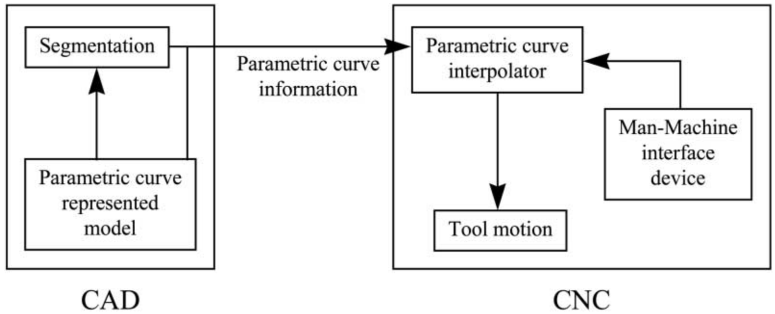

Fig. 1. The machining systems with parameters transmission.

process, Houng and Yang [4] and Shpitalni et al. [1] developed first-order approximation interpolation algorithms using Euler and Taylor's expansions, respectively. These first-order approximation interpolation algorithms provide uniform curve speed during the interpolation process. Furthermore, the second-order approximation and speed-controlled interpolation algorithms proposed by Yang and Kong [5] and Yeh and Hsu [6], respectively, yield more precise results. However, these algorithms do not consider chord error explicitly during the interpolation process.

In this paper, we propose an interpolation algorithm that confines the chord error within a specified tolerance range during the interpolation process. Since the chord error is closely related to the curve speed and the radius of curvature, the relations among chord error, curve speed, and radius of curvature are identified in the algorithm. The circular approximation is adopted for general parametric curves in the proposed method. Then, the adaptive curve speed based on the chord error of the approximated circular curve is derived and applied to the speed-controlled parameter iteration algorithm to make the chord error within the specified tolerance. To demonstrate the performance of the proposed chord error-controlled interpolator, a NURBS parametric curve example using a personal computer is also provided in this paper.

2. Speed-controlled interpolation

There are many interpolation algorithms that consider speed uniformity [1,4–6]. In particular, the first-order approximation interpolation algorithm is adequately applied to generate commands in general motion systems. Suppose $C(u)$ is the parametric curve function and the time function u is the curve parameter with $u(t_i) = u_i$ and $u(t_{i+1}) = u_{i+1}$. By using a Taylor expansion, the approximation up to the first derivative is

$$u_{i+1} = u_i + \left.\frac{\mathrm{d}u}{\mathrm{d}t}\right|_{t=t_i} \cdot (t_{i+1} - t_i) + H.O.T \tag{1}$$

Since the curve speed $V(u_i)$ can be represented as

$$V(u_i) = \left\| \frac{\mathrm{d}C(u)}{\mathrm{d}t} \right\|_{u=u_i} = \left\| \frac{\mathrm{d}C(u)}{\mathrm{d}u} \right\|_{u=u_i} \cdot \left.\frac{\mathrm{d}u}{\mathrm{d}t}\right|_{t=t_i}$$

the first derivative of u with t is then

$$\left.\frac{\mathrm{d}u}{\mathrm{d}t}\right|_{t=t_i} = \frac{V(u_i)}{\left\| \frac{\mathrm{d}C(u)}{\mathrm{d}u} \right\|_{u=u_i}} \tag{2}$$

Let the sampling time in interpolation be T_s seconds, that is $t_{i+1} - t_i = T_s$. The first-order approximation interpolation algorithm is thus obtained by substituting Eq. (2) into Eq. (1). By neglecting the higher-order terms, the interpolation algorithm in Eq. (1) can be expressed as:

$$u_{i+1} = u_i + \frac{V(u_i) \cdot T_s}{\left\| \frac{\mathrm{d}C(u)}{\mathrm{d}u} \right\|_{u=u_i}} \tag{3}$$

where $V(u_i)$ can be the feedrate command or the desired curve speed in a general machining process.

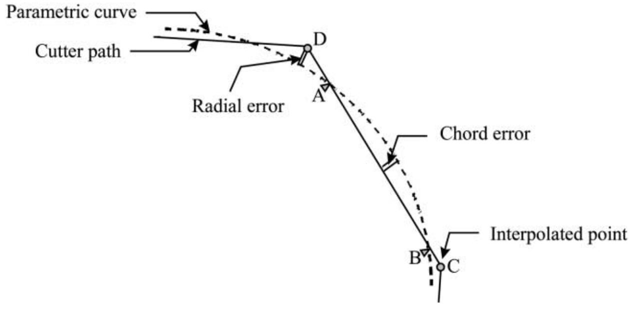

Fig. 2. Radial and the chord errors.

3. Adaptive feedrate and chord error

Although the first-order approximation interpolation algorithm given by Eq. (3) makes the curve speed almost equal to the desired value$V(u_i)$, the chord error, as shown in Fig. 2, exists and it may become unacceptable if an improper curve speed $V(u_i)$ is given. To keep the chord error within a tolerance range, the curve speed $V(u_i)$ has to be changed adaptively depending on the curvature during the interpolation process. To determine the relation between the chord error and the curve speed, the circular approximation is thus applied.

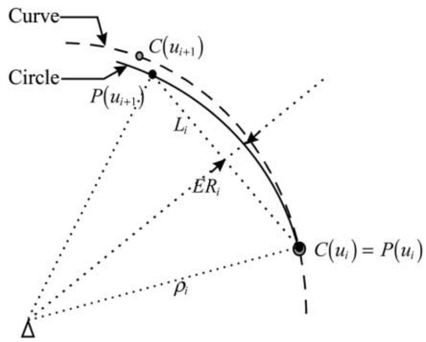

Fig. 3. Estimation of the next interpolated point.

Suppose the sectional curve of $u \in [u_i \quad u_{i+1})$ is an arc of a circle with the radius ρ_i at parameter $u = u_i$ as shown in Fig. 3. where, $\rho_i = 1/K_i$: the radius of curvature at parameter $u = u_i$, and the curvature K_i is

$$K_i = \frac{\left.\frac{\mathrm{d}C_x(u)}{\mathrm{d}u}\cdot\frac{\mathrm{d}^2C_y(u)}{\mathrm{d}u^2} - \frac{\mathrm{d}C_y(u)}{\mathrm{d}u}\cdot\frac{\mathrm{d}^2C_x(u)}{\mathrm{d}u^2}\right|_{u=u_i}}{\left\|\frac{\mathrm{d}C(u)}{\mathrm{d}u}\right\|^3_{u=u_i}}$$

where: $P(u_i)$ is the interpolated point on circle at $u = u_i$; $P(u_{i+1})$ is the estimated interpolated point at $u = u_{i+1}$; $C(u_i)$ is the interpolated point on curve at $u = u_i$; and $C(u_{i+1})$ is the interpolated point at $u = u_{i+1}$.

Since, $C(u_i) = P(u_i)$, and defining L_i as $\|P(u_{i+1}) - P(u_i)\| = L_i$, the approximated curve speed $V(u_i)$ is determined as

$$V(u_i) = \frac{L_i}{T_s} \tag{4}$$

and the chord error ER_i is derived as

$$ER_i = \rho_i - \sqrt{\rho_i^2 - \left(\frac{L_i}{2}\right)^2}, \tag{5}$$

The curve speed $V(u_i)$ corresponding to the chord error ER_i is then derived as

$$V(u_i) = \frac{2}{T_s}\cdot\sqrt{\rho_i^2 - (\rho_i - ER_i)^2} \tag{6}$$

By setting ER_i as the tolerance value of the chord error, $V(u_i)$ is thus obtained as the acceptable curve speed. Since the radius of the curvature ρ_i is much larger than the tolerance value of chord error ER_i, $V(u_i)$ is a real value in this formula.

Eq. (6) implies that the curve speed $V(u_i)$ should be changed adaptively according to the tolerance value of the chord error ER_i and the radius of curvature ρ_i. Since the adaptive curve speed is derived by applying a circular curve approximation, the next actual interpolated point is $C(u_{i+1})$ as shown in Fig. 3. This implies that the chord error after interpolation will be near the tolerance value and its value depends on the radius of curvature. To reduce the influence of chord error fluctuation, a conservative chord error tolerance value is preferred.

Since the curve speed is adaptively changed according to the radius of curvature, the following adaptive-feedrate law is proposed for parametric curves:

$$V(u_i) = \begin{cases} F, & \text{if } \frac{2}{T_s}\cdot\sqrt{\rho_i^2 - (\rho_i - ER_i)^2} > F \\ \frac{2}{T_s}\cdot\sqrt{\rho_i^2 - (\rho_i - ER_i)^2}, & \text{if } \frac{2}{T_s}\cdot\sqrt{\rho_i^2 - (\rho_i - ER_i)^2} \le F \end{cases}, \tag{7}$$

where F is the given feedrate command. If the instantaneous radius of curvature is small enough, the curve will lead to exceeded chord errors and the proposed interpolation algorithm will automatically reduce the feedrate F as

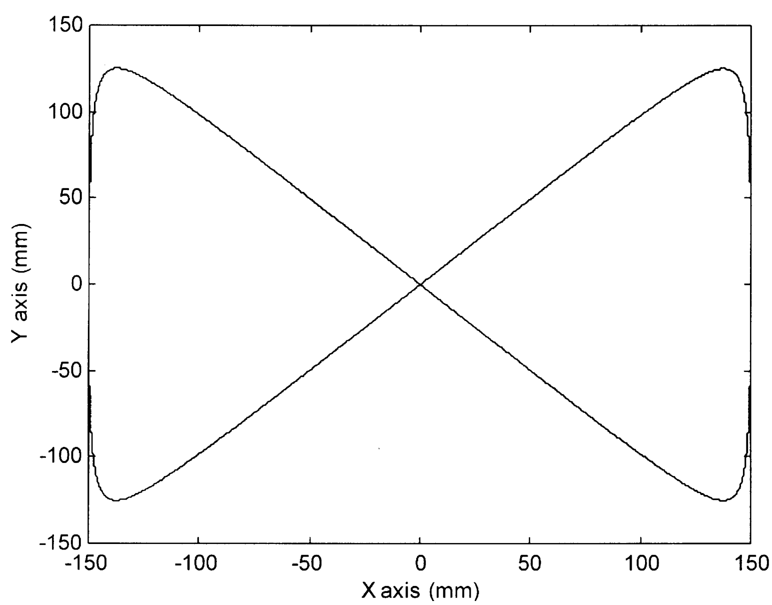

Fig. 4. The NURBS example.

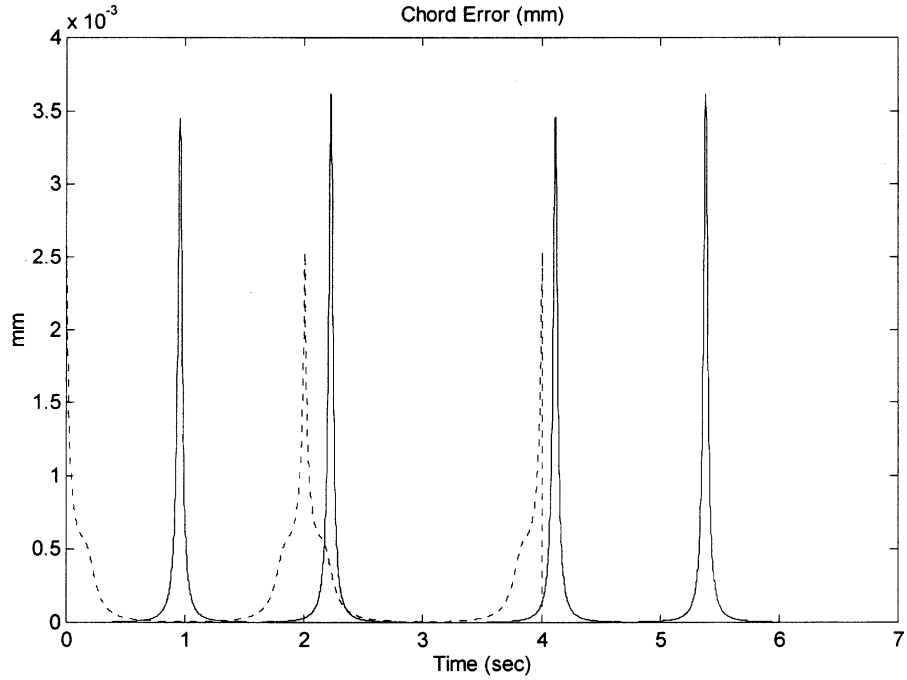

Fig. 5. Chord errors of the uniform-interval NURBS interpolation (dotted, $\Delta u = 0.0005$) and the speed-controlled NURBS interpolation (solid, $F = 12$ m/min).

$2/T_s \cdot \sqrt{\rho_i^2 - (\rho_i - ER_i)^2}$ to meet the specified chord error. Otherwise, the interpolation generates constant-speed operation with the given feedrate F.

To ensure the generated parameter sequence with the desired curve speed, the first-order approximation interpolation algorithm in Eq. (3) is employed here. Thus, the proposed algorithm achieves interpolation accuracy within the specified limits on both the chord error and the speed.

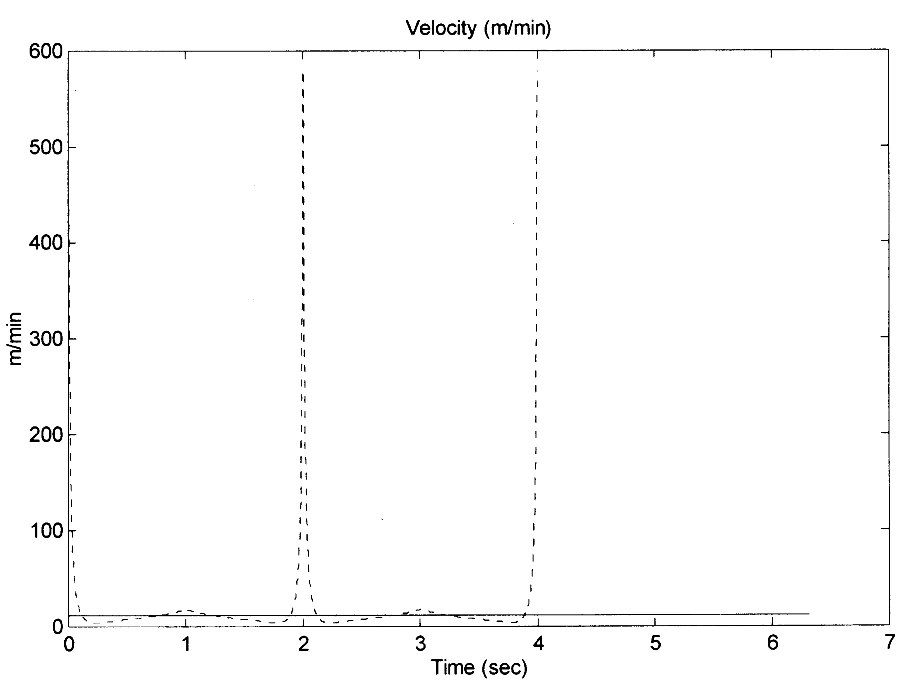

Fig. 6. Curve speed of the uniform-interval NURBS interpolation (dotted, $\Delta u = 0.0005$) and the speed-controlled NURBS interpolation (solid, $F = 12$ m/min).

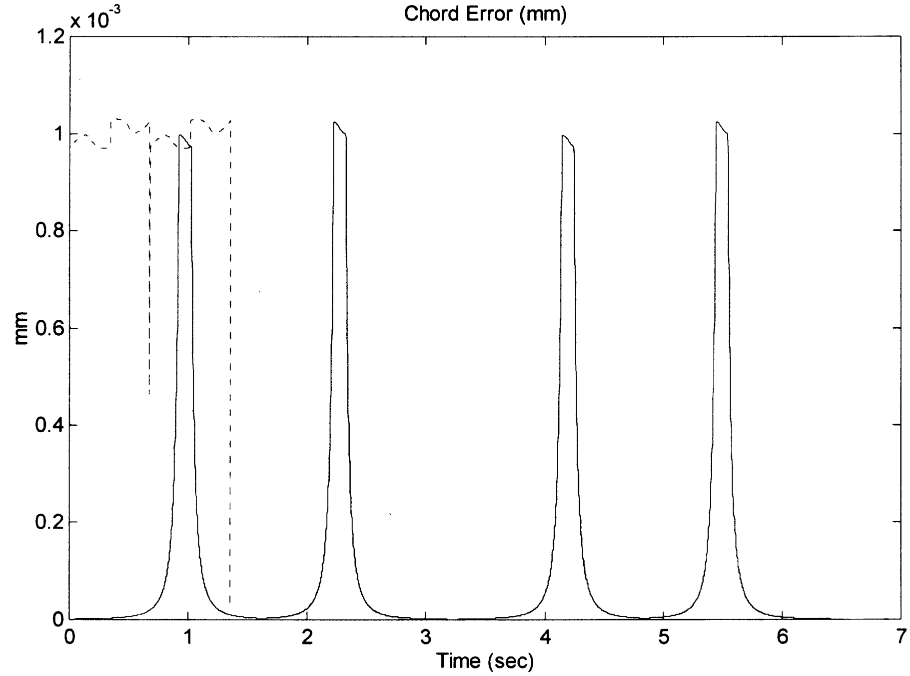

Fig. 7. Chord errors of the adaptive-feedrate NURBS interpolation (solid, $F = 12$ m/min and $ER = 1\mu$) and the interpolation specifying the chord error only (dotted, $ER = 1\mu$).

4. Simulation results

In this simulation, the interpolator is written in Turbo C 2.0 and is executed on a personal computer with both 80MHz and 200MHz CPU. The present interpolator is applied to a NURBS [7] parametric curve with two degrees as shown in Fig. 4.

The control points, weight vector, and knot vector of NURBS for the provided example are assigned as follows:

- The ordinal control points are

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -150 \\ -150 \end{bmatrix}, \begin{bmatrix} -150 \\ 150 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 150 \\ -150 \end{bmatrix}, \begin{bmatrix} 150 \\ 150 \end{bmatrix}, \text{ and } \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{(mm)}.$$

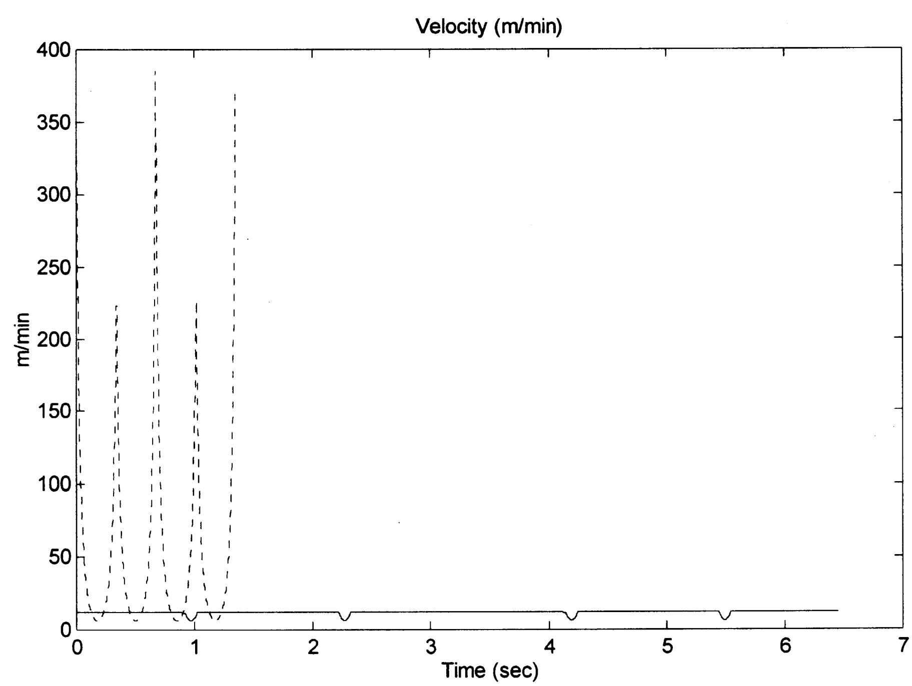

Fig. 8. Curve speed of the adaptive-feedrate NURBS interpolation (solid, $F = 12$ m/min and $ER = 1\mu$) and the interpolation specifying the chord error only (dotted, $ER = 1\mu$).

Table 1
Simulation results of chord errors for different algorithms

Interpolation	Measurement	Chord error (mm)			Curve speed (m/min)			CPU time (μs)		Path time (s)
		MAX	MIN	RMS	MAX	MIN	RMS	80 MHz	200 MHz	
Uniform-interval NURBS	$\Delta u = 0.00001$	1.060×10^{-6}	~ 0	1.689×10^{-7}	12.703	0.082	1.062	11	4	200.004
	$\Delta u = 0.0005$	0.0025	~ 0	4.221×10^{-4}	579.833	4.099	53.033			4.004
	$\Delta u = 0.001$	0.0097	~ 0	0.0017	1.065×10^{3}	8.198	100.321			2.002
Speed-controlled NURBS	$F = 6$ m/min	8.957×10^{-4}	~ 0	1.298×10^{-4}	6.076	5.926	6.0	27	10	12.644
	$F = 12$ m/min	0.0036	~ 0	5.192×10^{-4}	12.310	11.708	12.0			6.324
	$F = 20$ m/min	0.0102	~ 0	0.0014	20.879	19.207	20.001			3.796
Adaptive-feedrate NURBS ($ER = 1\mu$)	$F = 12$ m/min	0.001	~ 0	2.864×10^{-4}	12.263	6.338	11.803	76	25	6.454
	$F = 20$ m/min	0.001	~ 0	4.170×10^{-4}	20.508	6.338	18.973			4.078

- The weight vector is $W = [1 \quad 25 \quad 25 \quad 1 \quad 25 \quad 25 \quad 1]$.
- The knot vector is

$$U = \left[0 \quad 0 \quad 0 \quad \frac{1}{4} \quad \frac{1}{2} \quad \frac{1}{2} \quad \frac{3}{4} \quad 1 \quad 1 \quad 1 \right].$$

The interpolating processes are as follows:

- the sampling time in interpolation is $T_s = 0.002$ sec; and
- the feedrate command is given as $F = 200$ mm/sec = 12 m/min.

Fig. 5 shows results in terms of chord errors for two algorithms, the uniform interpolation and the first-order approximation interpolation. Speed fluctuation is significantly reduced by applying the first-order approximation algorithm, as shown in Fig. 6. For the proposed chord error-controlled interpolation algorithm, the tolerance of the chord error is set at 0.001 mm and the speed is simultaneously limited to under 12 m/min. Results are shown in Figs. 7 and 8. As summarized in Table 1, the present interpolation algorithm achieves the greatest interpolation accuracy. Note that the computation time for the present algorithm is the highest but still acceptable for interpolation. Discussions of the simulation results are as follows:

1. For computer graphics or CAD systems, speed is not crucial in representing the parametric curves. Results of the uniform-interval NURBS in Table 1 and Fig. 6 indicate that the chord error of the uniform-interval NURBS is acceptable. For example, with an average speed of around 50 m/min, the maximum chord error is only 2.5 as in Table 1. However, the speed of the uniform-interval NURBS varies dramatically, from 1/10 to 10 times the average speed, and thus machining quality may be seriously degraded. Also, in this interpolation, the interval corresponding to the specified feedrate is undetermined. Therefore, the uniform-interval NURBS is not suitable for machining processes because of its unstable feedrate.
2. Applying the speed-controlled NURBS significantly improves the speed accuracy, and results indicate that the speed variation is within 5%. As for the chord error, it is satisfactory at low speed. For example, with a feedrate of 6 m/min, the chord error of the speed-controlled NURBS is around 1μ. However, as the feedrate increases to 20 m/min, its chord error greatly increases to 10μ. Therefore, for a precise CNC system with an error budget 10μ which includes both interpolation errors and motion control errors, the speed-controlled NURBS interpolation would not be suitable for high-speed operation because of its excessive chord error.
3. For the present adaptive-feedrate NURBS, results in Fig. 8 show that the feedrate is automatically reduced at sharp corners while the chord error is confined well within the specified 1μ. It is known that higher machining quality is achieved if the feedrate can be maintained within a

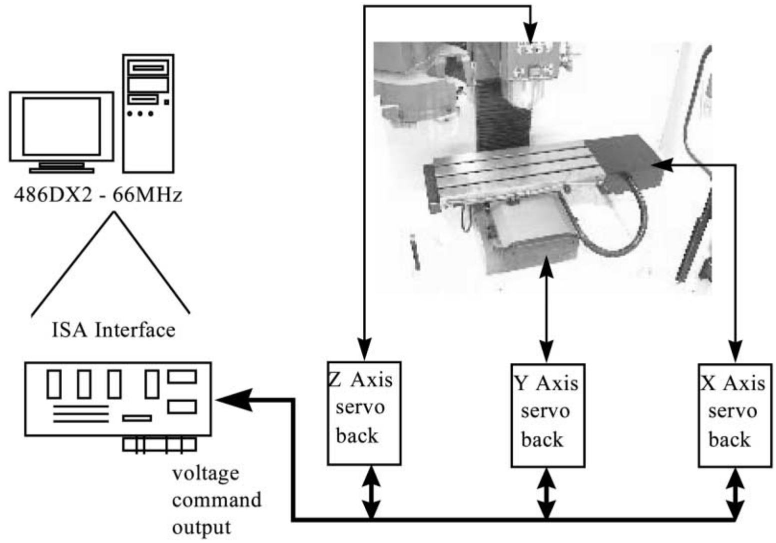

Fig. 9. The experimental setup.

specified range with a constant speed. Also, the dimension precision can be obtained with the minimized chord error. With the proposed adaptive-feedrate NURBS interpolation, although the machining time increases because of the reduced feedrate, better machining quality and precision are guaranteed even under high-speed operations, as shown in Figs. 7 and 8 and Table 1.

5. Experimental results

The experimental setup of a three-axis DYNA 1007 CNC machine center is shown in Fig. 9. A PC-486 generated the main control commands and recorded the signals, including: the input commands for different contours, the implementation of a variable-gain CCC, and the control inputs to the velocity loop of the AC servo motors. The PC-486 interface utilized an AD/DA card to send and receive the control input and position output at a sampling period of 1 ms. A NURBS path as in Fig. 4, with 1/3 scale and a feedrate $F = 4.5$ m/min were adopted. The sampling time in interpolation was $T_s = 0.001$ sec. To verify the adaptive-feedrate function of the proposed algorithm, the chord error was confined to within 0.001μ during the interpolation process, as shown in Fig. 10. The results shown in Fig. 11 indicate that the proposed interpolation algorithm leads to less contouring error than the speed-controlled NURBS interpolation, especially at the corner of the path magnified in

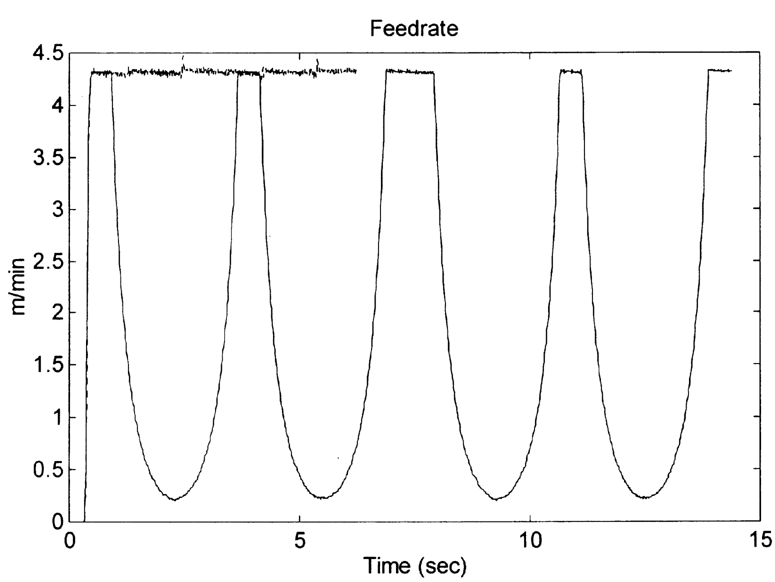

Fig. 10. Measured speed for the speed-controlled NURBS (dashed) and the adaptive-feedrate NURBS (solid).

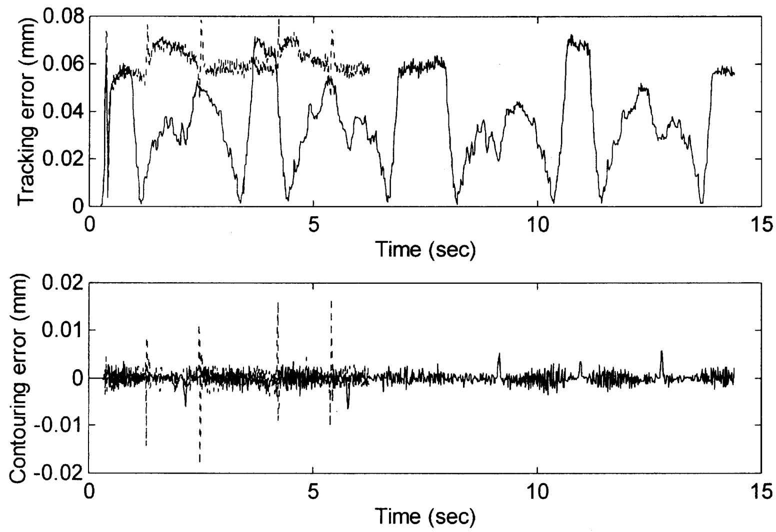

Fig. 11. Experimental results for the speed-controlled NURBS (dashed) and the adaptive-feedrate NURBS (solid).

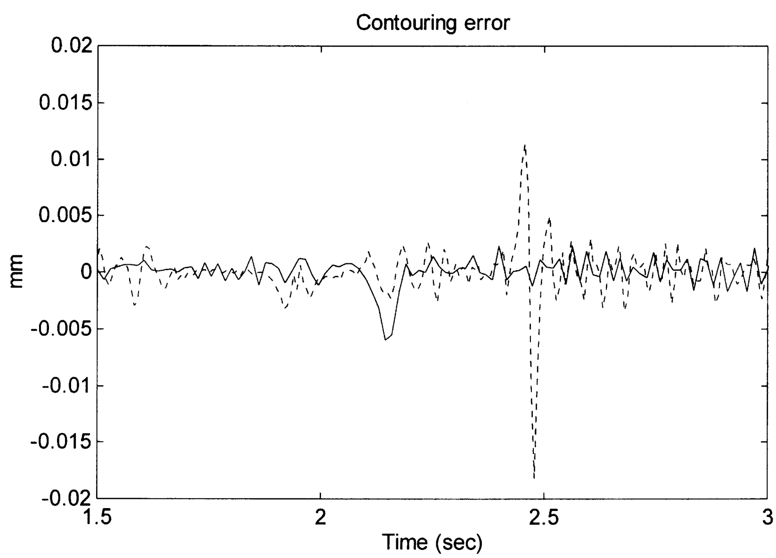

Fig. 12. Detailed results at the corner for the speed-controlled NURBS (dashed) and the adaptive-feedrate NURBS (solid).

Fig. 12. As summarized in Table 2, the contouring error of the proposed interpolation is one third that of the speed-controlled interpolation.

6. Conclusions

For parametric curves, interpolation accuracy in terms of position is mainly determined by the radial error and the chord error. With modern computing processors, the radial error is usually negligible. Therefore, the chord error is the main concern in general interpolation algorithms. The proposed interpolation achieves greater machining precision by using adaptive feedrate to reduce chord errors. Simulation results show that the proposed interpolation algorithm effectively improves interpolation accuracy in

Table 2
Experimental results of errors for different NURBS algorithms

Measurement	Tracking error (mm)		Contouring error (mm)	
Interpolation	MAX	RMS	MAX	RMS
Speed-controlled NURBS	0.080	0.059	0.017	0.002
Adaptive-feedrate NURBS	0.073	0.040	0.006	0.001

terms of the chord error and maintains speed accuracy at a specified level. Implementation on a CNC machining center also has proven the feasibility and the improved contouring performance of the proposed adaptive-feedrate interpolation algorithm.

Acknowledgements

This work was supported by the Taiwan Industrial Technology Research Institute under contract no. 873K77BE4, and by the National Science Council, R.O.C., under contract no. NSC 88-2212-E-009-026.

References

[1] Shpitalni M, Koren Y, Lo CC. Realtime curve interpolators. Computer-Aided Design 1994;26(11):832–8.
[2] Bedi S, Ali I, Quan N. Advanced techniques for CNC machines. Trans. of the ASME J. of Engineering for Industry 1993;115:329–36.
[3] Koren Y. Computer control of manufacturing systems. USA: McGraw-Hill, 1982.
[4] Huang JT, Yang DCH. A generalized interpolator for command generation of parametric curves in computer controlled machines. ASME Japan/USA Symposium on Flexible Automation 1992;1:393–9.
[5] Yang DCH, Kong T. Parametric interpolator versus linear interpolator for precision CNC machining. Computer-Aided Design 1994;26(3): 225–33.
[6] Yeh SS, Hsu PL. The speed-controlled interpolator for machining parametric curves. Computer-Aided Design 1999;31(5):349–57.
[7] Piegl L, Tiller W. The NURBS book. New York: Springer, 1997.

Syh-Shiuh Yeh received the BS degree in mechanical engineering, the MS and PhD degrees in electrical and control engineering from National Chiao Tung University, Taiwan, R.O.C., in 1994, 1996, and 2000, respectively. He is currently a researcher at the Mechanical Industry Research Laboratory, Industrial Technology Research Institute, R.O.C. His research interests include motion system and control design.

Pau-Lo Hsu (M'91) received the BS degree from National Kung University, Taiwan, the MS degree from the University of Delaware, and the PhD degree from the University of Wisconsin-Madison, in 1978, 1984, and 1987, respectively. Following two years of military service in King-Men, he was with San-Yang (Honda) Industry during 1980–1981 and Sandvik (Taiwan) during 1981–1982. In 1988, he joined the Department of Electrical and Control Engineering, National Cheng Kung University, Hsinchu, Taiwan, R.O.C. as Associate Professor. He became a professor in 1995. During 1998–2000, he served as the Chairman of the department. His research interests include mechatronics, CNC motion control, servo systems, and diagnostic systems.