

Look-ahead Pythagorean-hodograph Interpolation for High Speed CNC Machining

Behnam Moetakef Imani¹, Amirmohammad Ghandehariun²

¹Associate Professor, Ferdowsi University of Mashhad; imani@um.ac.ir

²M.Sc. Student, Ferdowsi University of Mashhad; am.gandehariun@stu-mail.um.ac.ir

Abstract

Various Methods for parametric interpolation for NURBS curves have been proposed in the past. However, the errors caused by the approximate nature of the NURBS interpolator were rarely taken into account. This paper proposes an integrated look-ahead algorithm for parametric interpolation along NURBS curves. The algorithm interpolates the sharp corners on the curve with the exact Pythagorean-hodograph (PH) interpolation. This will minimize the geometric and interpolator approximation errors simultaneously. The algorithm consists of three different modules: a sharp corner detection module, a jerk-limited module, and a dynamics module. Simulations are performed to show correctness of the proposed algorithm. It is shown that the developed method improves tracking and contour accuracies significantly compared to previously proposed adaptive-feedrate and curvature-feedrate algorithms.

Keywords: Numerical control, high speed machining, PH curve

Introduction

The function of the real-time interpolator in a computer numerical control (CNC) machine is to convert prescribed tool path and feedrate data into reference points for each sampling interval of the servo system. The closed-loop position and speed control can be achieved through comparing the actual machine position, measured by encoders on motor axes, with the reference point [1].

Parametric curves are extensively being used in a wide range of industries such as automotive, aerospace and dies/molds. Different representations are available for parametric curves, mainly including Bezier, B-spline, cubic spline and non-uniform rational B-spline (NURBS). Among these representations, NURBS have gained wide popularity to gradually become the industry standard [2]. Modern CNC machines not only provide linear/circular interpolations, but also offer parametric interpolations. Some researchers have shown that utilizing parametric interpolations will result in reduced feedrate fluctuations and chord errors and shorter machining time in comparison with linear/circular interpolations [3].

Accurate feedrate performance issues become increasingly important in the context of high-speed machining, where extreme feed acceleration /deceleration rates are required. In addition, tool chatter or breakage is a plausible result of the interpolator's

inability to properly maintain the commanded feedrates [4]. Interpolators for general B-spline/NURBS curves are typically obtained based on Taylor series expansions. Their main task is to compute parameter values of successive reference points. Such schemes inevitably incur truncation errors, caused by omission of higher-order terms [4].

To overcome this problem, the tool path can be described in terms of the Pythagorean-hodograph (PH) curves [5]. The algebraic structure of PH curves admits a closed-form reduction of the interpolation integral, yielding real-time CNC interpolator equations for constant or variable feedrates with high accuracy, robustness and flexibility [6].

Many interpolation methods for parametric curves have been proposed. Yeh et al. [7] developed a parametric curve interpolator using a Taylor series expansion algorithm. Farouki et al. [4] derived the exact Taylor series coefficients for variable-feedrate interpolators up to the cubic terms. Tsai et al. [6] proposed an algorithm for parametric interpolation with time-dependent feedrates along PH curves. Yeh et al. [8] proposed an adaptive-feedrate interpolation algorithm. The feedrate was adjusted based on confined chord errors. Zhiming et al. [9] suggested a strategy of variable feedrate machining based on geometric properties of tool path. Tsai et al. [3] proposed an integrated look-ahead dynamics-based algorithm for interpolation along NURBS curves which considered geometric and servo errors simultaneously.

In this paper, an integrated look-ahead Pythagorean hodograph interpolation algorithm for NURBS curves is proposed. The algorithm consists of a sharp corner detection module, a jerk-limited module, and a dynamics module to take into account chord errors, feedrate fluctuations, servo errors, and interpolation errors on sharp corners simultaneously. With the proposed algorithm, planning a higher feedrate profile while achieving better contour accuracy and shorter machining time is possible. Simulations are performed to show effectiveness of the proposed algorithm.

NURBS Interpolator Algorithm

Suppose $C(u)$ represents a NURBS curve and is given by [10]:

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

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

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

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

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

To implement NURBS interpolation, a second-order approximation interpolation algorithm is adopted here. Using the Taylor series expansion method, the curve approximation up to the second derivative is given as follows:

$$u_{k+1} = u_k + \left. \frac{du}{dt} \right|_{t=t_k} \Delta t + \left. \frac{d^2u}{dt^2} \right|_{t=t_k} \frac{(\Delta t)^2}{2!} \quad (4)$$

Substituting the derivatives into the Eq. (4), the second-order Taylor expansion can be expressed as [4]:

$$u_{k+1} = u_k + \frac{V_k T_s}{|C'(u_k)|} + \frac{T_s^2}{2} \left\{ \frac{A_k}{|C'(u_k)|} - \frac{V_k^2 [C'(u_k) \cdot C''(u_k)]}{|C'(u_k)|^4} \right\} \quad (5)$$

where V_k , A_k , T_s , $C'(u_k)$ and $C''(u_k)$ are the feedrate, the acceleration, the sampling time, and the first and second derivatives of the NURBS curve, respectively. The De Boor algorithm can significantly improve the computational performance for calculating $C(u_k)$, $C'(u_k)$ and $C''(u_k)$ compared to the recursive basis function [3], thus it has been used for the real-time implementation.

Pythagorean-Hodograph Interpolator Algorithm

A polynomial PH curve $r(\xi) = (x(\xi), y(\xi))$ is defined by its parametric hodograph [6]:

$$x'(\xi) = u^2(\xi) - v^2(\xi) \quad y'(\xi) = 2u(\xi)v(\xi) \quad (6)$$

These forms guarantee that $x'(\xi)$ and $y'(\xi)$ are elements of a Pythagorean triple of polynomials—they satisfy

$$x'^2(\xi) + y'^2(\xi) = \sigma^2(\xi) \quad (7)$$

where

$$\sigma(\xi) = |r'(\xi)| = u^2(\xi) + v^2(\xi) = \frac{ds}{d\xi} \quad (8)$$

is the parametric speed of $r(\xi)$. A PH quintic is obtained by substituting Bernstein-form quadratic polynomials

$$\begin{aligned} u(\xi) &= u_0(1-\xi)^2 + 2u_1(1-\xi)\xi + u_2\xi^2 \\ v(\xi) &= v_0(1-\xi)^2 + 2v_1(1-\xi)\xi + v_2\xi^2 \end{aligned} \quad (9)$$

into Eq. (6) and integrating [6].

Considering a time-dependent feedrate function $V(t)$ with the indefinite integral $F(t)$ imposed on the

PH curve $r(\xi)$, the interpolation equation yields to the solution of the relation

$$s(\xi) = F(t) \quad (10)$$

where $s(\xi)$ is the arc length [6]. For a sampling interval Δt , the real-time CNC interpolator must compute the parameter values ξ_1, ξ_2, \dots of reference points at times $\Delta t, 2\Delta t, \dots$. These values are roots of the polynomial equations

$$s(\xi_k) = F(k\Delta t) \quad k = 1, 2, \dots \quad (11)$$

The above equations can be solved to obtain ξ_k using a few Newton-Raphson iterations [6],

$$\xi_k^{(r+1)} = \xi_k^{(r)} - \frac{s(\xi_k^{(r)}) - F(k\Delta t)}{\sigma(\xi_k^{(r)})}; r = 0, 1, \dots \quad (12)$$

with starting approximation $\xi_k^{(0)} = \xi_{k-1}$.

The Look-Ahead PH-based Algorithm

1. System architecture

The look-ahead PH-based algorithm, which acts as a CNC controller is described in detail in this section. The controller consists of three main programs: a CNC interpreter, a reference point generator, and a motion controller. The CNC interpreter reads NC commands from NC files to generate and store NC blocks in the NC FIFO (First-In-First-Out) memory. The reference point generator generates consecutive reference points based on the proposed algorithm. The motion controller will then control the X-Y table based on the generated reference points.

The reference point generator algorithm consists of three different modules: a sharp corner detection module, a jerk limited feedrate planning module and a dynamics module. The sharp corner detection module looks ahead NC blocks to identify sharp corners of the curve. Then the curve is divided into small segments according to sharp corners. The second task of this module is to construct PH curves. Here, the region on the curve that is found to be a sharp corner will be approximated using a PH curve. The length of each segment is calculated and stored in the buffer. The jerk-limited module plans the feedrate profile of each segment based on constraints on chord errors, feedrate, acceleration/deceleration, and jerk. The dynamics module will further modify the feedrate profile to keep the contour errors bounded to their maximum commanded value. Finally, the motion controller performs real-time control on the X-Y table. Algorithms for each module are detailed in the following sections.

2. The sharp corner detection module

2.1. Identifying sharp corners

The sharp corner detection module plays an important role in the look-ahead algorithm. In this study, a sharp corner is defined as the feedrate sensitive zone at which the feedrate should be reduced to maintain contour accuracy. There are two criteria in identifying sharp corners. The first criterion is that the derivative

of the curvature at a sharp corner is equal to zero; it is given as:

$$\left. \frac{d\kappa(u)}{du} \right|_{u=u_k} = 0 \quad (13)$$

where $\kappa(u)$ is the curvature given as:

$$\kappa(u) = \frac{\left| \frac{dC}{du} \times \frac{d^2C}{du^2} \right|}{\left| \frac{dC}{du} \right|^3} \quad (14)$$

The first criterion is not sufficient in determining sharp corners as it does not include feedrate effects [3]. Therefore, in order to consider feedrate effects in identifying sharp corners, a second criterion is utilized. Based on this criterion, the curvature at the sharp corner zone should exceed the threshold value κ_{th} , which is defined as:

$$\kappa_{th} = \frac{A_{\max}}{V_{\max}^2} \quad (15)$$

where A_{\max} is the maximum acceleration limit and V_{\max} is the given feedrate in the NC block. In other words, the second criterion examines whether the centripetal accelerations of local max/min points exceed the maximum acceleration limit [3]. The region on the curve where this value is exceeded is identified as the sharp corner zone, and it should be replaced with a PH curve for a better supervision of the feedrate. These criterions can be illustrated by the curve shown in Figure 1 where segments AB and CD are identified as sharp corners.

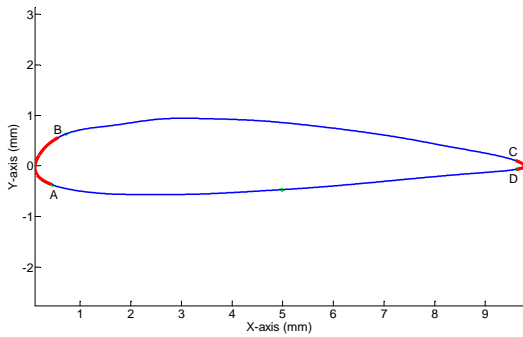


Figure 1: NURBS approximation of a NACA2415 airfoil profile

2.2. Constructing PH curve

Utilizing Taylor series in interpolation along the NURBS curve will inherently cause interpolation errors and feedrate fluctuations, which will reduce the contour accuracy and surface quality. However, PH curves admit analytic reduction of the interpolation integral, which will result in an exact interpolation equation. Therefore, approximating the sharp corner regions on the NURBS curve with PH curves will decrease feedrate fluctuations.

In order to approximate an area on the NURBS curve by a PH quintic, Hermite data of the segment, namely the start and end points and derivatives, should

be computed. Using the complex representation for a planar PH curve [11],

$$r(\xi) = x(\xi) + iy(\xi) \quad (16)$$

where ξ is a real parameter, the complex polynomial $w(\xi) = u(\xi) + iv(\xi)$ will be defined so that

$$r'(\xi) = w^2(\xi) \quad (17)$$

Considering r_0 , r_1 , d_0 and d_1 as complex representations of the start point, end point, start derivative and end derivative, respectively, $w(\xi)$ can then be expressed in Bernstein form [11],

$$w(\xi) = w_0(1-\xi)^2 + 2w_1(1-\xi)\xi + w_2\xi^2 \quad (18)$$

where

$$w_0 = \pm\sqrt{d_0} \quad (19)$$

$$w_2 = \pm\sqrt{d_1} \quad (20)$$

$$w_1 = \frac{1}{4} \left[-3(w_0 + w_2) \pm \sqrt{120(r_1 - r_0) - 15w_0^2 - 15w_2^2 + 10w_0w_2} \right] \quad (21)$$

The PH curve on the sharp corner should be constructed in the way that the error between the curve and the original NURBS curve is bounded. In order to achieve this, the module will calculate the root-mean-square of error between the two curves. If the calculated error exceeds the assigned maximum error, the sharp corner zone on the original curve will be divided into two segments and a PH curve will be constructed on each segment. This procedure will be repeated until the desired maximum error is achieved. This algorithm is utilized for the airfoil in Figure 1 with the maximum allowed error $e_{\max} = 1\mu m$. Table 1 shows the effect of dividing the sharp corner segments on the error between the original curve and the approximated PH curve.

Table 1: Comparison of error of PH approximation for the airfoil with $e_{\max} = 1\mu m$

Curve segment	Root-mean-square of error (μm)	
	Without curve division	With curve division
AB	4.9094	0.5205
CD	4.3893	0.5562

3. The jerk-limited module

The first task of the jerk-limited module is to obtain the feedrate at sharp corners. To achieve this task, the proposed algorithm combines the adaptive-feedrate interpolation scheme [8] with the curvature-based feedrate interpolation algorithm [9]. The algorithm for determining the feedrate at sharp corners is given as:

$$V_{sp}(u_k) = \min\{V_{af}(u_k), V_{cf}(u_k)\} \quad (22)$$

where $V_{af}(u_k)$ and $V_{cf}(u_k)$ are given as follows:

$$V_{af}(u_k) = \frac{2}{T_s} \sqrt{\frac{2\delta}{\kappa(u_k)} - \delta^2} \quad (23)$$

$$V_{cf}(u_k) = \frac{\kappa^*}{\kappa(u_k) + \kappa^*} V_{\max} \quad (24)$$

Here, V_{af} , V_{cf} , V_{\max} are adaptive-feedrate, curvature-feedrate, and feedrate commands, respectively. δ and κ are the chord tolerance and the curvature of a NURBS curve, respectively. $\bar{\kappa}$ maintains the derivative continuity of curvature-feedrate in Eq. (24).

Having obtained the feedrate at sharp corners, the second task of the jerk-limited module is to plan the feedrate profile of each segment such that the constraints on feedrate, acceleration/deceleration, and jerk are satisfied. The bell-shape feedrate profile [12] will be generated based on the calculated length of each segment, feedrates at the corners, maximum feedrate, maximum acceleration/deceleration and the jerk limit. By applying the sharp corner detection and jerk-limited modules, the feedrate profile for the airfoil is as shown in Figure 2, where maximum acceleration/deceleration and jerk limit are set to 2450mm/s² and 50 000 mm/s³.

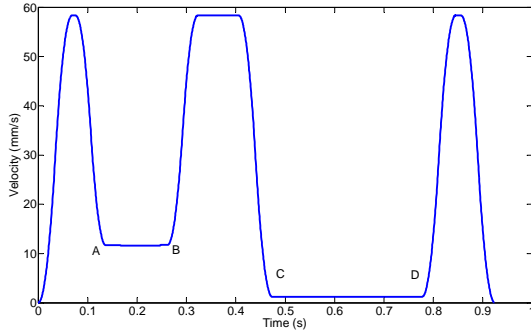


Figure 2: The feedrate profile of the airfoil after application of sharp corner and jerk-limited modules

4. The dynamics module

Application of the first and second modules will not guarantee the satisfaction of the constraint on the contour error. This is because no contour error information is included in these two modules.

In order to predict the contour errors, the AC servo control system for each axis should be modeled and simulated. The block diagram shown in Figure 3 is used as the dynamic model of the servo control system.

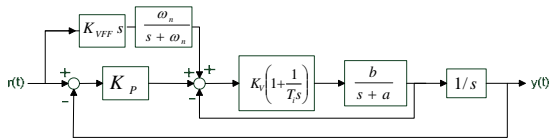


Figure 3: Block diagram of a servo control system [3]

The closed loop transfer function of the servo control system for each axis shown in Figure 3 is given by:

$$G(s) = \frac{Y(s)}{R(s)} = \frac{b_2 s^2 + b_1 s + b_0}{a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \quad (25)$$

where the parameters in Eq. (25) is given in Table 2.

Table 2: Parameters of servo control transfer function [3]

Parameter	X-axis	Y-axis
a ₀	1.938×10 ⁹	1.904×10 ⁹
a ₁	3.538×10 ⁷	3.496×10 ⁷
a ₂	2.135×10 ⁵	2.120×10 ⁵
a ₃	6.984×10 ²	6.948×10 ²
a ₄	1.00	1.00
b ₀	1.938×10 ⁹	1.904×10 ⁹
b ₁	3.476×10 ⁷	3.435×10 ⁷
b ₂	1.471×10 ⁵	1.466×10 ⁵

Based on Eq. (25), the transfer function between the tracking error $E(s)$ and the velocity $V(s)$ on each axis is obtained as:

$$\frac{E(s)}{V(s)} = \frac{a_4 s^4 + a_3 s^3 + (a_2 - b_2) s^2 + (a_1 - b_1) s + (a_0 - b_0)}{a_4 s^5 + a_3 s^4 + a_2 s^3 + a_1 s^2 + a_0 s} \quad (26)$$

Having developed the tracking error equation, the contour error equation can be approximated as [3]:

$$\varepsilon_e = -E_x \sin \varphi + E_y \cos \varphi \quad (27)$$

where E_x and E_y are tracking errors on the x-axis and y-axis, respectively. φ is the angle between the tool path tangent and the x-axis.

The dynamics module will use Eq. (26) and Eq. (27) to calculate the contour error. When the contour error ε_e is greater than the set maximum contour error ε_{\max} , the computation is stopped and the feedrate $V(u_k)$ at the time is recorded. Then the feedrate profile will be planned again with the maximum feedrate set to $V(u_k)$. This procedure will be repeated until the desired constraint on the contour error is satisfied. Figure 4 shows the feedrate profile of the airfoil curve using the look-ahead PH-based algorithm. The maximum feedrate on the BC segment is reduced as compared to Figure 2.

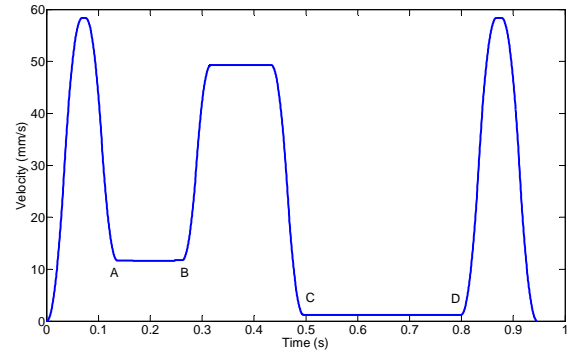


Figure 4: Feedrate profile of the airfoil using the look-ahead PH-based algorithm

Results and Discussions

In this section, numerical simulations are performed on the airfoil curve in three different angles of 0°, -45° and 45° as shown in Figure 5. Parameters of the interpolator for numerical simulations are as listed in Table 3.

Simulations are conducted to compare the performance among adaptive-feedrate [8], curvature-feedrate [9], and the look-ahead PH-based interpolation

algorithms. The servo control system as shown in Figure 3 is included in the simulation. Performances of the three interpolation algorithms are listed in Table 4.

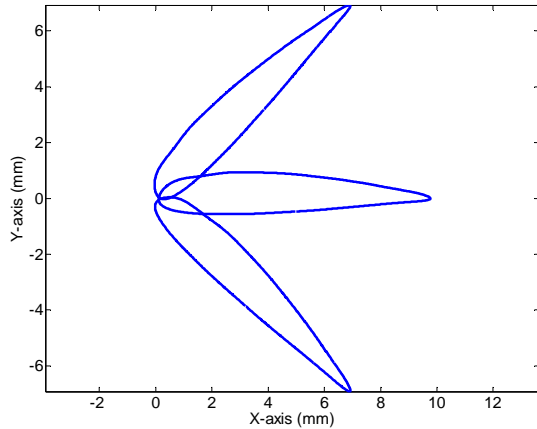


Figure 5: The airfoil curve at three different angles for numerical simulations

Table 3: Parameters of the interpolator for numerical simulations

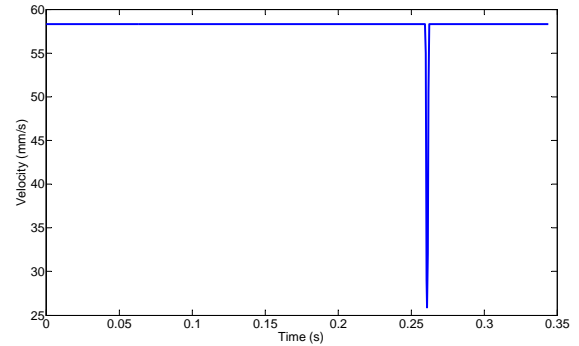
Maximum feedrate	V_{\max}	3500mm/min
Maximum acceleration	A_{\max}	2450mm/s ²
Maximum jerk	J_{\max}	5×10^4 mm/s ³
Chord error limitation	δ	1 μ m
Reference curvature	$\bar{\kappa}$	1mm ⁻¹
Contour error limit	ε_{\max}	15 μ m
PH approximation error limit	e_{\max}	1 μ m

Table 4: Performance comparison of different interpolation algorithms

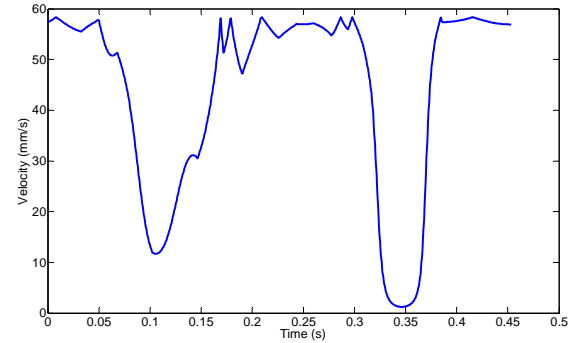
Interpolation algorithm		Adaptive feedrate	Curvature-feedrate	Look-ahead PH-based	
Tracking error (μ m)	X-axis	Max	358.69	195.78	57.97
		RMS	75.65	41.53	20.30
	Y-axis	Max	132.56	22.65	13.98
		RMS	27.31	8.39	3.56
Contour error (μ m)	Max	142.79	47.02	14.11	
	RMS	35.14	12.98	3.84	
	Mean	16.780	8.50	2.31	
Time (s)		0.3440	0.4535	0.9455	

Figure 6 compares the feedrate profiles of adaptive-feedrate, curvature-feedrate and the look-ahead PH-based algorithms. It can be noted that the proposed look-ahead PH-based algorithm with its look-ahead capability obtains smooth feedrate profile compared to the other two approaches.

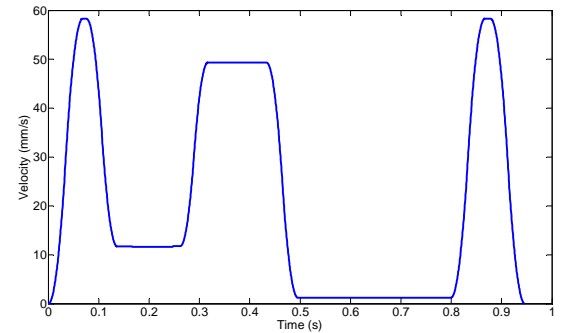
Figure 7 shows the contour errors of the above mentioned algorithms. The contour errors of the look-ahead PH-based interpolation are significantly smaller than those of the adaptive-feedrate and curvature-feedrate interpolations. The proposed algorithm modifies the feedrate of the segment BC to keep the contour errors within their maximum limitation of 15 μ m.



(a)



(b)



(c)

Figure 6: Comparison of feedrate profiles: (a) Adaptive-feedrate; (b) Curvature feedrate; (c) Look-ahead PH-based

The results of the simulation of the tracking and contour errors on the airfoil curve at three different angles are reported in Table 5. It can be noted that the tracking errors in machining the airfoil profile at three angles are different while the contour errors remain almost constant.

Table 5: Numerical simulation of the errors on the airfoil curve at three different angles

Airfoil curve angle		45°	-45°	0°	
Tracking error (μ m)	X-axis	Max	38.71	43.51	57.97
		RMS	14.19	14.93	20.30
	Y-axis	Max	43.66	38.85	13.98
		RMS	14.99	14.24	3.56
Contour error (μ m)	Max	14.12	14.12	14.11	
	RMS	3.84	3.84	3.84	
	Mean	2.31	2.31	2.31	

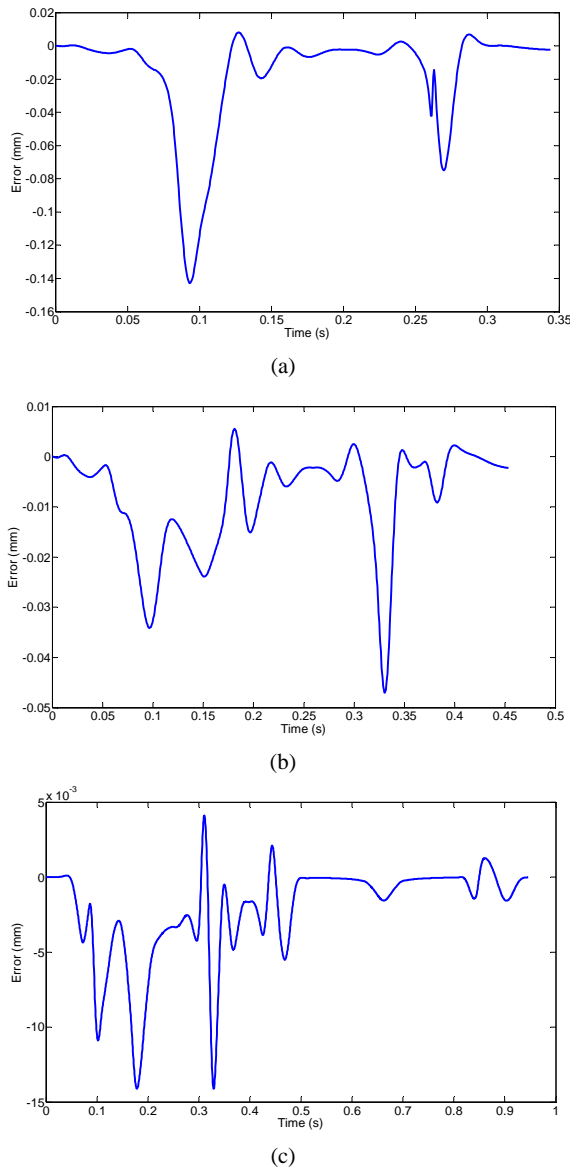


Figure 7: Comparison of contour error: (a) Adaptive-feedrate; (b) Curvature feedrate; (c) Look-ahead PH-based

Conclusion

A novel NURBS interpolator was proposed in this paper. Sharp corners of the curve are identified not only by the derivative of the curvature but also by the feedrate criterion. A PH curve is then constructed on the corner to reduce feedrate fluctuations. The error between the constructed curve and the original NURBS curve is bounded to its set maximum value. The curve is divided into small segments and a smooth jerk-limited profile is planned on each segment. The dynamics module further modifies the feedrate profile to keep the contour errors limited. Simulations were performed to validate the proposed algorithm. It is

shown that the look-ahead PH-based algorithm can increase contour accuracy significantly compared with adaptive-feedrate and curvature-feedrate interpolation algorithms. This research demonstrates the effectiveness of the proposed algorithm.

References

- [1] Farouki, R. T., 2008. *Pythagorean-Hodograph Curves: Algebra and Geometry Inseparable*, Springer.
- [2] Liu, X., Ahmad, F., Yamazaki, K., and Mori, M., 2005. "Adaptive interpolation scheme for NURBS curves with the integration of machining dynamics". *International Journal of Machine Tools and Manufacture*, 45, pp. 433-444.
- [3] Tsai, M. S., Nien, H. W., and Yau, H. T., 2008. "Development of an integrated look-ahead dynamics-based NURBS interpolator for high precision machinery". *Computer-Aided Design*, 40, pp. 554-566.
- [4] Farouki, R. T., and Tsai, Y. F., 2001. "Exact Taylor series coefficients for variable-feedrate CNC curve interpolators". *Computer-Aided Design*, 33, pp. 155-165.
- [5] Farouki, R. T., and Sakkalis, T., 1990. "Pythagorean hodographs". *IBM Journal of Research and Development*, 34, pp. 736-752.
- [6] Tsai, Y. F., Farouki, R. T., and Feldman, B., 2001. "Performance analysis of CNC interpolators for time-dependent feedrates along PH curves". *Computer Aided Geometric Design*, 18, pp. 245-265.
- [7] Yeh, S. S., and Hsu, P. L., 1999. "The speed-controlled interpolator for machining parametric curves". *Computer-Aided Design*, 31, pp. 349-357.
- [8] Yeh, S. S., and Hsu, P. L., 2002. "Adaptive-feedrate interpolation for parametric curves with a confined chord error". *Computer-Aided Design*, 34, pp. 229-237.
- [9] Zhiming, X., Jincheng, C., and Zhengjin, F., 2002. "Performance evaluation of a real-time interpolation algorithm for NURBS curves". *International Journal of Advanced Manufacturing Technology*, 20, pp. 270-276.
- [10] Piegl, L., and Tiller, W., 1997. *The NURBS Book*. 2nd Ed., Springer, New York.
- [11] Moon, H. P., Farouki, R. T., and Choi, H. I., 2001. "Construction and shape analysis of PH quintic Hermite interpolants". *Computer Aided Geometric Design*, 18, pp. 93-115.
- [12] Cheng, C. W., and Tsai, M. C., 2004. "Real-time variable feed rate NURBS curve interpolator for CNC machining". *International Journal of Advanced Manufacturing Technology*, 23, pp. 865-873.