2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics
Suntec Convention and Exhibition Center
Singapore, July 14-17, 2009

# An Adaptive Real-time NURBS Interpolator for CNC Machine Tools

H. Hu and X.T. Zhang,

*Department of Mechanical Engineering and Automation*
*Harbin Institute of Technology Shenzhen Graduate School*
*Shenzhen, China*
*E-mail:honghu@hit.edu.cn*

***Abstract*—This paper presents a real-time interpolation method along non-uniform rational B-spline (NURBS) curves which have been used by CNC systems. The development of NURBS interpolator promises modern manufacturing system to machine arbitrary geometries with great relief of the data-flow bottleneck. In this interpolation algorithm, an Adams-Bashforth method is used as a predictor to predict the next trajectory point on the NURBS curve and an adaptive corrector is proposed so that a specified deviation between the command feedrate and the desired feedrate is met. The convergent condition for the algorithm is presented and demonstrated. Both simulation and experimental results for the NURBS interpolation are employed to verify the feasibility of the algorithm and comparison with first and second order of Taylor's expansion method is provided to reveal the accuracy of this proposed algorithm.**

***Keywords*—adaptive interpolator, NURBS, real-time control**

## I. INTRODUCTION

In conventional CNC systems, the desire free-form curves and surfaces are generally achieved by the approximation of a series of straight line or circular segments jointed end-by-end. However, the approximation will generate a huge number of data and increase the load of the CNC systems in order to meet the machining accuracy. It will cause the motion speed and acceleration unsmooth due to the discontinuity of the segments. As a result, there is an increasing need for developing a novel interpolator to overcome these disadvantages. A wide variety of parametric curves have been proposed to be used for the interpolation [1] [2]. As the most representative one, the non-uniform rational B-splines, commonly refers to as NURBS, has become the industry standard for the representation, design, and data exchange of geometric information processed by computers. Many national and international standards, e.g., IGES, STEP and PHIGS, recognize NURBS as powerful tools for geometric design [3]. Its particular mathematical properties allow a complex contour to be presented using only a few parameters, by which CNC systems are greatly benefited. A higher accuracy than the conventional interpolation can be obtained since it can be executed directly without any segmentation. Moreover, the amounts of program blocks can be greatly reduced while maintain a high accuracy instead of choosing an appropriate tradeoff between the accuracy and data volume.

Several investigations have been made to develop real-time interpolators for the parametric curve. Bedi et al. [4] developed a uniform interpolation algorithm in which the independent variable was constantly increasing. Although the algorithm was simple and its segment length in each period was unequal, it could not provide constant feedrate for machining process. Shpitalni et al. [5] gave a parametric curve interpolator with its segmentation based on curve segments of equal length rather than equal independent variable increment. In order to calculate the next trajectory point, Shpitalni et al. used Taylor's first order expansion method which guaranteed a constant feedrate. Subsequently, Yang and Kong [1] and Yeh and Hsu [9] got more accurate results by extending Taylor series coefficients into the second order. These algorithms based on Taylor's expansion method developed real-time interpolators for general curves; however, they did not accurately achieve the desired goal since their systems were open-looped. In order to overcome this drawback, Tsai et al. [7] presented a predictor-corrector interpolator for parametric curves in which Tsai used the backward difference approximation method to predict the next trajectory point, and then corrected it with the current feedrate command feedback compensation scheme by which the deviations between the current and desired feedrate commands would always fall within the specified tolerance. The method could reach more accuracy, and also brought unexpected fluctuation to the feedrate.

This study presents a self-adjustment interpolator for parametric curves and allows us to calculate the next trajectory point rapidly and accurately. This proposed method uses Adams-Bashforth method to initially estimate the next trajectory point which subsequently determines the current feedrate. If the ratio of current feedrate and desired feedrate overrides a pre-specified feedrate tolerance, the corrector will be in action to adjust the next trajectory point automatically according to the current feedrate feedback until an acceptable ratio error is met.

Supported by Research Funding of Shenzhen Government

## II. REAL-TIME INTERPOLATOR FOR NURBS

### A. Definition of NURBS curves

In 3-dimensions space, all points on a curve in parametric form are represented by 3 coordinates separately as an explicit function

$$C(u)=(x(u),y(u),z(u)) \qquad a\le u\le b$$

where $C(u)$ is a vector-valued function of the independent variable, $u$ is an arbitrary independent parameter usually normalized to [0,1].

A p-th degree NURBS curve is defined by [3]

$$C(u)=\frac{\sum_{i=0}^{n}N_{i,p}(u)w_iP_i}{\sum_{i=0}^{n}N_{i,p}(u)w_i} \qquad a\le u\le b \tag{1}$$

where $\{P_i\}$ are control points, $\{w_i\}$are weights, and $\{N_{i,p}(u)\}$ are the p-th degree B-spline basis functions defined on non-periodic (and non-uniform) knot vector

$$U=\{\underbrace{a,\ldots,a}_{p+1},u_{p+1},\ldots,u_{m-p-1},\underbrace{b,\ldots,b}_{p+1}\}$$

and $N_{i,p}(u)$ can be defined by the recursive formula

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

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

The m-th derivative of the NURBS with respect to $u$ can be obtained by taking the derivative of Eq. (1)

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

where the m-th derivative of basis function can be produced as

$$N_{i,p}^{(m)}(u)=(p-1)\left[\frac{N_{i,p-1}^{(m-1)}(u)}{u_{i+p-1}-u_i}-\frac{N_{i+1,p-1}^{(m-1)}(u)}{u_{i+p}-u_{i+1}}\right]$$

### B. Conventional interpolator

The essence of a real-time interpolator is that the tool path machined in each sampling period $T$ should be based on the equal increment of the segment length rather than the equal increment of the independent parameter $\Delta u$ . The federate $V(t)$ along the NURBS curve is defined by

$$V(t)=\frac{ds}{dt}=\left(\frac{ds}{du}\right)\left(\frac{du}{dt}\right) \text{ or } \frac{du}{dt}=\frac{V(t)}{ds/du} \tag{2}$$

where

$$\frac{ds}{du}=\sqrt{(x')^2+(y')^2+(z')^2}$$
$$x'=\frac{dC_x(u)}{du},y'=\frac{dC_y(u)}{du},z'=\frac{dC_z(u)}{du} \tag{3}$$

Substituting Eq. (3) into Eq. (2) yields

$$\frac{du}{dt}=\frac{V(t)}{\sqrt{(x')^2+(y')^2+(z')^2}} \tag{4}$$

Although our goal is to get the value of $u$ at an arbitrary time $t$, it is difficult to obtain the closed-form solution of Eq. (4). In practice, alternatively, a recursive solution based on Taylor's expansion around $t=KT$ can be accepted

$$u_{k+1}=u_k+T\left.\frac{du}{dt}\right|_{t=KT}+\frac{T^2}{2}\left.\frac{d^2u}{dt^2}\right|_{t=KT}+\varepsilon_{u=u_k} \tag{5}$$

The second order derivative of $u$ with respect to $t$ is denoted

$$\frac{d^2u}{dt^2}=-\frac{V(t)\cdot(x'\cdot x''+y'\cdot y''+z'\cdot z'')\cdot\frac{du}{dt}}{\left(x'^2+y'^2+z'^2\right)^{\frac{3}{2}}}$$
$$x''=\frac{d^2C_x(u)}{du^2},y''=\frac{d^2C_y(u)}{du^2},z''=\frac{d^2C_z(u)}{du^2} \tag{6}$$

Substituting Eq. (4) and Eq. (6) into Eq. (5) yields the equation of 3-dimension parametric curve interpolator based on second order Taylor's expansion:

$$u_{k+1}=u_k+\left.\frac{TV(t)}{\sqrt{\left(x'^2+y'^2+z'^2\right)}}\right|_{t=kT}-\left.\frac{T^2V^2(t)\cdot(x'\cdot x''+y'\cdot y''+z'\cdot z'')}{2\cdot\left(x'^2+y'^2+z'^2\right)^2}\right|_{t=kT} \tag{7}$$

Form Eq. (7), the value of the next trajectory point on the curve $C(u)$ at the time $t_{k+1}$can be calculated by the current value of $u_k$ and the first and second order derivative of the current position($x_k$, $y_k$, $z_k$) with respect to the independent parameter $u$. Repeat this process, a vector of $u$ ($0\le u\le 1$) is obtained to determine a parametric curve with its segmentation based on the equal increment of the segments length. However, this widespread used method is an open-loop method by which feedrate can greatly deviate from the desired one. To overcome this drawback, a close-loop adaptive Adams-Bashforth method is proposed to generate the value of $u_{k+1}$ so that the feedrate can be limited in a tolerance range.

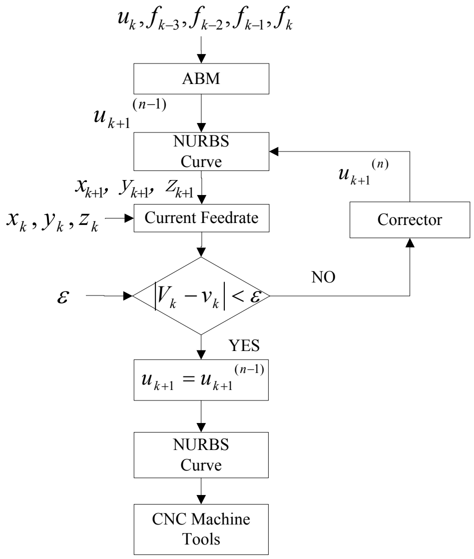


Fig. 1. Flowchart of the adaptive interpolator.

### C. Adaptive interpolator

The concept of the feedback in interpolation is firstly introduced by Lo [16] who used the desired feedrate and the current federate. This approach enables users to repeatedly update the value of $u_k$ until the current feedrate falls into the tolerance range. The flowchart of adaptive scheme is shown in Fig. 1.

The first step is to predict the next trajectory point. For simplification, Eq. (2) can be written as

$$\frac{du}{dt} = \frac{V(t)}{ds/du} = f(u) \tag{8}$$

A numerical algorithm Adams-Bashforth method [17] is used to solve Eq. (8) by which the predicted estimation of $u_{k+1}$ can be derived by

$$u_{k+1} = u_k + \frac{T}{24}(-9f(u_{k-3}) + 37f(u_{k-2}) - 59f(u_{k-1}) + 55f(u_k))$$
(9)

where $T$ is the time intervals..

The value of $u_0$, $u_1$, $u_2$ and $u_3$ have to be determined in advance to use Eq. (9). Here, $u_0$ is set to be zero, $u_1$, $u_2$, $u_3$ are given by the fourth order Runge-Kutta method

$$u_{k+1} = u_k + \frac{T}{6}(f_{k1} + 2f_{k2} + 2f_{k3} + f_{k4}) \tag{10}$$

where

$$f_{k1} = f(t_k, u_k)$$
$$f_{k2} = f(t_k + \frac{T}{2}, u_k + \frac{Tf_{k1}}{2})$$
$$f_{k3} = f(t_k + \frac{T}{2}, u_k + \frac{Tf_{k2}}{2})$$
$$f_{k4} = f(t_k + T, u_k + Tf_{k3})$$
$$f(t,u) = \frac{V(t)}{\sqrt{(x')^2 + (y')^2 + (z')^2}}$$

The second step is to obtain a corrected estimate of $u_{k+1}$. The ratio of the current feedrate and the desired feedrate is adopted as a proportional factor to adjust $u_{k+1}$ so that the resulting current feedrate can get proximately with the desired one.

$$u_{k+1}^{(n)} = u_k + \frac{V_k}{v_k^{(n-1)}}(u_{k+1}^{(n-1)} - u_k) \tag{11}$$

where $V_k$ is the desired feedrate at time $t_k$, $v_k^{(n-1)}$ is the resulting feedrate after $n$-1 iterations, which is defined as

$$v_k^{(n-1)} = \frac{|C(u_{k+1}^{(n-1)}) - C(u_k)|}{T} \tag{12}$$

Let $\varepsilon^{(n-1)} = |v_k^{(n-1)} - V_k| / V_k$ be the feedrate error after $n$-1 iterations, $\varepsilon$ is the specified feedrate tolerance, the terminated condition for the iteration is

$$\varepsilon^{(n-1)} \le \varepsilon \tag{13}$$

Thus, the proposed algorithm achieves the specified feedrate accuracy by the two steps provided above. In order to terminate the iteration, the terminated condition must be met after several times of the iteration and it requires the time intervals $T$ to be satisfied with the convergent condition which is presented in the next section.

### D. Convergent condition

The function iteration will not converge unless a tiny step size $T$ is used [18]. However, small $T$ will bring some undesirable drawbacks such as enormous data size. The tradeoff between the convergence and computation becomes an issue. In the section, we will find a bound value of $T$ within which the convergent condition is always to be satisfied.

Consider the values of $u_{k+1}$ at $n$ and $n$-1 times of iteration deriving by Eq. (11)

$$u_{k+1}^{(n)} = u_k + \frac{V_k}{u_k^{(n-1)}}(u_{k+1}^{(n-1)} - u_k) \tag{14}$$

$$u_{k+1}^{(n-1)} = u_k + \frac{V_k}{v_k^{(n-2)}}(u_{k+1}^{(n-2)} - u_k) \tag{15}$$

Subtracting (15) from (14) yields

$$u_{k+1}^{(n)} - u_{k+1}^{(n-1)} = \frac{V_k}{v_k^{(n-1)}}(u_{k+1}^{(n-1)} - u_k) - \frac{V_k}{v_k^{(n-2)}}(u_{k+1}^{(n-2)} - u_k)$$
(16)

where

$$v_k^{(n-1)} = \frac{|C(u_{k+1}^{(n-1)}) - C(u_k)|}{T}$$
$$v_k^{(n-2)} = \frac{|C(u_{k+1}^{(n-2)}) - C(u_k)|}{T}$$

then Eq. (17) can be derived as follows from Eq. (16)

$$u_{k+1}^{(n)} - u_{k+1}^{(n-1)} = \frac{TV_k}{\left|C(u_{k+1}^{(n-1)}) - C(u_k)\right|}(u_{k+1}^{(n-1)} - u_k) - \frac{TV_k}{\left|C(u_{k+1}^{(n-2)}) - C(u_k)\right|}(u_{k+1}^{(n-2)} - u_k) \tag{17}$$

Denote $h$=$u_{k+1}$-$u_k$, the derivative of $C(u)$ with respect to $u$ at $u_{k+1}$ is defined as

$$C'(u_{k+1}) = \lim_{h\to 0}\frac{C(u_{k+1}) - (u_k)}{h}$$

Eq. (17) can be written as

$$u_{k+1}^{(n)} - u_{k+1}^{(n-1)} = TV_k\left(\frac{1}{\left|C'(u_{k+1}^{(n-1)})\right|} - \frac{1}{\left|C'(u_{k+1}^{(n-2)})\right|}\right) \tag{18}$$

Since $1/\left|C'(u)\right|$ has a bounded first order derivative, from Lipschitz function

$$\left|\frac{1}{\left|C'(u_{k+1}^{(n-1)})\right|} - \frac{1}{\left|C'(u_{k+1}^{(n-2)})\right|}\right| \le L_1\left|u_{k+1}^{(n-1)} - u_{k+1}^{(n-2)}\right| \tag{19}$$

where $L_1$ is defined as the maximum value of $\frac{d}{du}\frac{1}{\left|C'(u)\right|}$ with the value of $u$ between $u_{k+1}^{(n-2)}$ and $u_{k+1}^{(n-1)}$, and

$$L_1 = \max\left|\frac{d}{du}\frac{1}{\left|C'(u)\right|}\right|$$

Substituting Eq. (19) into Eq. (18) yields

$$\left|u_{k+1}^{(n)} - u_{k+1}^{(n-1)}\right| \le TV_k L_1\left|u_{k+1}^{(n-1)} - u_{k+1}^{(n-2)}\right| \tag{20}$$

By repeating this process, Eq. (21) can be obtained

$$\left|u_{k+1}^{(n)} - u_{k+1}^{(n-1)}\right| \le TV_k L_1 \cdot TV_k L_2 \cdots TV_k L_{n-1}\left|u_{k+1}^{(1)} - u_{k+1}^{(0)}\right| \tag{21}$$

Let $L$ denote the maximum value of $L_i$ ($i = 1,2,\ldots,n-1$), Eq. (21) can be expressed as

$$\left|u_{k+1}^{(n)} - u_{k+1}^{(n-1)}\right| \le (TV_k L)^{n-1}\left|u_{k+1}^{(1)} - u_{k+1}^{(0)}\right|$$

Since $u_{k+1}^{(1)}$ and $u_{k+1}^{(0)}$ are bounded, the sequence $u_{k+1}^{(n)}$ is convergent when $TV_kL$ is under the convergent condition $TV_kL<1$. Assume $L$ occurs after $k$ times of iteration, written as $\left|\frac{d}{du}\frac{1}{\left|C'(u)\right|}\right|_{u=u_{k+1}}$, the convergent condition is

$$TV_k\left|\frac{d}{du}\frac{1}{\left|C'(u)\right|}\right|_{u=u_{k+1}} < 1$$

So we can get

$$T < \frac{1}{V_k\left|\frac{d}{du}\frac{1}{\left|C'(u)\right|}\right|_{u=u_{k+1}}} \tag{22}$$

As a result, Eq. (22) is the convergent condition for the adaptive interpolator, if it is satisfied, then $\left|u_{k+1}^{(n)} - u_{k+1}^{(n-1)}\right| \to 0$ when $n\to\infty$, from Eq. (11) the proximity of feedrate $v_k^{(n-1)} \to V_k$ can be achieved eventually.

## III. SIMULATION RESULTS

The evaluation of the conventional interpolator and adaptive interpolator along a NURBS curve is measured by using simulation results based on the feedrate profile error. Fig. 2 shows a NURBS tool path with its control points, the weight vector and knot vector are given as

Control points: (0,0), (−100,−100), (−100,100) (0,0) (100,−100) (100,100) (0,0),

Weight vector: [5, 5, 10, 1, 10, 5, 5],

Knot vector: [0, 0, 0, 0.25, 0.5, 0.5, 0.75, 1, 1, 1].

The length of the tool path is calculated by Simpson's rule. In this case, the total length is $S$ = 679.5234 mm. Considering the convergent condition, sampling time is chosen as $T$= 0.008 s. A trapezoidal feedrate profile is adopted as the desired feedrate with the maximum federate $V$= 100 mm/s, acceleration $A$= 150 mm/s$^2$. The feedrate tolerance is specified as $\varepsilon$ = 1 mm/s.

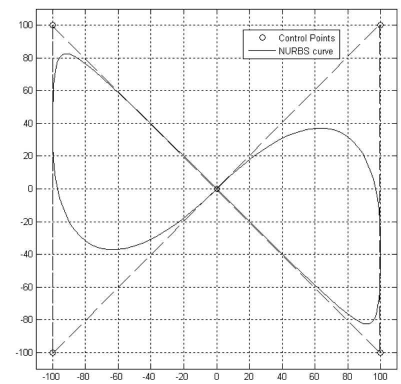


Fig. 2. NURBS curve

By the parameters specified above, total interpolation time can be calculated as $T_{total}=S/V+V/A$ =7.4619s. Table 1 reveals the simulation results using Taylor's expansion method and adaptive ABM. The interpolation feedrate based on these methods and their errors comparing with the desired feedrate are shown in Fig. 3.

Table 1 Simulation results for discussed interpolators (mm/s)

| | 1st order Taylor | 2nd order Taylor | Adaptive ABM |
|---|---|---|---|
| Max feedrate | 5.5303 | 3.4747 | 1.7198 |
| Min feedrate | -6.379 | -4.557 | -1.351 |

From Fig.3 it can be concluded that the interpolator based on the second order Taylor's expansion performed a better quality than that based on first order Taylor's expansion. In contrast with the Taylor's method by which error can just be controlled by adjusting step size, the adaptive ABM algorithm can limit the feedrate error in a pre-specified range. As a result, the demand for the different feedrate accuracy can be satisfied.

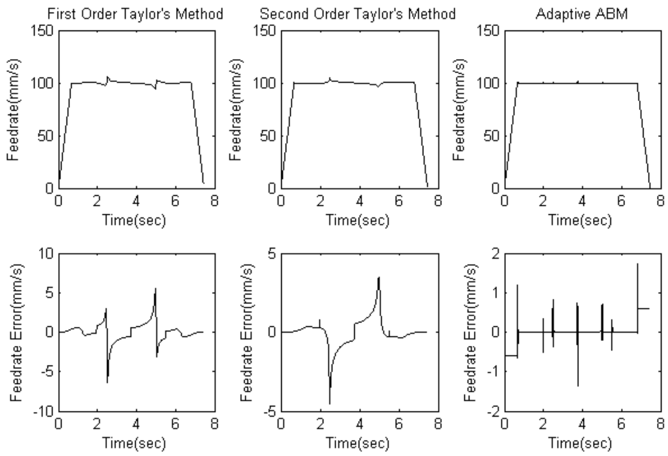


Fig. 3 Feedrate profiles and feedrate errors

## IV. EXPERIMENT RESULTS

The experimental setup of an X-Y table is used to verify the practical application of the adaptive ABM algorithm. This platform is developed using a motion control board with a TI TMS320F2812 DSP which would perform advanced capabilities of both calculation and motion control in real-time. Program code of NURBS expressions, feedrate planning, and interpolation algorithm companied with the motion control are written in C language and executed by the DSP. Grating measurements are used to detect transient position of the X-Y table along X, Y direction respectively to practice adaptive interpolation.

The X-Y table is driven by two DC servo motors, and each motor performs a resolution of 4000 steps per revolution. Position feedback is practiced by grating measurements with the pitch of screw in both X and Y directions are 16 mm. Other parameters of the interpolator is set as follows

The maximum feedrate: $V$= 100 mm/s;

Acceleration: $A$= 150 mm/s$^2$,

Sampling time: $T$= 0.008 s,

Feedrate tolerance: $\varepsilon$ = 3 mm/s.

The resulting tool path (shown in Fig. 4) is obtained by the grating measurements and the interpolation feedrate is derived from dividing the length between two adjacent trajectory points by the interpolation interval. Interpolation by the first and second order of Taylor's expansion is experimented to provide a comparative result by giving an inferior performance to that of interpolation by adaptive Adams-Bashforth algorithm. Table 2 shows the experimental results using Taylor's expansion method and adaptive ABM respectively. The contour error and feedrate profile performed by using first and second order Taylor's expansion and adaptive ABM are depicted in Fig. 5-Fig. 7.

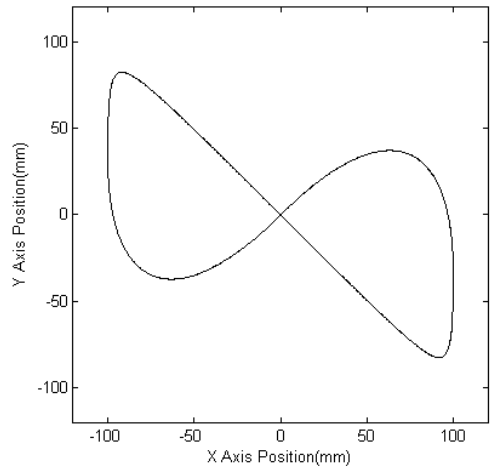


Fig. 4 Desired contour (dashed) and resulting contour (solid) with adaptive ABM

Table 2 Experimental results for discussed interpolators

| | 1st order Taylor | 2nd order Taylor | Adaptive ABM |
|---|---|---|---|
| Max couture error(mm) | 0.7097 | 0.465 | 0.1653 |
| Max feedrate error(mm/s) | 7.7524 | 6.3433 | 3.2118 |

As mentioned above, the convergent condition for adaptive algorithm is that the interpolation interval must be limited in a specified range determined by the desired feedrate for a definite trajectory. Nevertheless, since the interpolation process is in real-time, besides enormous data size, small interpolation interval would cause other disastrous consequences. At each interpolation period, adaptive Adams-Bashforth method is adopted to calculate the next trajectory point meanwhile X-Y table is still running at the feedrate calculated in the last interpolation period. Assume the interpolation interval is too small or the feedrate is too large, the current point obtained from grating measurements may have gone beyond the next trajectory point calculated in an interpolation period, in this condition the iteration will never be convergent.

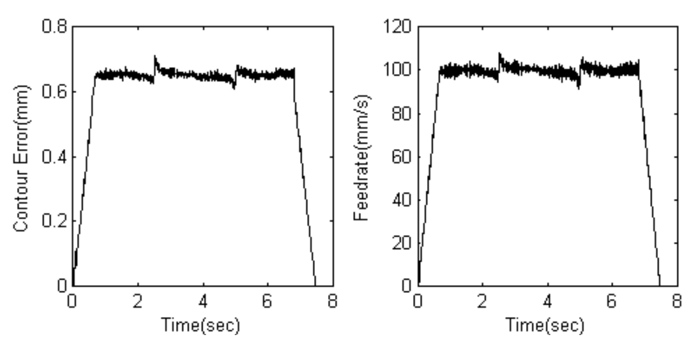


Fig. 5 Contour error and feedrate profile using first order Taylor's expansion method

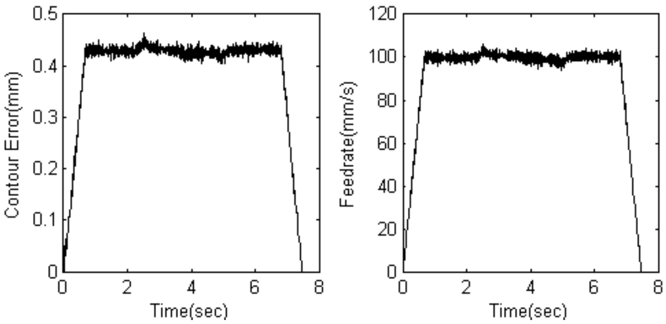


Fig. 6 Contour error and feedrate profile using second order Taylor's expansion method

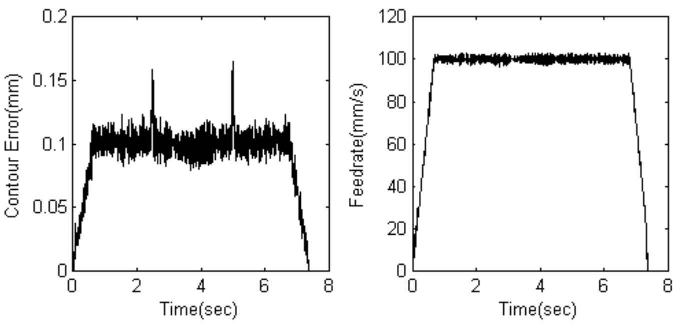


Fig. 7 Contour error and feedrate profile using second order Taylor's expansion method

## V. CONCLUSION

For free-form curve, the essential of interpolation is to keep in real-time and achieve expected accuracy. This paper proposed an effective method to predict and correct the next trajectory point in real-time. Since the deviation between current feedrate and the desired feedrate is pre-determined, the specified accuracy is ensured by this adaptive interpolator. Simulation and experimental results show that the adaptive interpolator tracks a more accurate tool path by providing an access to control feedrate deviation which is uncontrollable by using conventional Taylor's expansion method.

## VI. REFERENCES

[1] D.C.H. Yang and T.Kong, "Parametric interpolator versus linear interpolator for precision CNC machining". Computer-Aided Design, vol. 26, 1994, pp 225-234

[2] Rida T. Farouki, Carla Manni, Alessandra Sestini, "Real-time CNC interpolators for Bezier conics". Computer Aided Geometric Design vol. 18, 2001, pp 639-655

[3] Piegl L, Tiller W, The NURBS Book, 2nd edn. Springer. Berlin Heidelberg New York, 1997

[4] S. Bedi, I. Ali, N. Quan, "Advanced techniques for CNC machine". Engineering for Industry, vol. 115, 1993, pp 329-336

[5] M. Shpitalni, Y. Koren, C.C.Lo, "Realtime curve interpolators". Computer-Aided Design vol. 26, 1994, pp 832-838

[6] R.F.Farouki, Y.F.Tsai, "Exact Taylor series coefficient for variable-feedrate CNC curve interpolator". Computer-Aided Design vol. 33, 2001, pp 155-165

[7] M.C.Tsai, C.W.Cheng, "A real-time predictor-corrector interpolator for CNC machining". J. Manuf. Sci. Eng. vol. 125, 2003, pp 449-460

[8] Q.G.Zhang, R.B.Greenway, "Development and implementation of a NURBS curve motion interpolator". Robotics and Computer-Integrated Manufacturing vol.14, 1998, pp 27-36

[9] S.S.Yeh, P.L.Hsu, "The speed-controlled interpolator for machining parametric curves". Computer-Aided Design vol. 31, 1999, pp 349-357

[10] H.T.Yau, J.B.Wang, "Fast Bezier interpolator with real-time look-ahead function for high-accuracy machining", Int. J. Mach. Tools Manuf. vol. 47, 2007, pp 1518-1529

[11] K.Erkorkmaz, Y.Altintas, "High speed CNC system design". Int. J. Mach. Tools Manuf. vol. 41, 2001, pp 1323-1345

[12] W.T.Lei, M.P.Sung, L.Y.Lin, J.J.Huang, "Fast real-time NURBS path interpolation for CNC machine tools". Int. J. Mach. Tools Manuf. vol. 47, 2007, pp 1530-1541

[13] S.S.Yeh, P.L.Hsu, "Adaptive-feedrate interpolation for parametric curves with a confined chord error". Computer-Aided Design vol. 34, 2002, pp 229-237

[14] T.Yong, R.Narayanaswami, "A parametric interpolator with confined chord errors, acceleration and deceleration for NC machining". Computer-Aided Design vol. 35, 2003, pp 1249-1259

[15] J.Y.Lai, K.Y.Lin, S.J.Tseng, W.D.Ueng, "On the development of a parametric interpolator with confined chord error, feedrate, acceleration and jerk". Int J Adv Manuf Technol. Doi: 10.1007/s00170-007-0954-7, 2007

[16] C.C. Lo, "Feedback command generators for CNC machine tools". J. Manuf. Sci. Eng. vol. 119, 1997, pp 587-592

[17] W.W.Yang, W.W.Cao, T.S.Chung, J.Morris, Applied Numerical Methods Using MATLAB. John Wiley & Sons, Inc., Hoboken, New Jersey. 2005

[18] D.I.Kim, J.W.Jeon, S.kim, "software acceleration deceleration methods for industrial robots and CNC machine tools". Mechatronics vol. 4, 1994, pp 37-53

[19] K.S.Fu, R.C.Gonzalez, C.S.G.Lee, Robotics: Control, Sensing, Vision and Interlligence. McGraw-Hill. New York. 1997