DOI: 10.1007/s12541-012-0124-1

Fast and Precision NURBS Interpolator for CNC Systems

Dae-Kyun Baek[1], **Tae-Jo Ko**[2], and **Seung-Han Yang**[3,#]

1 Institute of Mechanical Technology, Kyungpook National University, 1370, Sankyuk-dong, Buk-gu, Daegu, South Korea, 702-701
2 School of Mechanical Engineering, Yeungnam University, 214-1, Dae-dong, Gyeongsan, Gyeongbuk, South Korea, 712-749
3 School of Mechanical Engineering, Kyungpook National University, 1370, Sankyuk-dong, Buk-gu, Daegu, South Korea, 702-701
Corresponding Author / E-mail: syang@knu.ac.kr, TEL: +82-53-950-6569, FAX: +82-53-950-6550

KEYWORDS: CNC controller, NURBS interpolation, Parameter update, Parametric curve

This paper presents a performance evaluation of a NURBS interpolator that uses not Taylor's expansion method but the recursive characteristics of NURBS. Taylor's expansion has mostly been used for NURBS interpolation. However, it is very complicated and gives an indispensable truncation error. A fast and precision NURBS interpolator replacing Taylor's expansion is presented for CNC systems in robots and CNC machine tools. The presented interpolation algorithm uses the recursive equation of the NURBS formula rather than Taylor's expansion. A simulation study is conducted to demonstrate the advantages of this proposed interpolator compared with those using Taylor's equation. The feedrate error and calculation time are simulated in the performance evaluation. The recursive method of NURBS interpolation is faster and more accurate than Taylor's expansion.

Manuscript received: November 25, 2011 / Accepted: January 19, 2012

1. Introduction

Free-formed surfaces are widely used in modern industrial products such as automobiles, electric or electronic goods, household items, and so on. These are represented by 3 dimensional CAD modeling systems that use parametric or nonparametric mathematical expressions, and finally manufactured by CNC systems. However, conventional CNC systems such as robotics, machine tools, and other automation devices deal with axis motion along a straight-line or circular path, because they only provide linear and circular interpolation with respect to the parametric surface. For these systems, very long CNC programs containing a great number of linear segments are often needed. In this case, the large storage space and the long computing time required by these programs are the main problems that need to be addressed. Another drawback is that the velocity is not continuous at the conjunction of the line segments. There is also a conflict between the accuracy and efficiency, which is inherent in the segmentation process. The conflict arises from the fact that higher accuracy results in smaller segments. Since this are traced by the existing linear or circular interpolator on a one-at-a-time basis, their processing induces repeated acceleration-deceleration cycles on the CNC machine, thus resulting in machining inaccuracies and substantially increasing the whole machining time. Therefore, studies have been performed on the development of general interpolators which can express various lines and curves in a block, control the motion smoothly.[1] This is done by transferring the information of the parametric curves into the CNC systems, which then interpolate the parametric curves.

NURBS models have long been favored in CAD systems[2] because they offer an exact uniform representation of both analytical and free-form parametric curves. Most researchers have developed a variety of interpolation algorithms for parametric curves using Taylor's expansion. The most representative method is to use a first-order approximation of Taylor's expansion and the derivative of the parametric curve, and to get equal chord lengths.[3-5] However, since the first-order approximation leads to a high truncation error, other researchers have studied methods of reducing the interpolation errors using a second-order approximation of Taylor's expansion and varying federate.[6-8] Parametric interpolators with confined acceleration/deceleration were also proposed using the second-order approximation of Taylor's expansion.[9-13] Lin et al.[14] developed a dynamics-based NURBS interpolator with real-time look-ahead algorithm. Liu et al.[15] proposed a NURBS interpolator with the integration of the machining dynamics. Du et al.[16] presented an adaptive NURBS interpolator with acceleration/deceleration control scheme by considering a preset jerk range. These parametric interpolators for ACC/Dec and machining dynamics also used the second-order approximation of

Taylor's expansion. Cheng et al.[17] also studied a contour error reduction for the NURBS curve with a fuzzy feedrate regulator using the second order approximation of Taylor's expansion.

Whatever interpolation algorithms are used, the first-order approximation of Taylor's expansion leads to a high interpolation error and the second-order approximation results in a complicated calculation and inaccurate feedrate. The proposed algorithm interpolates the NURBS curve using the NURBS equation itself.[18] A temporary incremental value is calculated using the proportional difference equation from the previous incremental value and the previous chord length, and the final current incremental value is determined after recursively updating the temporary incremental value using the NURBS equation. The proposed interpolator was compared with a Taylor's expansion interpolator to evaluate its performance. The variables of the proposed interpolator that determine the accuracy and calculation time were varied, and its performance was evaluated in comparison with the first-order as well as second-order approximations of Taylor's expansion. The feedrate errors and calculation times were simulated in the performance evaluation.

2. NURBS representation

NURBS curves[2] are represented parametrically by the following Eq. (1):

$$\mathbf{P}(u) = \frac{\sum_{i=0}^{n} N_{i,k}(u) w_i \mathbf{B}_i}{\sum_{i=0}^{n} N_{i,k}(u) w_i} \tag{1}$$

where $N_{i,k}(u)$ is a blending function defined by the recursive formula:

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

and

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

where $[u_i, \cdots, u_{i+k}]$ is the knot vector, u the parameter, $\mathbf{P}(u)$ the vector to a point defined at some value of u, $B_i(u)$ the control points, in the 3-D case, $\mathbf{B}_i = \{X_i, Y_i, Z_i\}^T$, n the number of control points-1, and w_i the weight factors. The blending function in Eqs. (2) and (3) has recursive characteristics that constitute the basic concept of the proposed interpolation algorithm.

3. NURBS interpolation algorithm

3.1 NURBS interpolation with Taylor's expansion

The purpose of interpolating the parametric curves is to find the incremental value of the parameters as follows:

$$u_{j+1} = u_j + \Delta u_j \tag{4}$$

where u_j is the current parameter, u_{j+1} is the next parameter, and Δu_j is the incremental value. The interpolated points are calculated by substituting the updated parameter into the corresponding mathematical model. Most existing parametric interpolators are based on Taylor's expansion. The first-order approximation of Taylor's expansion for calculating u_{j+1} is as follows:

$$u_{j+1} = u_j + \frac{V_j \Delta T}{\left\| \frac{d\,\mathrm{P}(u)}{du} \right\|_{u=u_j}} \tag{5}$$

where ΔT is the sampling interval (sec), and V_j is the instantaneous speed (mm/sec). The second-order approximation of Taylor's expansion is

$$u_{j+1} = u_j + \frac{V_j \Delta T}{\left\| \frac{d\mathrm{P}(u)}{du} \right\|_{u=u_j}} + \frac{\Delta T^2}{2} \left(\frac{\left. \frac{dV_j}{dt} \right|_{t=t_j}}{\left\| \frac{d\mathrm{P}(u)}{du} \right\|_{u=u_j}} - \frac{V_j^2 \left. \left(\frac{d\mathrm{P}(u)}{dt} \cdot \frac{d^2\mathrm{P}(u)}{du^2} \right) \right|_{u=u_j}}{\left\| \frac{d\mathrm{P}(u)}{du} \right\|_{u=u_j}^4} \right). \tag{6}$$

As shown in Eq. (6), the second-order approximation of Taylor's expansion is so complex that it leads to a large computation load. In the following section, the NURBS interpolation algorithm using the recursive characteristics of the NURBS equation is presented.

3.2 NURBS interpolation with recursive method

In this research, the recursive characteristics of NURBS were used to develop a NURBS interpolator. The chord length s_j between points $\mathrm{P}(u_{j+1})$ and $\mathrm{P}(u_j)$ on the NURBS curve is expressed as in Eq. (7).

$$s_j = \left| \mathbf{P}(u_{j+1}) - \mathbf{P}(u_j) \right| \tag{7}$$

The current chord length s_j is similar to the adjacent lengths s_{j-1} and s_{j+1}, and this is the basic idea used to determine the incremental value Δu_{j-1} in order to make the chord lengths constant. Fig. 1 shows how to determine any incremental value Δu_j from the previous incremental value Δu_j and the previous chord length s_{j-1}. The desired chord length is d, which is the resolution of the CNC system, and $\mathrm{P}(u_j)$ is a point vector interpolated by Δu_{j-1}. As shown in Fig. 1, the incremental value Δu_j is updated m times from Δu_{j_1} to Δu_{j_m}. As the incremental value is updated, the interpolated chord length becomes closer to the desired chord length, d.

The updating rule uses the recursive characteristics of NURBS that the adjacent chord lengths are similar. Assuming that chord length has a linear relationship with the incremental value in a very small local region, the first temporary incremental value Δu_{j_1} can be calculated as follows:

$$\Delta u_{j-1} : s_{j-1} = \Delta u_{j_1} : d, \quad j > 1 \tag{8}$$

where $s_{j-1} = \left| \mathbf{P}(u_j) - \mathbf{P}(u_{j-1}) \right|$. Since the previous incremental value, Δu_{j-1} gives the chord length s_{j-1}, the chord length d is proportional to Δu_{j_1}. Eq. (8) is rearranged as in Eq. (9).

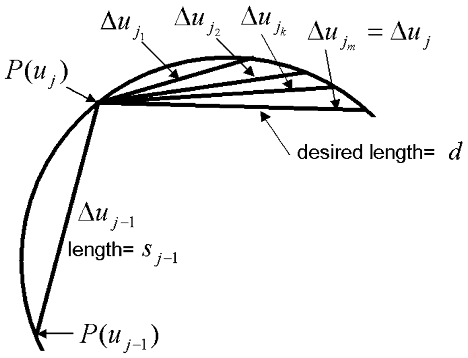

Fig. 1 The concept of interpolation

$$\Delta u_{j_1} = \frac{d \cdot \Delta u_{j-1}}{s_{j-1}}, \quad j > 1 \tag{9}$$

However, as the temporary incremental value Δu_{j_1} does not give the exact chord length d, an updated value Δu_{j_k} can be successively calculated using the generalized equation in Eq. (10).

$$\Delta u_{j_k} = \frac{d \cdot \Delta u_{j_{(k-1)}}}{s_{j_{(k-1)}}}, \quad j > 1, k = 2 \text{ ... m} \tag{10}$$

where $s_{j_{(k-1)}} = \left|\mathbf{P}(u_{j_{(k-1)}}) - \mathbf{P}(u_j)\right|$, and $u_{j_{(k-1)}} = u_j + \Delta u_{j_{(k-1)}}$. The final temporary value Δu_{j_m} becomes the current incremental value Δu_j. If the temporary parameter $u_{j_{(k-1)}}$ in Eq. (10) is larger than $u_{\max}$ near the last interpolation, the point $\mathbf{P}(u_{j_{(k-1)}})$ cannot be calculated from the NURBS equations. This problem can be solved as shown in Eq. (11).

$$u_{j_{(k-1)}} = u_{\max}, \text{ if } u_{j_{(k-1)}} > u_{\max} \tag{11}$$

Accordingly, Eqs. (9), (10), and (11) determine the parameter u_j successively, such that the curve segment s_j (length of movement in each sampling time interval T) is constant and, thus, can be used in the feedrate or velocity control. The number of iterations, m, affects not only the calculation time, but also the accuracy of the interpolation. If m in Eq. (10) is large, the calculation time is long but the interpolation is precise. Otherwise, the calculation time is short and the interpolation is not precise. Eqs. (9), (10), and (11) are used for determining the current incremental value from the previous one and the previous chord length. However, there is no previous incremental value and chord length available for determining the initial incremental value Δu_0. Therefore, the total sum of the lengths between the adjacent control points and the difference between $u_{\max}$ and $u_{\min}$ were used to estimate the previous incremental value and the previous chord length for the calculation of the initial incremental value. The total sum of the lengths between the adjacent control points is expressed in Eq. (12).

$$t = \sum_{i=0}^{n-1} \left|\mathbf{B}_{i+1} - \mathbf{B}_i\right| \tag{12}$$

If u_s is the difference between the maximum and the minimum knot values, a proportional expression for the first temporary incremental value, Δu_{0_1} can be written as follows:

$$u_s : t = \Delta u_{0_1} : d \tag{13}$$

Eq. (13) can be rewritten as Eq. (14).

$$\Delta u_{0_1} = \frac{u_s d}{t} \tag{14}$$

If $\mathbf{P}(u_{0_1})$ is the interpolated point with $u_{0_1} = u_{\min} + \Delta u_{0_1}$ from the NURBS equation, the chord length between the starting point and $\mathbf{P}(u_{0_1})$ is as follows:

$$s_{0_1} = \left|\mathbf{P}(u_{0_1}) - \mathbf{B}_0\right| \tag{15}$$

Consequently, the generalized equation to determine the initial incremental value can be written as Eq. (16).

$$\Delta u_{0_k} = \frac{d \cdot \Delta u_{0_{(k-1)}}}{s_{0_{(k-1)}}}, \quad k = 2 \ldots l \tag{16}$$

where $s_{0_{(k-1)}} = \left|\mathbf{P}(u_{0_{(k-1)}}) - \mathbf{B}_0\right|$, and $u_{0_{(k-1)}} = u_{\min} + \Delta u_{0_{(k-1)}}$. The final initial incremental value Δu_0 is equal to Δu_{0_l}. The parameter l in Eq. (16) is the number of iterations required for determining Δu_0, and it must be greater than m in Eq. (10) because the previous incremental value and the previous chord length for determining the initial incremental value are roughly calculated in Eq. (14).

Since the incremental value Δu_j in Eq. (10) is determined during machine motion, the iteration number m must be selected considering the calculation time. If $m = 2$, the incremental value Δu_j is determined using the NURBS equation once, because Eq. (9) does not use the NURBS equation, but the already known previous incremental value Δu_{j-1} and chord length s_{j-1}. Therefore, the number of times the NURBS equation is used to determine the incremental value Δu_j is m-1. The variable $n = m$-1 is defined as the recursive order, which means the number of times the NURBS equation is used for the interpolation. If $m = l$, the interpolation using Eq. (16) for Δu_0 has a greater error than that using Eq. (10) for Δu_j because Δu_{0_1} is roughly calculated in Eq. (14). Thus, l needs to be greater than m. Even if l is greater than m, the motion control is alright, because the initial incremental value is determined before the start of the motion. This makes the calculation time of the initial incremental value Δu_0 sufficient. Therefore, Δu_0 can be calculated by means of an algorithm using a tolerance value, as described in section 3.3, instead of the fixed iteration number l in Eq. (16).

3.3 Determination of the initial incremental value using an algorithm

The stepwise procedures for determining the initial incremental value of the NURBS curve parameter are as follows:

Step 1: Select a tolerance value e (%) of the first chord length to the desired chord length d, and calculate Δu_{0_1} from Eq. (14).

Step 2: Set $k = 2$.

Step 3: Calculate Δu_{0_k} as follows:

$$\Delta u_{0_k} = \frac{d \cdot \Delta u_{0_{(k-1)}}}{s_{0_{(k-1)}}}$$

where $s_{0_{(k-1)}} = \left|\mathbf{P}(u_{0_{(k-1)}}) - \mathbf{B}_0\right|$, and $u_{0_{(k-1)}} = u_{\min} + \Delta u_{0_{(k-1)}}$.

Step 4: Set $u_{0_k} = u_{\min} + \Delta u_{o_k}$, and obtain $\mathbf{P}(u_{0_k})$ and s_{0_k} using the NURBS equation.

Step 5: If $\left|\frac{d - s_{0_k}}{d}\right| \times 100 \geq e$, set $k = k + 1$, and go to Step 3. If not, select $\Delta u_{0_k} = \Delta u_0$ and finish the algorithm.

4. Simulation results

A simulation study was conducted to evaluate the performance of the proposed method with the NURBS curve of Fig. 2, and we focused on comparing this proposed algorithm with Taylor's expansion. The desired chord length of the interpolation d was 0.4 mm, the feedrate V was 200 mm/sec, and the sampling time T was 0.002 sec. The personal computer employed has a 3.4GHz CPU.

Fig. 3 shows the chord lengths with the constant incremental value $\Delta u_j = 0.0005$. As shown in Fig. 3, the maximum and minimum cord lengths are 0.85 mm and 0.074 mm, respectively, and the chord lengths do not vary dramatically, but little by little, as the parameter u_j increases, due to the recursive characteristics of the blending function in Eq. (3). The chord lengths should be the desired value of 0.4 mm, so that the NURBS interpolator with recursive method is used and compared with the Taylor expansion.

To obtain the initial incremental value u_0 in the simulation study, the algorithm in Section 3.3 was used with tolerance $e = 0.01$ %. Fig. 4 shows the chord lengths of the recursive NURBS interpolator with $n = 2$, in which the NURBS equation is used two times for interpolating. As shown in Fig. 4, the chord lengths are almost the same as the desired value 0.4 mm, and the proposed interpolator is very accurate. For the comparison between the proposed interpolation method and Taylor's expansion, the feedrate was simulated. The first order approximation of Taylor's expansion is compared with the proposed algorithm with $n = 1$, because the NURBS equation is used once. The performances of the algorithms are shown in Figs. 5 and 6, respectively. The feedrate of the first order approximation of Taylor's expansion varies within [194.53, 205.81] mm/sec, so that the maximum error is 5.81 mm/sec. The feedrate of the proposed interpolator with $n = 1$ varies within [199.66, 200.001] mm/sec, so that the maximum error is 0.34 mm/sec. The feedrate error of the recursive method using the NURBS equation once is about 6 % of the first order approximation of Taylor's expansion. As mentioned above, the proposed interpolator uses the NURBS equation in Eq. (10) and Taylor's expansion uses the derivative of NURBS equation in Eq. (5). The calculation time of the derivative of NURBS is longer than that using the NURBS equation to determine the incremental value. The calculation time of the proposed method with $n = 1$ is 6.37 μsec, whereas that of the first order approximation of Taylor's expansion is 25.15 μsec.

The second order approximation of Taylor's expansion is compared with the proposed algorithm with $n = 2$ using the NURBS equation twice, as shown in Figs. 7 and 8, respectively. As shown, the proposed method is much more accurate than Taylor's expansion. The feedrate of the second order approximation of Taylor's expansion varies within [199.8, 200.4] mm/sec, so that the maximum error is 0.4 mm/sec. The feedrate of the proposed interpolator with n = 2 varies within [199.99, 200.01] mm/sec, so that the maximum error is 0.01 mm/sec. The feedrate error of the recursive method using the NURBS equation twice is about 2.5 % of the second order approximation of Taylor's expansion. Even though the proposed algorithm with $n = 1$ in Fig. 6 uses NURBS equation once, it interpolates the NURBS curve more precisely than the second order approximation of Taylor's expansion in Fig. 7. The calculation time of the proposed interpolator with $n = 2$ is 9.54 μsec,

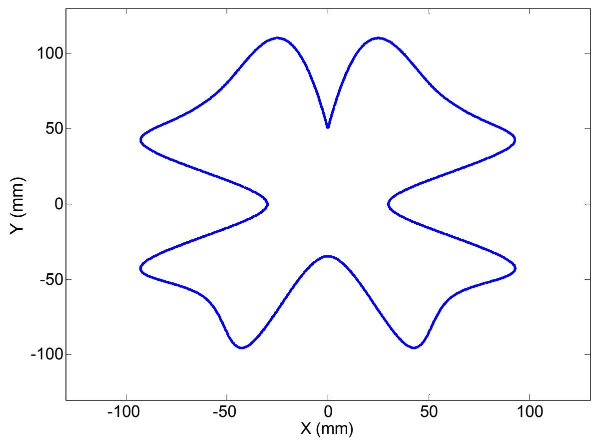

Fig. 2 NURBS curve for simulation

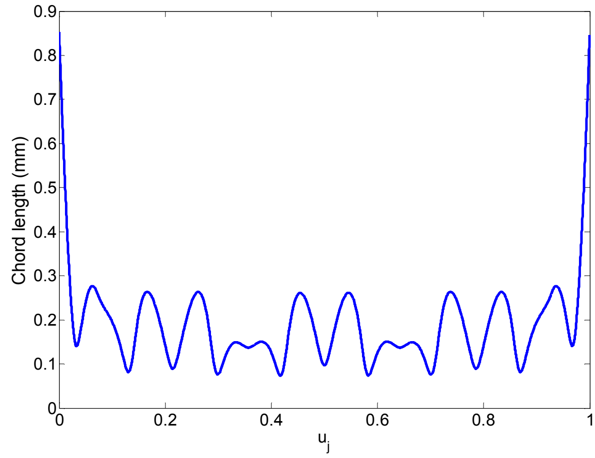

Fig. 3 The chord length with the constant incremental value (Δu_j = 0.0005)

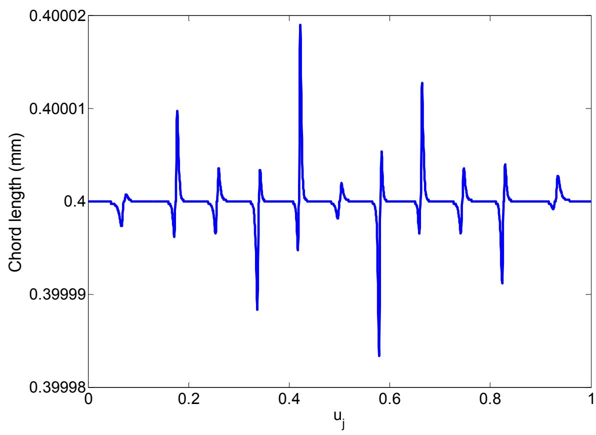

Fig. 4 The chord lengths with the proposed interpolation ($n = 2$)

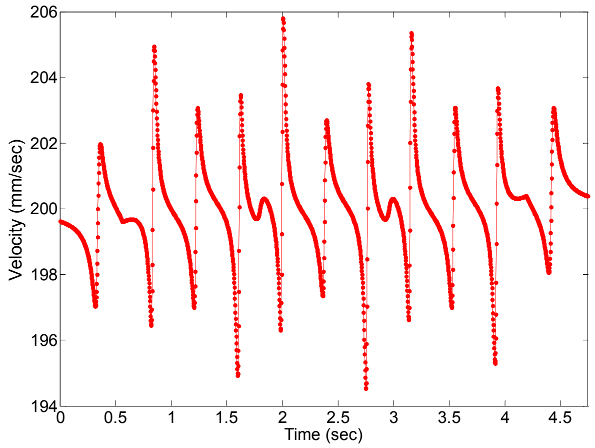

Fig. 5 Feedrate of the first-order approximation of Taylor's expansion

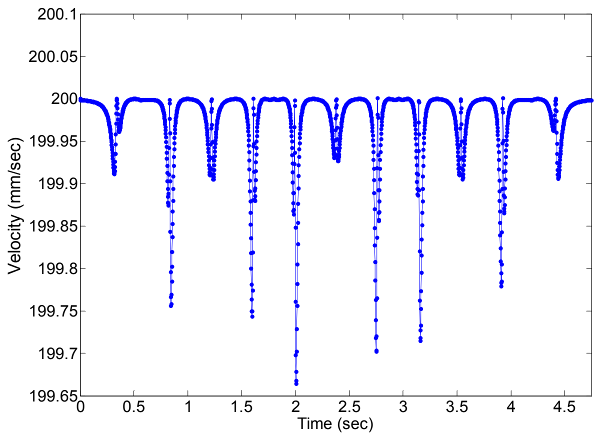

Fig. 6 Feedrate of the proposed interpolation (n = 1)

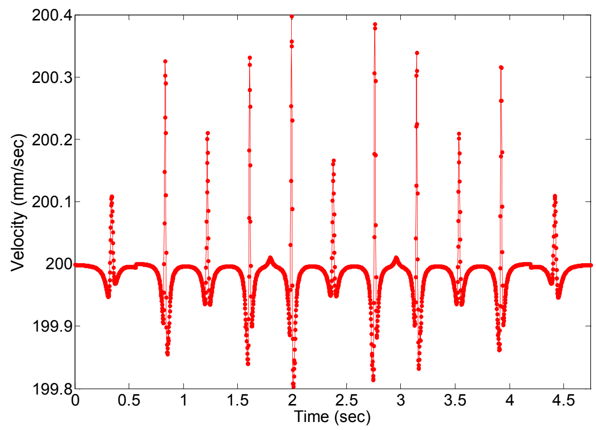

Fig. 7 Feedrate of the second-order approximation of Taylor's expansion

whereas that of the first order approximation of Taylor's expansion is 28.50 μsec. The proposed algorithm is much faster than the method based on Taylor's expansion.

Fig. 9 shows the feedrate of the proposed algorithm with $n = 3$ that uses the NURBS equation 3 times for interpolation. The maximum error of the feedrate with $n = 3$ is 2.64e-4 mm/sec and the proposed algorithm converges with $n = 4$, so that the feedrate is approximately equal to 200 mm/sec for all curve lengths. The

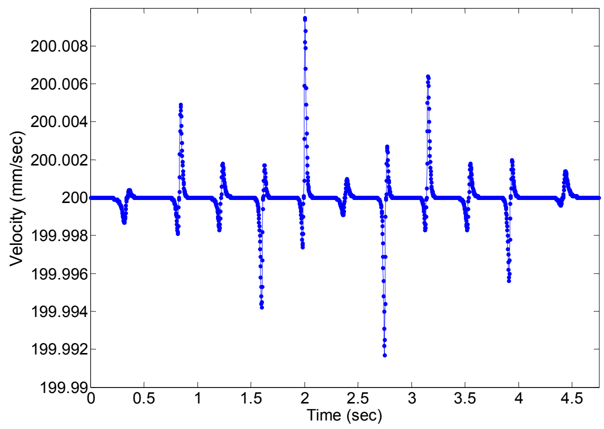

Fig. 8 Feedrate of the proposed interpolation (n = 2)

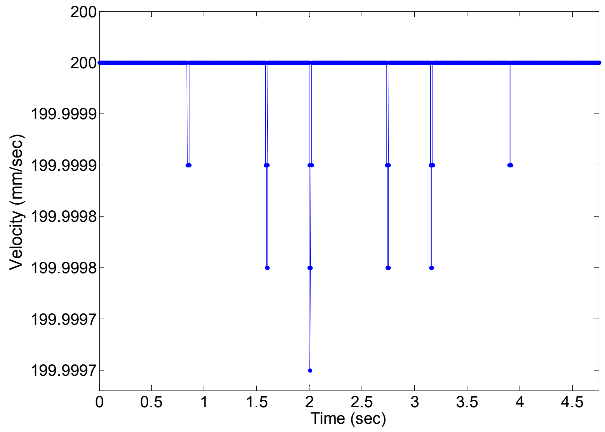

Fig. 9 Feedrate of the proposed interpolation (n = 3)

Table 1 Comparison of the maximum feedrate error (mm/sec)

The proposed algorithm		Taylor's expansion	
$n = 1$	0.34	1'st order	5.81
$n = 2$	0.01	2'nd order	0.40
$n = 3$	2.64e-4		
$n = 4$	1.01e-5		

Table 2 Comparison of the calculation time (μsec)

The proposed algorithm		Taylor's expansion	
$n = 1$	6.37	1'st order	25.15
$n = 2$	9.54	2'nd order	28.50
$n = 3$	12.64		
$n = 4$	15.80		

maximum error of the feedrate with $n = 4$ is 1.01e-5 mm/sec. The maximum feedrate error and the calculation time are listed in Tables 1 and 2, respectively. As the recursive order n increases, the feedrate error dramatically decreases, but the calculation time increases only slightly, because using the NURBS equation once takes very little time.

Another NURBS curve was simulated, as shown in Fig. 10, that varies the curvature more dramatically than that in Fig. 2. The second order approximation of Taylor's expansion is compared with the proposed algorithm with recursive order $n = 2$, as shown in Figs. 11 and 12, respectively. As shown, the feedrate errors of the two algorithms are larger than those in Figs. 7 and 8, respectively,

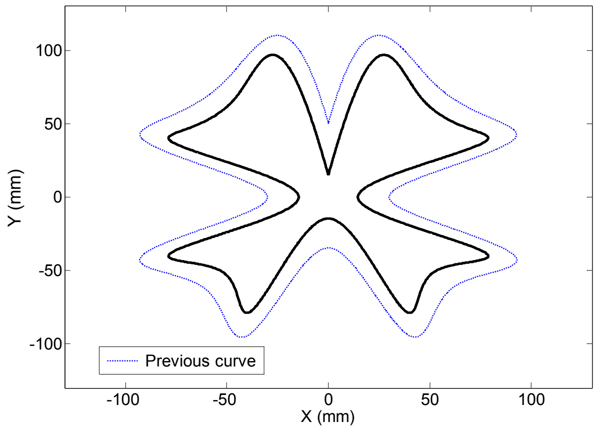

Fig. 10 Another example of a NURBS curve

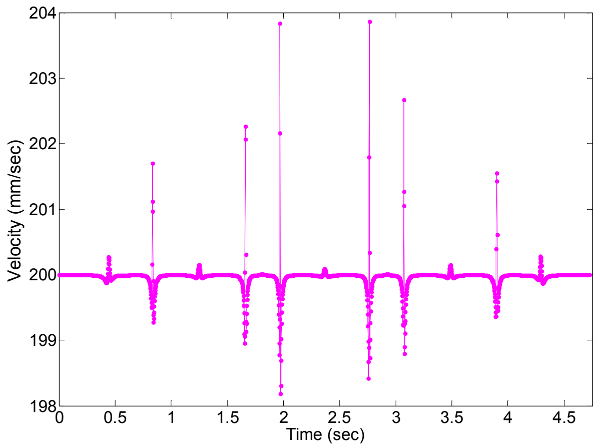

Fig. 11 Feedrate of the second-order approximation of Taylor's expansion

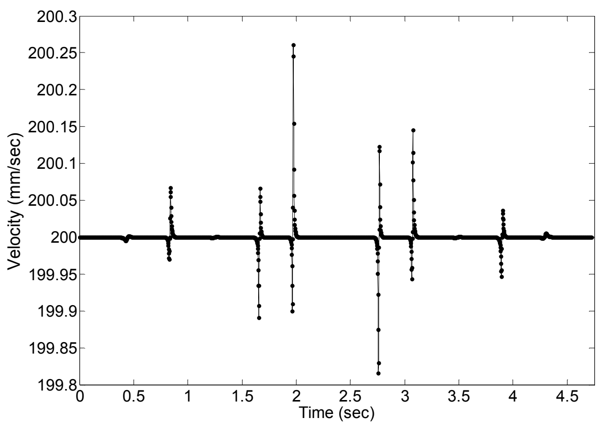

Fig. 12 Feedrate of the proposed interpolation (n = 2)

because the curvature of the curve is very small. The proposed method can reduce the feedrate error by using a higher recursive order, n.

The interpolations were carried out by increasing the recursive order n from $n = 3$ step by step. The feedrate errors gradually decrease as the recursive order n becomes larger. Examples of the feedrate with $n = 4$ and $n = 5$ in the maximum curvature are shown in Fig. 13. Interpolating the NURBS curve in Fig. 10, the proposed algorithm converged at $n = 6$, that is to say, the feedrate is almost

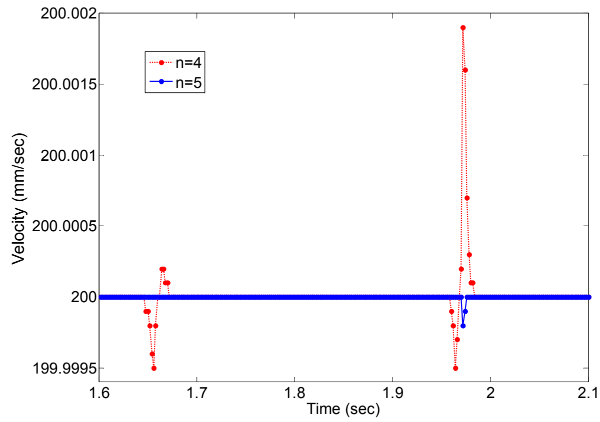

Fig. 13 Feedrate of the proposed interpolation (n = 4, n = 5)

equal to 200 mm/sec in the case of all of the curve lengths. The maximum error of the feedrate is 8.5e-5 mm/sec with $n = 6$. However, it is very difficult to obtain a higher order derivative than the second one to reduce the truncation error for Taylor's expansion, because of not only the complexity but the calculation time involved in differentiating the NURBS curve. In the case where Taylor's expansion is used for the NURBS interpolation, only the first or second order approximation was used in all of the references. The proposed recursive method is very simple and the calculation time is also very short. Only the NURBS equation is successively used to update the incremental parameter, Δu_j and the error becomes smaller as the recursive order n adopted becomes higher.

5. Conclusion

The recursive method of NURBS interpolation is very simple and fast, because the proposed algorithm does not use Taylor's expansion, but the recursive equation of the NURBS formula to determine the incremental parameter for the interpolation. Taylor's expansion for NURBS interpolation that has been used by most researchers is very complicated and gives an unavoidable truncation error. However, the proposed recursive algorithm with very high accuracy can replace the conventional Taylor's expansion for NURBS interpolation. A simulation study was conducted to demonstrate the advantages of the proposed interpolator compared with the method using Taylor's equation. It can be seen that this interpolator using the new concept of interpolation is sufficiently fast and precise for modern CNC systems. The recursive method of NURBS interpolation is much simpler, faster and more accurate than the conventional Taylor's expansion.

ACKNOWLEDGEMENT

This work was supported by the Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0020089).

REFERENCES

1. Mohan, S., Kweon, S. H., Lee, D. M., and Yang, S. H., "Parametric NURBS Curve Interpolators : A Review," Int. J. Precis. Eng. Manuf., Vol. 9, No. 2, pp. 84-92, 2008.
2. Piegl, L. and Tiller, W., "The NURBS Book, 2nd ed.," Springer, 1997.
3. Koren, Y., Lo, C. C., and Shpitalni, M., "CNC interpolators: algorithms and analysis," Manufacturing Science and Engineering, Vol. 64, pp. 83-92, 1993.
4. Yang, D. C. H. and Kong, T., "Parametric interpolator versus linear interpolator for precision CNC machining," Computer-Aided Design, Vol. 26, No. 3, pp. 225-234, 1994.
5. Zhang, Q. G. and Greenway, R. B., "Development and implementation of a NURBS curve motion interpolator," Robotics and Computer-Integrated Manufacturing, Vol. 14, No. 1, pp. 27-36, 1998.
6. Farouki, R. T. and Tsai, Y. F., "Exact Taylor series coefficients for variable-feedrate CNC curve interpolators," Computer-Aided Design, Vol. 33, No. 2, pp. 155-165, 2001.
7. Yeh, S. S. and Hsu, P. L., "The speed-controlled interpolator for machining parametric curves," Computer-Aided Design, Vol. 31, No. 5, pp. 349-357, 1999.
8. Tikhon, M., Ko, T. J., Lee, S. H., and Kim, H. S., "NURBS interpolator for constant material removal rate in open NC machine tools," International Journal of Machine Tools & Manufacture, Vol. 44, No. 2-3, pp. 237-245, 2004.
9. Yong, T. and Narayanaswami, R., "A parametric interpolator with confined chord errors, acceleration and deceleration for NC machining," Computer-Aided Design, Vol. 35, No. 13, pp. 1249-1259, 2003.
10. Du, D., Liu, Y., Yan, C., and Li, C., "An accurate adaptive parametric curve interpolator for NURBS curve interpolation," The International Journal of Advanced Manufacturing Technology, Vol. 32, No. 9-10, pp. 999-1008, 2007.
11. Xu, R. Z., Xie, L., Li, C. X., and Du, D. S., "Adaptive parametric interpolation scheme with limited acceleration and jerk for NC machining," The International Journal of Advanced Manufacturing Technology, Vol. 36, No. 3-4, pp. 343-354, 2008.
12. Lai, J. Y., Lin, K. Y., Tseng, S. J., and Ueng, W. D., "On the development of a parametric interpolator with confined chord error, feedrate, acceleration and jerk," The International Journal of Advanced Manufacturing Technology, Vol. 37, No. 1-2, pp. 104-121, 2008.
13. Seker, M., Narayanan, V. N., and Yang, S. H., "Design of jerk bounded feedrate with ripple effect for adaptive NURBS interpolator," The International Journal of Advanced Manufacturing Technology, Vol. 37, No. 5-6, pp. 545-552, 2008.
14. Lin, M. T., Tsai, M. S., and Yau, H. T., "Development of a dynamic-based NURBS interpolator with real-time look ahead algorithm," International Journal of Machine Tools & Manufacture, Vol. 47, No. 15, pp. 2246-2262, 2007.
15. Liu, X., Ahmad, F., Yamazaki, K., and Mori, M., "Adaptive interpolation scheme for NURBS curves with the integration of machining dynamics," International Journal of Machine Tools & Manufacture, Vol. 45, No. 4-5, pp. 433-444, 2005.
16. Du, D., Liu, Y., Guo, X., Yamazaki, K., and Fujishinma, M., "An accurate adaptive NURBS curve interpolator with real-time flexible acceleration/deceleration control," Robotics and Computer-Integrated Manufacturing, Vol. 26, No. 4, pp. 273-281, 2010.
17. Cheng, M. Y., Su, K. H., and Wang, S. F., "Contour error reduction for free-form contour following task of biaxial motion control systems," Robotics and Computer-Integrated Manufacturing, Vol. 25, No. 2, pp. 323-333, 2009.
18. Baek, D. K., Yang, S. H., and Ko, T. J., "Precision NURBS interpolator based on recursive characteristics of NURBS," The International Journal of Advanced Manufacturing Technology, DOI: 10.1007/s00170-012-4179-z, 2012.