Int J Adv Manuf Technol (2006) 30: 98–104
DOI 10.1007/s00170-005-0021-1

ORIGINAL ARTICLE

Chung-Wei Cheng · Wen-Peng Tseng

Design and implementation of a real-time NURBS surface interpolator

Received: 9 September 2004 / Accepted: 5 March 2005 / Published online: 21 September 2005

Abstract In recent years, various parametric curve interpolators have been proposed for high-speed and high-accuracy machining. However, these methods are based on a constant cutter location (CL) velocity. Hence, the cutter contact (CC) velocity along the surface tends to vary, resulting in a nonuniform machining performance. Furthermore, machining complex surfaces requires the provision of a large number of tool paths and therefore the associated NC data files are generally very large. To overcome these limitations, the current study presents a novel real-time NURBS surface interpolator which ensures a constant CC velocity along the CC paths and its intervals. A PC-based real-time motion control network utilizing SSCNET is developed to achieve the goal of multi-axis synchronous motion. In this study, both the NURBS surface interpolator algorithms and the SSCNET communication protocols are realized by Embedded XP with the RTX real-time kernel. The experimental results confirm that the proposed real-time NURBS surface interpolator is capable of achieving a satisfactory performance.

Keywords Interpolator · Motion control network · NURBS surface

1 Introduction

Conventional CNC motion controllers provide only line or arc interpolators. As a result, when performing mold machining operations, the tool paths, which are also known as cutter location (CL) paths, are approximated with piecewise line or arc segments by the CAM system. However, this approximation approach causes several problems: (1) the NC data file tends to be rather large and therefore prolongs the data transmission and machining times; and (2) the cutter needs to accelerate/decelerate at the beginning and end of each line segment and this may introduce a velocity discontinuity at the junction of the two connected line segments.

To overcome these problems, various parametric curve interpolators have been proposed by a number of investigators [1–6]. Among these parametric curves, non-uniform rational B-spline (NURBS) has attracted particular interest [7] since it provides a common mathematical form for representing standard analytical shapes. By manipulating the values of the control points, weights, and knots, a wide variety of part forms can be generated. Accordingly, several investigators [8–12] have proposed the use of NURBS curve interpolators for advanced machining applications.

Although parametric or NURBS curve interpolators have been shown to be efficient, they may not be so for machining parametric surfaces. In other words, despite their ability to reduce the number of segments required in the tool path direction, surface machining still demands the provision of a sequence of tool paths (i.e., a composite set of parametric curves). Consequently, when a very fine segmentation in the direction of the tool path interval is required, the file size of the NC program tends to be very large. Previous parametric and NURBS curve interpolators have focused on maintaining a constant CL velocity and have allowed the cutter contact (CC) velocity to vary. However, a variable CC velocity frequently results in a non-uniform and unsatisfactory machining performance. In an attempt to solve these problems, Lo [13] proposed a novel approach, i.e., the parametric surface interpolator, capable of generating the CL motion commands required for the machining of simple parametric surfaces with a constant CC velocity along the CC path. More recently, Lin [14] realized a real-time parametric surface interpolator for the machining of simple surfaces. However, little work has been performed with respect to the real-time implementation of NURBS surface interpolators.

C.-W. Cheng (✉) · W.-P. Tseng
N600, Mechanical Industry Research Laboratories,
Industrial Technology Research Institute,
Hsin-Shi,
Tainan County, 741, Taiwan
e-mail: CWCheng@itri.org.tw
Tel.: +886-6-5089017
Fax: +886-6-5089056

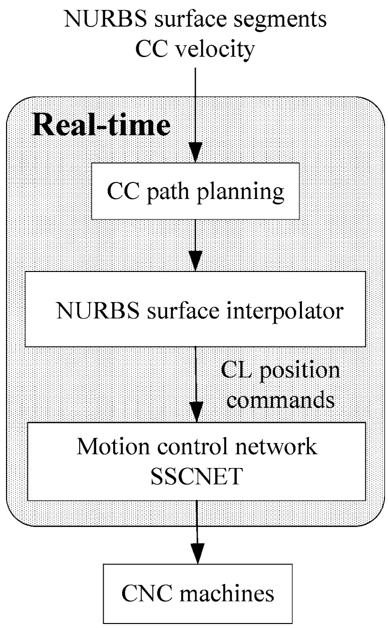


Fig. 1 Proposed method for machining NURBS surface segments with constant CC velocity

Consequently, this study develops an efficient algorithm for realizing a NURBS surface interpolator in real time. As shown in Fig. 1, the proposed method has the capability of: (1) real-time CC path planning of the NURBS surface; (2) real-time generation of the CL position commands; and (3) maintaining the desired CC velocity along the CC paths and the CC path intervals.

To achieve the goal of multi-axis synchronous motion, this study develops a PC-based real-time motion control network utilizing servo system control network (SSCNET), which is a real-time serial communication protocol proposed by Mitsubishi Electric. SSCNET prescribes communication protocols for the cyclic transmission of process data, provides a master-slave communication architecture for the synchronization control of networked motor drives, and supports the non-cyclic transmission of diagnostic data. Furthermore, to accomplish real-time performance, this study employs an Embedded XP operating system with the RTX 6.0 real-time kernel [15]. The effectiveness of the developed NURBS surface interpolator with the integrated SSCNET motion control network is verified through its application to two surface machining operations.

The remainder of this paper is organized as follows. Section 2 provides an introduction to the real-time CC path planning of a NURBS surface, while Sect. 3 presents the real-time NURBS surface interpolator and its computer implementation procedures. Section 4 presents the current experimental results. The paper closes with the presentation of some brief conclusions in Sect. 5.

2 Real-time CC path planning for NURBS surface

A parametric surface in 3-D space can be expressed as:

$$S(u,v) = x(u,v)\vec{i} + y(u,v)\vec{j} + z(u,v)\vec{k} \quad 0 \le u,v \le 1 \tag{1}$$

The corresponding NURBS surface representation is given by [7]:

$$S(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}P_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}} \tag{2}$$

where $P_{i,j}$ are the 3-D control points; $w_{i,j}$ are the corresponding weights of $P_{i,j}$; $N_{i,p}(u)$ $N_{j,q}(v)$ are the so-called blending functions, defined on the knot vectors U=[u_i,...,u_{i+p}]

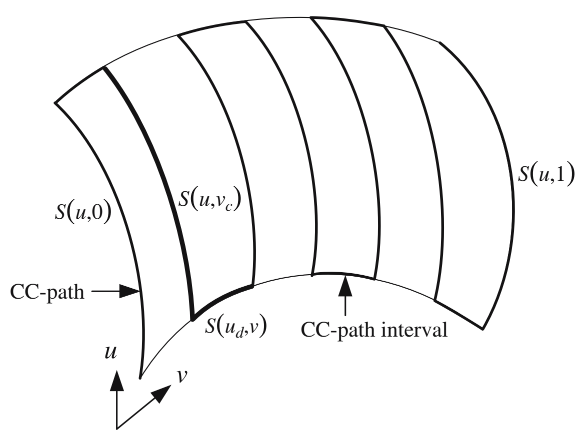


Fig. 2 CC path planning for machining parametric surface $S(u,v)$

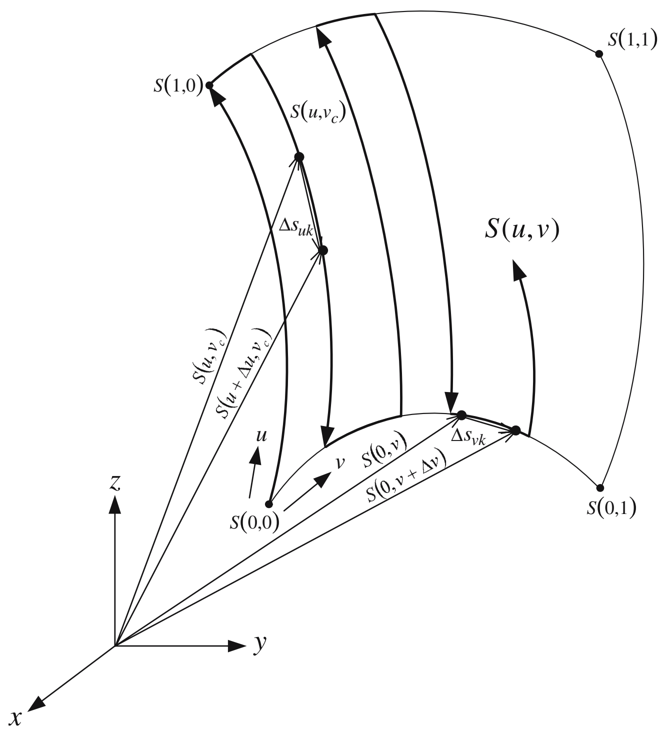


Fig. 3 Interpolation of CC path and CC path interval

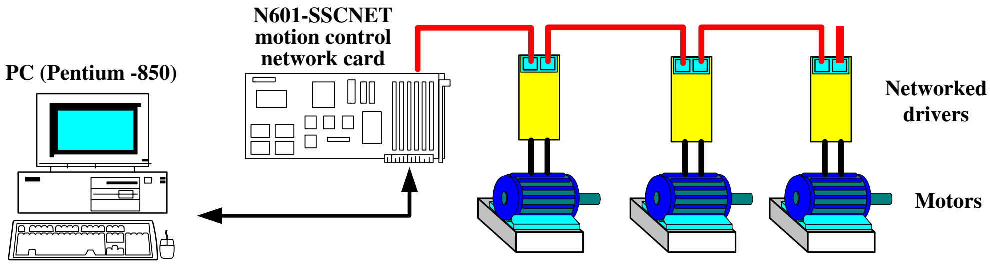


Fig. 4 Layout of experimental system

and V=[v_j,...,v_{j+q}], respectively; and p and q are the orders of blending functions $N_{i,p}(u)$ and $N_{j,q}(v)$, respectively.

Real-time CC path planning decomposes a NURBS surface into a group of NURBS curves (CC paths, see Fig. 2) along which the cutter moves during the machining operation. To generate the CC paths efficiently, thereby permitting real-time control, this study adopts the iso-parametric machining method [16] with the following strategies:

(1) The CC paths are assigned along the u-direction by fixing parameter v, i.e., the *cth* CC path $S(u,v_c)$ is defined (see Fig. 2).
(2) The CC path intervals are assigned along the v-direction by fixing parameter u, i.e., the *dth* CC path interval S(u_d,v) is defined (see Fig. 2). In general, u_d=0 or 1.
(3) The boundary curves, $S(u,0)$ and $S(u,1)$ (see Fig. 2), are specified as the first and final CC path, respectively.
(4) The present study considers the use of a ball-end cutter for 3-axis machining. Therefore, the CL position commands for the servo controller can be generated by offsetting the CC points along the normal direction of the surface by a distance equal to the cutter radius.

CC path planning requires the first-partial derivates of $S(u, v_c)$ versus u and of $S(u_d, v)$ versus v, i.e.,

$$S_u(u, v_c) = \frac{\partial S(u, v_c)}{\partial u} \tag{3}$$

$$S_v(u_d, v) = \frac{\partial S(u_d, v)}{\partial v} \tag{4}$$

For the NURBS surface interpolator, the second-partial derivates of $S(u, v_c)$ versus u and of $S(u_d, v)$ versus v are required, i.e.,

$$S_{uu}(u, v_c) = \frac{\partial^2 S(u, v_c)}{\partial u^2} \tag{5}$$

$$S_{vv}(u_d, v) = \frac{\partial^2 S(u_d, v)}{\partial v^2} \tag{6}$$

During interpolation of the *cth* CC path, $S(u, v_c)$, the increment $\Delta v_{c,k}$ between the current path and the following CC path at each interpolated point $S(u_k, v_c)$ can be calculated from [13]:

$$\Delta v_{c,k} = \frac{\Delta l}{(N \times T)S_v(u_k, v_c)} \tag{7}$$

where Δl is the incremental length of the CC path interval, and T and N are the unit tangent and unit normal vectors, respectively, at the interpolated point $S(u_k, v_c)$. At the end of the *cth* CC path, the minimum value of $\Delta v_{c,k}$ $\forall k$, denoted

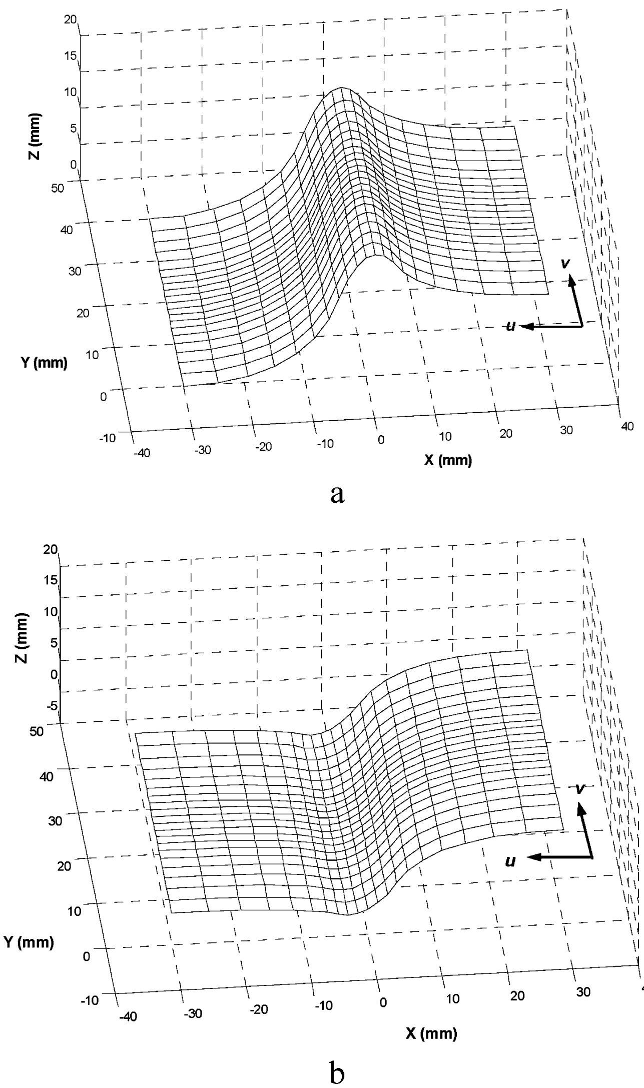


Fig. 5 Two NURBS surfaces for performance evaluation

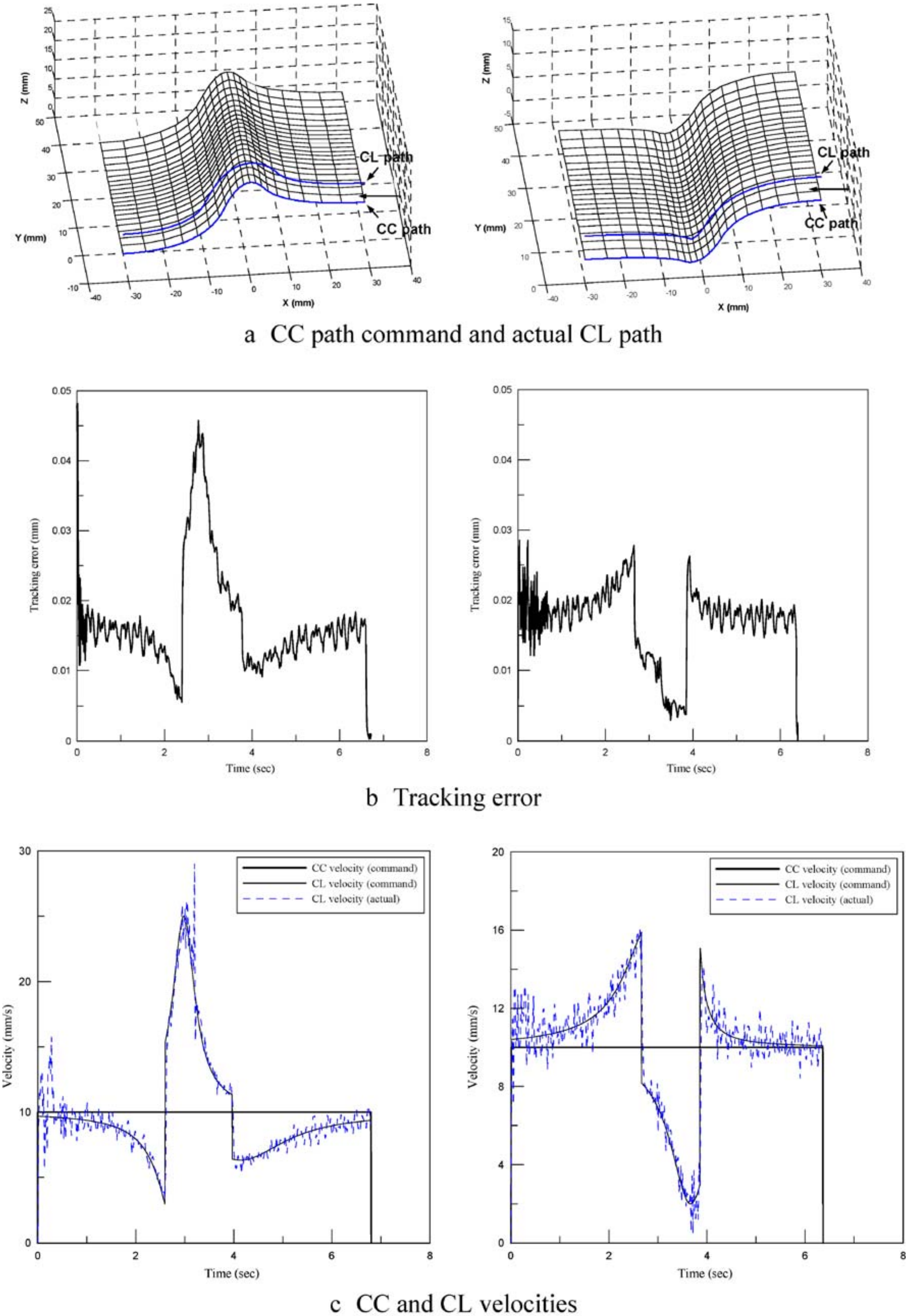
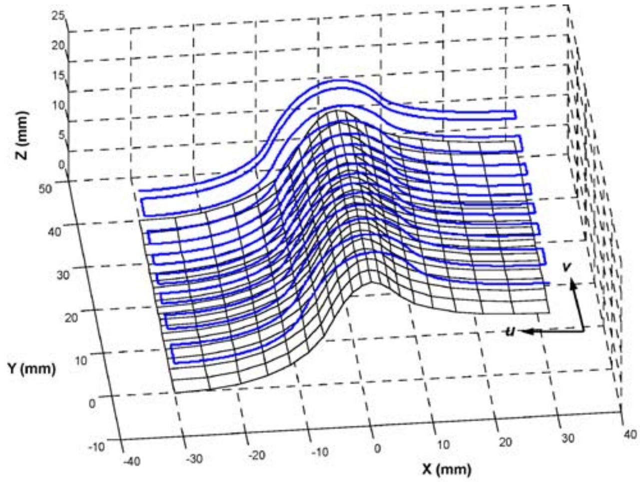
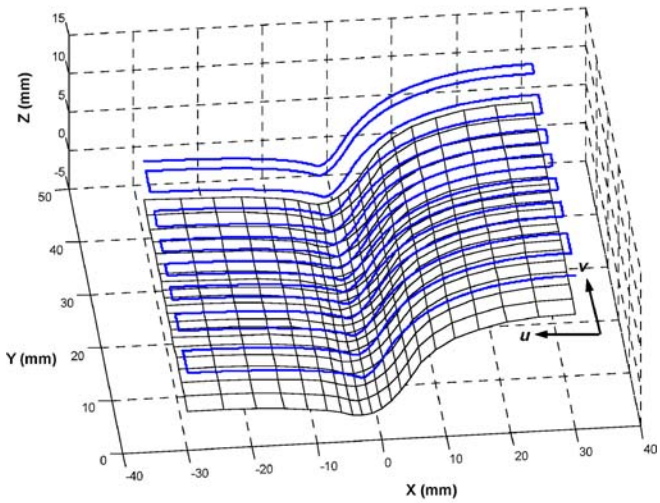


Fig. 6 Results of experiment #1



a



b

Fig. 7 Actual CL paths of experiment #2

by Δv_c, is obtained and used immediately as a parameter increment for the next CC-path.

The formulas for computing Δl, T, and N are shown in Eqs. 8, 9, and 10, respectively:

$$\Delta l = \sqrt{\frac{8hrR_v}{R_v \pm r}} \tag{8}$$

$$T(u_k, v_c) = \frac{S_u(u_k, v_c)}{|S_u(u_k, v_c)|} \tag{9}$$

$$N(u_k, v_c) = \frac{S_u(u_k, v_c) \times S_v(u_k, v_c)}{|S_u(u_k, v_c) \times S_v(u_k, v_c)|} \tag{10}$$

In Eq. 8, h is the allowable scallop height, r is the radius of the ball-end cutter, and R_v is the radius of curvature in the CC path interval direction (v-direction). Note that the nature of the ± sign depends on whether the NURBS surface in the CC path interval direction is convex (+) or concave (−). R_v is given by:

$$R_v = \frac{\left|S_v(u_i, v_j)\right|^3}{\left|S_v(u_i, v_j) \times S_{vv}(u_i, v_j)\right|} \tag{11}$$

3 Real-time NURBS surface interpolator

3.1 Developing the real-time NURBS surface interpolator

The purpose of the NURBS surface interpolator is to convert the NURBS surface segments into CL position commands for each axis in real time such that the motion of the individual axes are coordinated and the required constant CC velocity is achieved. As shown in Fig. 3, the key to achieving a constant CC velocity lies in specifying the incremental distances Δs_{uk} of the CC path in the u-direction and Δs_{vk} of the CC path interval in the v-direction in accordance with equal increments of the sampling period T, rather than on the basis of equal increments of Δu and Δv. Accordingly, at each sampling period, a method for determining successive values of u and v is required to ensure that appropriate values of Δs_{uk} and Δs_{vk} are accurately generated.

The constant CC velocity, V_{cc}, along the CC path, $S(u, v_c)$, in the u-direction is given by:

$$V_{cc} = \left\| \frac{dS(u, v_c)}{dt} \right\|. \tag{12}$$

Since v_c is fixed, the chain rule can be applied. Hence:

$$V_{cc} = \left\| \frac{dS(u, v_c)}{dt} \right\| = \left\| \frac{\partial S(u, v_c)}{\partial u} \frac{du}{dt} \right\| = \| S_u(u, v_c) \| \frac{du}{dt}. \tag{13}$$

Rearranging Eq. 13 gives:

$$\frac{du}{dt} = \frac{V_{cc}}{\| S_u(u, v_c) \|}. \tag{14}$$

Realizing a real-time interpolator for a NURBS surface requires a computationally efficient means of solving Eq. 14. Generally, it is difficult to obtain an exact solution, and hence approximation forms are often used. A previous study by the present authors [12] reported that Taylor's second-order approximation is an appropriate choice for implementing a real-time NURBS curve interpolator. Hence, the current study adopts the same approximation method to realize the current NURBS surface interpolator. The second-order approximation of du/dt is given by:

$$u_{k+1} \approx u_k + T \left. \frac{du}{dt} \right|_{t=tk} + \frac{T^2}{2} \left. \frac{d^2u}{dt^2} \right|_{t=tk} \tag{15}$$

where $u_k=u(t_k)$ denotes the value of u at the kth sampling time instant, i.e. $t_k=kT$.

The expression of d^2u/dt^2 can be derived from Eq. 14 as:

$$\frac{d^2u}{dt^2}=\frac{-V_{cc}^2[S_u(u,v_c)\bullet S_{uu}(u,v_c)]}{\|S_u(u,v_c)\|^4} \tag{16}$$

Substituting Eqs. 14 and 16 into Eq. 15 gives the Taylor's second-order NURBS surface interpolator for generating u_{k+1} as:

$$u_{k+1}=u_k+\frac{TV_{cc}}{\|S_u(u,v_c)\|_{u=uk}}-\frac{T^2V_{cc}^2}{2}\times\left(\frac{[S_u(u,v_c)\bullet S_{uu}(u,v_c)]_{u=uk}}{\|S_u(u,v_c)\|_{u=uk}^4}\right). \tag{17}$$

By discarding the higher order terms of Eq. 17, the simplified Taylor's first-order NURBS surface interpolator for generating u_{k+1} can be expressed as:

$$u_{k+1}=u_k+\frac{TV_{cc}}{\|S_u(u,v_c)\|_{u=uk}}. \tag{18}$$

By following a similar approach, it can be shown that the Taylor's second- and first-order NURBS surface interpolators for the CC path interval, $S(u_d,v)$, in the v-direction are given by Eqs. 19 and 20, respectively, as:

$$v_{k+1}=v_k+\frac{TV_{cc}}{\|S_v(u_d,v)\|_{v=v_k}}-\frac{T^2V_{cc}^2}{2}\times\left(\frac{[S_v(u_d,v)\bullet S_{vv}(u_d,v)]_{v=v_k}}{\|S_v(u_d,v)\|_{v=v_k}^4}\right) \tag{19}$$

$$v_{k+1}=v_k+\frac{TV_{cc}}{\|S_v(u_d,v)\|_{v=v_k}}. \tag{20}$$

Substituting the computed values of (u_{k+1}, v_c) or (u_d, v_{k+1}) into Eq. 2 yields the next 3-axis CC point at time t_{k+1}. However, in 3-axis CNC machining, the servo controller actually requires the next 3-axis CL position command rather than the next CC point. Since the present study considers the case of a ball-end cutter, the next 3-axis CL position command, $P_{CL}(x_{k+1}, y_{k+1}, z_{k+1})$, at time t_{k+1} is simply an offset of the CC point, $S(u_i, v_j)$, and can be derived from:

$$P_{CL}(x_{k+1},y_{k+1},z_{k+1})=S(u_i,v_j)+r\cdot N(u_i,v_j) \tag{21}$$

where r is the ball-end cutter radius and $N(u_i,v_j)$ is the unit normal vector defined in Eq. 10.

3.2 Real-time implementing procedures

The real-time procedures for implementing the proposed NURBS surface interpolator can be summarized as follows:

(1) Fix $v=v_c$ for the cth CC path (v_c=0 for the first path), and then for each CC point, $S(u_k, v_c)$, find the increment $\Delta v_{c,k}$ between the current and the next CC path using Eq. 7.
(2) Execute the Taylor's second-order NURBS surface interpolator to calculate the next u_{k+1} using Eq. 17. According to the computed value of (u_{k+1}, v_c), generate the next 3-axis CL position command, $P_{CL}(x_{k+1}, y_{k+1}, z_{k+1})$, using Eq. 21.
(3) Repeat steps (1) to (2) until u_{k+1}>1. If u_{k+1}>1, find the minimum value of $\Delta v_{c,k}\ \forall k$, denoted Δv_c, increase the counter c for the next CC path by one and update the variable v_c by $v_c=\sum_{i=1}^{c-1}\Delta v_i$. If v_c>1, let v_c=1, perform machining of the final CC path, and then terminate the machining task. Otherwise, proceed to the next step.
(4) Fix $u=u_d$ for the CC path interval. If the counter c is an odd number, u_d=0. Otherwise, u_d=1. Execute the Taylor's first-order (for simplicity) NURBS surface interpolator (for the CC path interval) to calculate v_{k+1} using Eq. 20.
(5) According to the computed value of (u_d, v_{k+1}) from step (4), generate the next 3-axis CL position command, $P_{CL}(x_{k+1}, y_{k+1}, z_{k+1})$, using Eq. 21.
(6) Repeat steps (4) to (5) until $v_{k+1}>v_c$. If $v_{k+1}>v_c$, return to step (3).

4 Experimental results

The layout of the experimental system used in the present study is illustrated in Fig. 4. A PC-based system with a motion control network card (N601-SSCNET developed by MIRL/ITRI) was used to realize multi-axis synchronous motion. The controlled plant consisted of three servomotors controlled by SSCNET networked drivers to emulate a three degree-of-freedom motion. An embedded XP operating system with the RTX 6.0 real-time kernel was used to implement the following algorithms in real time: (1) NURBS surface representation; (2) CC path planning; (3) NURBS surface interpolator; and (4) SSCNET communication protocol. The algorithms were all implemented using Visual C++ 6.0 and were executed on a personal computer equipped with a Pentium III 850 MHz CPU with a sampling time of 3.555 ms.

Figure 5 presents the two NURBS surfaces with different control points used for performance evaluation purposes in the current study. In both surfaces, the orders p and q are specified as $p=q=3$, the weights $w_{i,j}$ are set to 1, and the knot vectors are $U=V=[0, 0, 0, 1/3, 2/3, 1, 1, 1]$. For each NURBS surface, two experiments were performed to evaluate the real-time performance of the proposed NURBS surface interpolator: (1) machining of the first CC path, $S(u,0)$; and (2) machining of the entire NURBS surface. In both cases, the feedrate was specified as a constant 10 mm/s, the ball-end cutter radius was 5 mm and the allowable scallop height was 0.5 mm. Note that a large value of the scallop height was chosen specifically in order to more clearly observe the CL paths.

Figure 6 presents the experimental results obtained when machining the first CC path, $S(u, 0)$, with a constant CC velocity of 10 mm/s. Figure 6(a) confirms that the actual CL path tracks the planned CL path satisfactorily. This finding is reinforced from an observation of the corresponding tracking errors presented in Fig. 6(b). Additionally, Fig. 6(c) shows that the actual CL velocity closely follows the instructed CL velocity. Since the CC velocity can be maintained at a virtually constant value along the CC path, the machining efficiency and quality are improved significantly, i.e., because the cutting actually occurs around the CC point rather than around the CL point. Interestingly, it is observed that if a constant CC velocity is to be maintained along the NURBS surface, the CL velocity must decrease when the surface is concave and must increase when the surface is convex. Figure 7 presents the experimental results for machining of the complete NURBS surfaces. It is found that the CL paths for complete surface machining can be successfully generated.

5 Conclusions

This study has utilized real-time CC path planning and interpolation algorithms to develop a novel real-time NURBS surface interpolator capable of producing CL position commands with constant CC velocity for precision machining. Compared with existing methods, which involve a variable CC velocity, the constant CC velocity of the current interpolator provides a superior machining efficiency and quality. The experimental results obtained from the NURBS surface interpolator algorithms and the SSCNET communication protocols realized by Embedded XP with the RTX 6.0 real-time kernel fully confirm the effectiveness of the proposed method.

References

1. Shpitalni M, Koren Y, Lo CC (1994) Realtime curve interpolators. Comput Aided Des 26:832–838
2. Yang DCH, Kong T (1994) Parametric interpolator versus linear interpolator for precision CNC machining. Comput Aided Des 26:225–234
3. Wang FC, Wright RK (1998) Open architecture controllers for machine tools, part 2: a real time quintic spline interpolator. ASME J Manuf Sci Eng 120:425–432
4. Farouki RT, Tsai YF (2001) Exact Taylor series coefficients for variable-feedrate CNC curve interpolators. Comput Aided Des 33:155–165
5. Yeh SS, Hsu PL (2002) Adaptive-feedrate interpolation for parametric curves with a confined chord error. Comput Aided Des 34:229–237
6. Nam SH, Yang MY (2004) A study on a generalized parametric interpolator with real-time jerk-limited acceleration. Comput Aided Des 36:27–36
7. Piegl L, Tiller W (1997) The NURBS Book, 2nd edn. Springer, Berlin Heidelberg New York
8. Zhang Q, Greenway RB (1998) Development and implementation of a NURBS curve motion interpolator. Robot Comput Integr Manuf 14:27–36
9. Yang MY, Park JH (2002) A study on a open architecture CNC system with a NURBS interpolator for WEDM. Int J Adv Manuf Technol 19:664–668
10. Zhiming X, Jincheng C, Zhengjin F (2002) Performance evaluation of a real-time interpolation algorithm for NURBS curves. Int J Adv Manuf Technol 20:270–276
11. Cheng MY, Tsai MC, Kuo JC (2002) Real-time NURBS command generators for CNC servo controllers. Int J Mach Tools Manuf 42:801–813
12. Cheng CW, Tsai MC (2004) Real-time variable feedrate NURBS curve interpolator for CNC Machining. Int J Adv Manuf Technol 23:865–873
13. Lo CC (1998) A new approach to CNC tool path generation. Comput Aided Des 30:649–655
14. Lin RS (2000) Real-time surface interpolator for 3-D parametric surface machining on 3-axis machine tools. Int J Mach Tools Manuf 40:1513–1526
15. Cherepov M, Hirst M, Jones C, Zimmerman M (2002) Hard real-time with Venturcom RTX on Microsoft Windows XP/Xpe. Venturcom Inc., White Paper
16. Lin RS, Koren Y (1996) Efficient tool-path planning for machining free-form surfaces. ASME J Eng Ind 118:20–28