A Real-time Scheme for Evaluating NURBS Surfaces on NC Milling Machines

Syh-Shiuh Yeh

Dept. Mechanical Engineering, National Taipei University of Technology, Taipei 10608, Taiwan

ABSTRACT: This study aims to develop a scheme for the real-time evaluation of NURBS surfaces on NC milling machines with a limited interpolation time. NURBS surfaces are usually applied to NC milling machines because of the flexibility they offer in modeling processes. However, for obtaining high-quality machining results, the interpolation time is usually limited to a small value, and thus the copious and complicated operations associated with NURBS surfaces limit their applications in actual machining processes. To interpolate NURBS surfaces within a limited interpolation time, a real-time scheme for evaluating NURBS surfaces is proposed in this paper. Based on the formula of NURBS surfaces in extended space, an efficient computation structure is developed for efficiently computing NURBS surfaces and their partial derivatives within the limited interpolation time. Simulations and machining tests on a vertical machining center show that the proposed approach provides good evaluation results with short computation times and that it is feasible for the real-time interpolation of NURBS surfaces in actual machining processes.

Keywords: NURBS Surface, NC, Milling Machine, Evaluation, Interpolation

1. INTRODUCTION

In the current decade, a new data transmission standard named STEP-NC has been proposed for improving the performance of integrated CAD/CAM/CNC systems [1–4]. According to the STEP-NC standard, the surface information of a model is directly sent to the CNC machine without segmentation, and thus the interpolator on the CNC machine must provide functions for interpolating the surfaces in real-time. The NURBS surface is widely applied in modeling applications because it offers a common mathematical form for representing free-form surfaces. Moreover, the NURBS surface provides a flexible design scheme for free-form surfaces by manipulating some characteristic parameters such as control points, weights, and knot vectors. Therefore, it is important to design an algorithm for interpolating NURBS surfaces on CNC machines. Based on the improved iso-scallop strategy for regulating scallop height, Jee and Koo [5] proposed an efficient tool-path generation method for interpolating NURBS surfaces on CNC machines. Tsai et al. [6] proposed a NURBS surface interpolator design for generating cutter-location commands in real-time and for maintaining a constant cutter-contact velocity along a NURBS surface such that the machining efficiency and quality are significantly improved. However, the copious and complicated operations associated with NURBS surfaces usually limit the execution performance of proposed interpolation algorithms in actual machining applications.

* Corresponding Author: ssyeh@ntut.edu

In this paper, a real-time scheme is proposed to reduce the computation time required for evaluating NURBS surfaces on NC milling machines. To efficiently compute NURBS surfaces and their partial derivatives, an efficient computation structure is proposed by considering NURBS surfaces in extended space. Because the formula for NURBS surfaces is modified as a non-rational function, the proposed computation structure provides an efficient method for computing NURBS surfaces on NC machines. Since tool-radius and length compensations are usually required in machining processes, interpolators must also evaluate the partial derivatives of NURBS surfaces within a limited interpolation time. By analyzing the branch structure of the basis functions, we find that because of the inherent properties of iteration structures and redundant computations, the computation of the basis functions and their derivatives may significantly affect the computation time required for evaluating the partial derivatives of NURBS surfaces. In this paper, following the recent report by Yeh and Sun [7] of an efficient method for evaluating NURBS basis functions and their derivatives on NC machines, the proposed computation structure is incorporated into the evaluation method proposed by Yeh and Sun [7] for efficiently evaluating NURBS surfaces and their derivatives on NC milling machines. Finally, simulation and machining test results are provided to test the performance of the proposed approach. The simulation results indicate that the proposed scheme can provide good evaluation results, and the machining tests performed on a vertical machining center show that the proposed scheme is feasible for implementing real-time NURBS surface interpolators on NC milling machines.

2. REVIEW OF NURBS SURFACES

The formula for NURBS surfaces is given by [8]

$$S(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}P_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}} = \sum_{i=0}^{n}\sum_{j=0}^{m} R_{i,j}(u,v)P_{i,j} \quad (1)$$

and

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

where

- i and j are the index-of-sum along the u and v directions, respectively;
- n is the upper index-of-sum and (n+1) denotes the number of control points along the u direction;
- m is the upper index-of-sum and (m+1) denotes the number of control points along the v direction;
- $P_{i,j}$ are control points for shaping the NURBS surfaces;
- $w_{i,j}$ are the corresponding weights of control points $P_{i,j}$;
- $N_{i,p}(u)$ are the basis functions of a NURBS surface along the u direction, and p is the degree of $N_{i,p}(u)$;
- $N_{j,q}(v)$ are basis functions of a NURBS surface along the v direction, and q is the degree of $N_{j,q}(v)$;
- $R_{i,j}(u, v)$ is the rational basis function of a NURBS surface.

The recurrence formulas for computing basis functions $N_{i,p}(u)$ and $N_{j,q}(v)$ are shown in Eq. (3) and Eq. (4), respectively.

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

$$N_{i,0}(u) = \begin{cases} 1 & if\ u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$N_{j,q}(v) = \frac{v-v_j}{v_{j+q}-v_j}N_{j,q-1}(v) + \frac{v_{j+q+1}-v}{v_{j+q+1}-v_{j+1}}N_{j+1,q-1}(v)$$

$$N_{j,0}(v) = \begin{cases} 1 & \text{if } v_j \le v < v_{j+1} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where u_i and v_i are the knots on a NURBS surface along the u and v directions, respectively. The knot vectors U and V, formed by serial knots, are given by Eqs. (5) and (6).

$$U = \left[u_0 \quad u_1 \quad ..... \quad u_{n+p+1} \right]^T \quad (5)$$

$$V = \left[v_0 \quad v_1 \quad ..... \quad v_{m+q+1} \right]^T \quad (6)$$

The partial derivatives of NURBS $S(u,v)$ surfaces are given by Eqs. (7) and (8) [8].

$$S_u(u,v) = \frac{\partial S(u,v)}{\partial u} = \sum_{i=0}^{n}\sum_{j=0}^{m} \frac{\partial R_{i,j}(u,v)}{\partial u}P_{i,j} \quad (7)$$

$$S_v(u,v) = \frac{\partial S(u,v)}{\partial v} = \sum_{i=0}^{n}\sum_{j=0}^{m} \frac{\partial R_{i,j}(u,v)}{\partial v}P_{i,j} \quad (8)$$

where

$$\frac{\partial R_{i,j}(u,v)}{\partial u} = \frac{N_{i,p}^{(1)}(u)N_{j,q}(v)w_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}} - \frac{\left(N_{i,p}(u)N_{j,q}(v)w_{i,j}\right)\cdot\left(\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}^{(1)}(u)N_{j,q}(v)w_{i,j}\right)}{\left(\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}\right)^2} \quad (9)$$

$$\frac{\partial R_{i,j}(u,v)}{\partial v} = \frac{N_{i,p}(u)N_{j,q}^{(1)}(v)w_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}} - \frac{\left(N_{i,p}(u)N_{j,q}(v)w_{i,j}\right)\cdot\left(\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}^{(1)}(v)w_{i,j}\right)}{\left(\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}\right)^2} \quad (10)$$

Eqs. (11) and (12) represent the first derivatives of the basis functions, $N_{i,p}^{(1)}(u)$ and , $N_{j,q}^{(1)}(v)$ respectively.

$$N_{i,p}^{(1)}(u) = p\cdot\left(\frac{N_{i,p-1}(u)}{u_{i+p}-u_i} - \frac{N_{i+1,p-1}(u)}{u_{i+p+1}-u_{i+1}}\right) \quad (11)$$

$$N_{j,q}^{(1)}(v) = q\cdot\left(\frac{N_{j,q-1}(v)}{v_{j+q}-v_j} - \frac{N_{j+1,q-1}(v)}{v_{j+q+1}-v_{j+1}}\right) \quad (12)$$

NURBS surfaces $S(u, v)$ and their partial derivatives $S_u(u, v)$ and $S_v(u, v)$ can be obtained by applying Eqs. (1)–(12). However, for reducing the evaluation time, two additional topics must be considered:

- the computation structure of the NURBS surfaces, as shown in Eqs. (1)–(2) and Eqs. (7)–(10);
- the method of computing basis functions $N_{i,p}(u)$ and $N_{j,q}(v)$ and their derivatives $N_{i,p}^{(1)}(u)$ and $N_{j,q}^{(1)}(v)$.

In this paper, an efficient computation structure is developed for efficiently evaluating NURBS surfaces and their partial derivatives by reconsidering the formula for NURBS surfaces in extended space. The efficient method proposed by Yeh and Sun [7] is used to shorten the time required to evaluate the basis functions and their derivatives.

3. AN EFFICIENT COMPUTATION STRUCTURE FOR NURBS SURFACES

Referring to the formula for NURBS surfaces, Eq. (1) may be rewritten as:

$$W(u,v)\cdot S(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}P_{i,j} \tag{13}$$

where the function $W(u, v)$ is defined by Eq. (14).

$$W(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j} \tag{14}$$

By representing Eqs. (13) and (14) together, new surface functions $r(u, v)$ and new control points $r_{i,j}$ are derived in Eqs. (15) and (16).

$$\begin{aligned} r(u,v) &= \begin{bmatrix} W(u,v)\cdot S(u,v) \\ W(u,v) \end{bmatrix} = \\ &= \begin{bmatrix} \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}P_{i,j} \\ \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j} \end{bmatrix} \\ &= \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)\begin{bmatrix} w_{i,j}P_{i,j} \\ w_{i,j} \end{bmatrix} \end{aligned} \tag{15}$$

$$r_{i,j} = \begin{bmatrix} w_{i,j}P_{i,j} \\ w_{i,j} \end{bmatrix} \tag{16}$$

Equation (17), obtained from Eq. (15), denotes the formula of NURBS surfaces in extended space [8].

$$r(u,v) = \begin{bmatrix} W(u,v)\cdot S(u,v) \\ W(u,v) \end{bmatrix} = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)r_{i,j} \tag{17}$$

Based on Eq. (17), the partial derivatives of surface functions $r(u, v)$ are derived as Eqs. (18) and (19).

$$\begin{aligned} \frac{\partial r(u,v)}{\partial u} &= \begin{bmatrix} W(u,v)\frac{\partial S(u,v)}{\partial u} + \frac{\partial W(u,v)}{\partial u}S(u,v) \\ \frac{\partial W(u,v)}{\partial u} \end{bmatrix} \\ &= \sum_{i=0}^{n}\sum_{j=0}^{m} \frac{dN_{i,p}(u)}{du} N_{j,q}(v) r_{i,j} \end{aligned} \tag{18}$$

$$\begin{aligned} \frac{\partial r(u,v)}{\partial v} &= \begin{bmatrix} W(u,v)\frac{\partial S(u,v)}{\partial v} + \frac{\partial W(u,v)}{\partial v}S(u,v) \\ \frac{\partial W(u,v)}{\partial v} \end{bmatrix} \\ &= \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)\frac{dN_{j,q}(v)}{dv} r_{i,j} \end{aligned} \tag{19}$$

Eqs. (17)–(19) describe the structure for computing NURBS surfaces $S(u, v)$ and their partial derivatives $\frac{\partial S(u,v)}{\partial u}$ and $\frac{\partial S(u,v)}{\partial v}$. The proposed computation structure is summarized as follows:

1. Compute basis functions $N_{i,p}(u)$ and $N_{j,q}(v)$ and their derivatives, $\frac{dN_{i,p}(u)}{du}$ and $\frac{dN_{j,q}(v)}{dv}$.
2. Compute functions $r(u,v)$, $\frac{\partial r(u,v)}{\partial u}$, and $\frac{\partial r(u,v)}{\partial v}$ by

$$r(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)\cdot N_{j,q}(v)\cdot\begin{bmatrix} w_{i,j}P_{i,j} \\ w_{i,j} \end{bmatrix} \equiv \begin{bmatrix} r(u,v)_{11} \\ r(u,v)_{21} \end{bmatrix}$$

$$\frac{\partial r(u,v)}{\partial u} = \sum_{i=0}^{n}\sum_{j=0}^{m} \frac{dN_{i,p}(u)}{du}\cdot N_{j,q}(v)\cdot\begin{bmatrix} w_{i,j}P_{i,j} \\ w_{i,j} \end{bmatrix} \equiv \begin{bmatrix} \left(\frac{\partial r(u,v)}{\partial u}\right)_{11} \\ \left(\frac{\partial r(u,v)}{\partial u}\right)_{21} \end{bmatrix}$$

$$\frac{\partial r(u,v)}{\partial v} = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)\cdot \frac{dN_{j,q}(v)}{dv}\cdot\begin{bmatrix} w_{i,j}P_{i,j} \\ w_{i,j} \end{bmatrix} \equiv \begin{bmatrix} \left(\frac{\partial r(u,v)}{\partial v}\right)_{11} \\ \left(\frac{\partial r(u,v)}{\partial v}\right)_{21} \end{bmatrix}$$

3. Compute NURBS surfaces $S(u, v)$ and their partial derivatives, $\frac{\partial S(u,v)}{\partial u}$ and $\frac{\partial S(u,v)}{\partial v}$, by

$$S(u,v) = \frac{r(u,v)_{11}}{r(u,v)_{21}}$$

$$\frac{\partial S(u,v)}{\partial u} = \frac{\left(\frac{\partial r(u,v)}{\partial u}\right)_{11} - \left(\frac{\partial r(u,v)}{\partial u}\right)_{21}\cdot S(u,v)}{r(u,v)_{21}}$$

$$\frac{\partial S(u,v)}{\partial v} = \frac{\left(\frac{\partial r(u,v)}{\partial v}\right)_{11} - \left(\frac{\partial r(u,v)}{\partial v}\right)_{21}\cdot S(u,v)}{r(u,v)_{21}}$$

Clearly, compared with the existing approach shown in Eqs. (1)–(2) and Eqs. (7)–(10), the proposed computation structure (Eqs. (17)–(19)) can provide an efficient method for computing NURBS surfaces and their partial derivatives.

4. SIMULATIONS AND MACHINING TESTS

A personal computer running the MS-DOS operating system on a Pentium 200 MHz CPU is employed to verify the computation efficiency of the different approaches. The recorded evaluation time is the average value of the time for computing 10,000 NURBS points. Figure 1 shows the evaluation time compared with that of the conventional approach shown in Eqs. (1)–(10). Clearly, the conventional approach requires more time than the proposed approach. Figure 1 also shows that raising the degree of a NURBS surface substantially increases the evaluation time required by the conventional approach. However, by applying the proposed approach, raising the degree increases the evaluation time only slightly. Therefore, the proposed approach is more suitable for evaluating NURBS surfaces of higher degrees. Also, the simulation results indicate that the proposed approach may be feasibly implemented on real-time NURBS surface interpolators on NC milling machines.

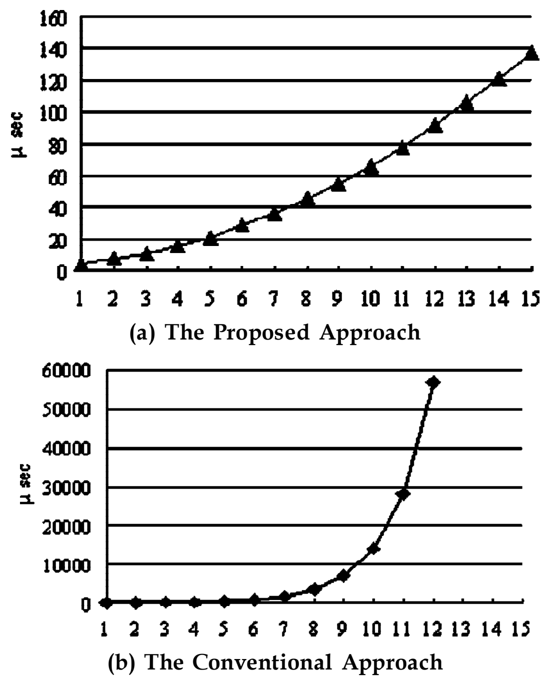

Figure 1: The Evaluation Time using Different Approaches

The proposed approach is applied to a LEADWELL MCV-OP CNC machining center, which is composed of an industrial PC, a DSP-based motion control card, and a mechanical system with three 3-phase Panasonic AC servo packs. Figure 2 shows the applied LEADWELL MCV-OP CNC machining center. An industrial PC with a Pentium III CPU is used to provide functions including the interface for human and machine operations, the interpreter for interpreting NC codes, the motion generator for generating motion commands, and the central processor for handling machining procedures. The DSP-based motion control card, which has a high-performance TI TMS320C32 digital signal processor (DSP), is applied to interpolate the motion commands and to control the three AC servo packs at a sampling period of 1 ms.

Figure 2: LEADWELL MCV-OP CNC Machining Center

A NURBS surface is applied to test the proposed approach. The upper index-of-sum along the u direction is set as $n = 9$, and the upper index-of-sum along the v direction is set as $m = 9$. The control points for shaping the applied NURBS surface are arranged as shown in Fig. 3. The weights of the control points are set to unity. The degrees, p and q, are set as $p = 2$ and $q = 2$. The knot vectors along the u and v directions are designed as

$$U = [0.0 \quad 0.0 \quad 0.0 \quad 0.125 \quad 0.25 \quad 0.375 \quad 0.5 \quad 0.625 \quad 0.75 \quad 0.875 \quad 1.0 \quad 1.0 \quad 1.0]^T$$

$$V = [0.0 \quad 0.0 \quad 0.0 \quad 0.125 \quad 0.25 \quad 0.375 \quad 0.5 \quad 0.625 \quad 0.75 \quad 0.875 \quad 1.0 \quad 1.0 \quad 1.0]^T$$

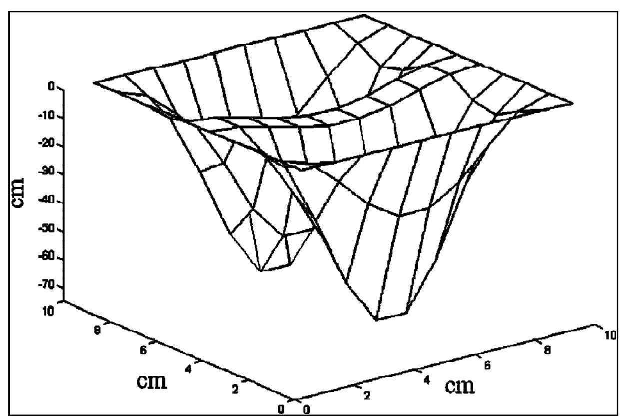

Figure 3: Control Points of the Applied NURBS Surface

Figure 4 shows the applied method for tool-radius and length compensations. As shown in Fig. 4, "a" denotes the cutting point and "b" denotes the reference point. The two variables u and v are the parameters of the parametric surfaces along different directions. R and H are the radius and length of cutting tool, respectively. The vector denotes the position vector of the reference point "b" in the workspace.

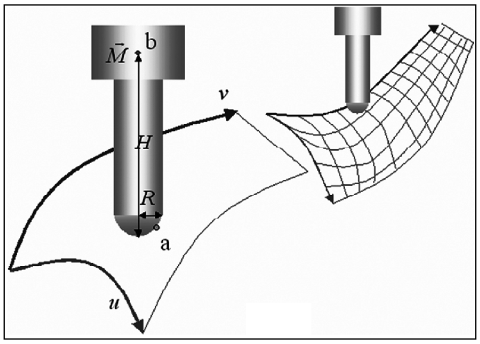

Figure 4: Tool Radius and Length Compensations of Parametric Surfaces

To avoid undercutting, the radial direction of the ball-nose endmill is maintained along the normal to the tangential plane constructed by the partial derivatives $S_u(u, v)$ and $S_v(u, v)$ at cutting point "a." Therefore, the vector $\vec{M}$ is derived as

$$\vec{M} = \vec{P}_a + R \cdot \vec{n}_a + (H - R) \cdot \vec{z}$$

where $\vec{P}_a$ is the position vector of contact point "a" in the workspace; $\vec{n}_a$, the normalized normal vector of the tangential plane; and $\vec{z}$, the unity vector along the z-axis. The normalized normal vector $\vec{n}_a$ is derived as

$$\vec{n}_a = \frac{\left(\vec{t}_{au} \times \vec{t}_{av}\right)}{\left\|\vec{t}_{au} \times \vec{t}_{av}\right\|}$$

where the tangential vectors $\vec{t}_{au}$ and $\vec{t}_{av}$ are obtained from the partial derivatives $S_u(u, v)$ and $S_v(u, v)$, respectively.

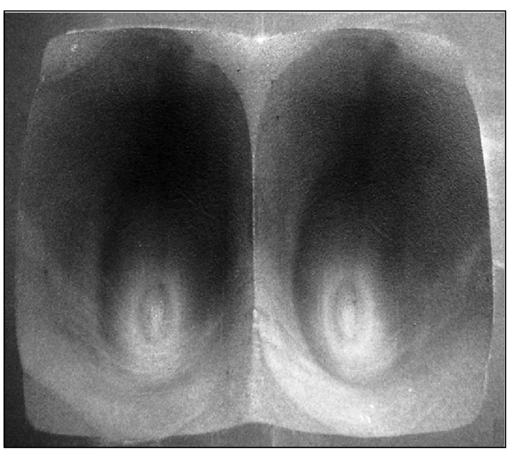

Figure 5: NURBS Surface Machining Results

Figure 5 shows the machining results obtained by using the proposed approach. The spindle speed is 4,485 RPM and the feedrate is 2,400 $^{mm}/_{min}$. The applied ball-nose endmill has a diameter of 6 mm and a length of 100 mm. Before the actual machining, the machining simulation results obtained by applying the proposed approach indicate that the proposed approach almost generates the same surface as that generated by the conventional approach. The maximum error is 0.0351 μm, which is smaller than the least unit (1 μm) in conventional NC machines. Moreover, Fig. 5 indicates that the proposed approach may be feasibly implemented on NURBS surface interpolators in NC milling machines.

5. CONCLUSIONS

In this paper, an efficient NURBS surface computation structure is proposed to reduce the computation time required for evaluating NURBS surfaces on NC milling machines. The computation structure considering the formula in extended space provides an efficient method for computing NURBS surfaces and their partial derivatives. Some simulation and machining results indicate that the proposed approach provides good computation efficiency and good evaluation results in the real-time evaluation of NURBS surfaces on NC milling machines.

ACKNOWLEDGEMENT

The author would like to thank Mr. Jin-Chu Sun and Mr. Mao-Feng Tu from the Mechanical and Systems Laboratories of Industrial Technology Research Institute for their contributions in preparing the experimental setup. Comments and suggestions from Mr. Jui-Kuan Lin are greatly appreciated.

REFERENCES

[1] X.W. Xu, "Realization of STEP-NC Enabled Machining", *Robotics and Computer-Integrated Manufacturing,* **22**, 2006, 144–153.

[2] K.H. Ho, S.T. Newman, R.D. Allen, "STEP-NC Compliant Information Modelling for Wire Electrical Discharge Machining Component Manufacture", *Proc. Instn. Mech. Engrs, Part B, Journal of Engineering Manufacture,* **219**, 2005, 777–784.

[3] W. Lee, Y.B. Bang, "Design and Implementation of an ISO14649-compliant CNC Milling Machine", *International Journal of Production Research,* **41**, 2003, 3007–3017.

[4] S.H. Suh, D.H. Chung, B.E. Lee, J.H. Cho, S.U. Cheon, H.D. Hong, H.S. Lee, "Developing an Integrated STEP-compliant CNC Prototype", *Journal of Manufacturing Systems*, **21**, 2002, 350–362.

[5] S. Jee, T. Koo, "Tool-path Generation for NURBS Surface Machining", *Proceedings of the American Control Conference*, **3**, 2003, 2614–2619.

[6] M.C. Tsai, C.W. Cheng, M.Y. Cheng, "A Real-time NURBS Surface Interpolator for Precision Three-axis CNC Machining", *International Journal of Machine Tools and Manufacture*, **43**, 2003, 1217–1227.

[7] S.S. Yeh, J.T. Sun, "A Scheme for Interpolating NURBS Curves on NC Machines", *Computer-Aided Design and Applications*, **4**, 2007, 31–40.

[8] L.A. Piegl, W. Tiller, *The NURBS Book*, Springer, New York, 1997.