A Scheme for Interpolating NURBS Curves on NC Machines

Syh-Shiuh Yeh[1] and Jin-Tsu Sun[2]

[1]National Taipei University of Technology, ssyeh7@hotmail.com
[2]Industrial Technology Research Institute, ricky@itri.org.tw

ABSTRACT

This study aims to develop a scheme for rapidly interpolating non-uniform rational B-spline (NURBS) curves on CNC machines with limited interpolation time. NURBS curves are usually applied to represent free-form curves because of the flexibility in modeling processes. However, since interpolation time is usually limited for obtaining high-quality machining results, the copious and complicated operations limit NURBS curves in machining applications. In this paper, a fast scheme with a new computation structure, an extended basis function, and a moving data method is proposed for rapidly interpolating NURBS curves in limited interpolation time. Simulations and machining tests are given for testing the performance of the proposed scheme. Comparing with the existing approaches, the proposed approach provides good interpolation results with little interpolation time in actual machining processes.

Keywords: NURBS curve, interpolation, CNC machine.

1. INTRODUCTION

In modern CAD/CAM systems, curves for parts such as dies, vanes, aircraft models, and car models are usually represented in parametric forms. However, since conventional CNC machines only provide linear and circular arc interpolators, the performance of conventional CAD/CAM/CNC systems is usually limited in applications. Recently, the CAD/CAM/CNC systems shown in Fig. 1 are widely applied to manufacturing processes for improving machining results. However, a segmentation unit is required to generate parametric curve information for CNC machines, and the interpolator in CNC machines requires a function for interpolating parametric curves.

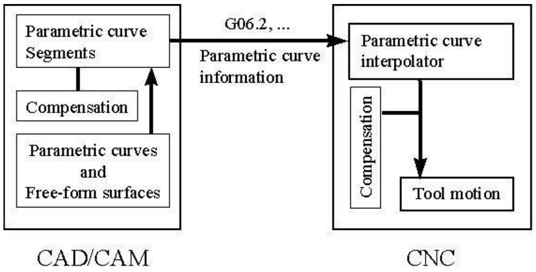
Parametric curve Segments
Compensation
Parametric curves and Free-form surfaces
G06.2, ...
Parametric curve information
Parametric curve interpolator
Compensation
Tool motion
CAD/CAM
CNC

Fig. 1: The CAD/CAM/CNC systems with curve information transmission.

Some researchers have proposed different approaches for precisely approximating planar parametric curves in CAD/CAM systems [6][8-10][13], and some investigators have proposed different interpolation algorithms for precisely interpolating parametric curves on CNC machines [1-5][11][14-17]. The approximation [6][8-10][13] in CAD/CAM systems is usually implemented on off-line computation systems and thus the computation time may not be a critical problem. However, for obtaining good machining results, the interpolation algorithms [1-5][11][14-17] must be implemented on CNC machines with limited interpolation time. Thus, the copious and complicated operations of NURBS curves usually limit the machining performances of interpolation algorithms [1-5][11][14-17] in actual machining applications.

In this paper, a fast scheme is proposed to reduce the computation time for interpolating NURBS curves. A new computation structure of NURBS curves, an extended branch structure of basis functions, and a moving data method with common data registers are proposed for implementing NURBS curve interpolators in real-time. Since the tool radius and length compensations are usually required in actual machining processes, interpolators must also interpolate the derivatives of NURBS curves in limited interpolation time. To efficiently compute NURBS curves and their derivatives, a new computation structure is proposed by considering NURBS curves in extended space. Moreover, an extended branch structure is developed for simultaneously computing basis functions and their derivatives. The basis function with extended branch structure used for simultaneously computing basis functions and their derivatives defines an extended basis function in this paper. Usually, interpolators spend much time for computing basis functions that have inherent properties of iteration structure and redundant computation. It is important to reduce the computation time of basis functions for interpolating NURBS curves. Cheng and Tsai [4] proposed an efficient computation method to compute basis functions and their derivatives in real-time. Because two adjacent basis functions require the same basis functions from sub-orders, only a single run of computation for the same order of basis functions is required and can be reused for higher-order calculation. The computation method proposed by Cheng and Tsai [4] considers the tree structure of a basis function with the same index-of-sum. In this paper, a new computation method considering the tree structures of basis functions with adjacent index-of-sum is proposed for further reducing the computation time of basis functions and their derivatives. By analyzing the tree structures of basis functions, a moving data method with common data registers is proposed for efficiently computing basis functions with adjacent index-of-sum. Finally, simulations and machining tests are given to test the performance of the proposed approach in this paper. Simulation results indicate that the proposed approach not only provides good interpolation results but also needs little interpolation time for interpolating NURBS curves. The machining tests performed on a vertical machining center show that the proposed scheme is feasible for interpolating NURBS curves in limited interpolation time on CNC machines.

2. REVIEW OF NURBS CURVES

The formulas of NURBS curves are given by [4][12]

$$P(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i P_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} = \sum_{i=0}^{n} R_{i,p}(u) P_i \tag{2.1}$$

and

$$R_{i,p}(u) = \frac{N_{i,p}(u) w_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \tag{2.2}$$

where i is the index-of-sum; n is the upper index-of-sum and $(n+1)$ denotes the number of control points; P_i are control points for shaping NURBS curves; w_i are the corresponding weights of control points P_i; $N_{i,p}(u)$ are basis functions of a NURBS curve and p is the degree of $N_{i,p}(u)$; $R_{i,p}(u)$ is the rational basis function of a NURBS curve.

The recurrence formulas for computing basis functions $N_{i,p}(u)$ are shown in Eqn. (2.3).

$$\begin{aligned} N_{i,p}(u) &= \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \\ N_{i,0}(u) &= \begin{cases} 1 & \text{if } u_i \le u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{2.3}$$

where u_i are knots of a NURBS curve. The knot vector U formed by serial knots are given by Eqn. (2.4).

$$U = \begin{bmatrix} u_0 & u_1 & \ldots\ldots & u_{n+p+1} \end{bmatrix}^T \tag{2.4}$$

Fig. 2 shows the branch structure of basis functions $N_{i,p}(u)$.

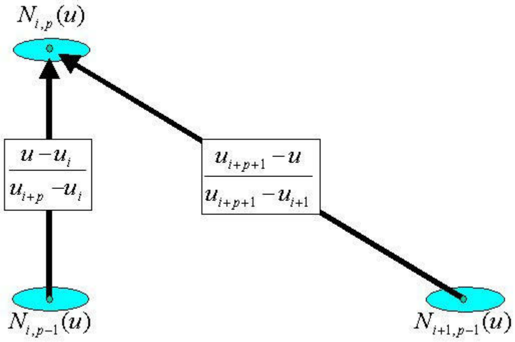
$N_{i,p}(u)$
$\frac{u-u_i}{u_{i+p}-u_i}$
$\frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}$
$N_{i,p-1}(u)$
$N_{i+1,p-1}(u)$

Fig. 2: The branch structure of basis functions $N_{i,p}(u)$.

The first and the second derivatives of NURBS curves are given by Eqns. (2.5)-(2.6) [4][12].

$$P^{(1)}(u)=\sum_{i=0}^{n} R_{i,p}^{(1)}(u)P_i \tag{2.5}$$

$$P^{(2)}(u)=\sum_{i=0}^{n} R_{i,p}^{(2)}(u)P_i \tag{2.6}$$

where

$$R_{i,p}^{(1)}(u)=\frac{N_{i,p}^{(1)}(u)w_i}{\sum_{i=0}^{n} N_{i,p}(u)w_i}-\frac{N_{i,p}(u)w_i\sum_{i=0}^{n} N_{i,p}^{(1)}(u)w_i}{\left(\sum_{i=0}^{n} N_{i,p}(u)w_i\right)^2} \tag{2.7}$$

$$R_{i,p}^{(2)}(u)=\frac{N_{i,p}^{(2)}(u)w_i}{\sum_{i=0}^{n} N_{i,p}(u)w_i}-\frac{2\cdot\left(N_{i,p}^{(1)}(u)w_i\sum_{i=0}^{n} N_{i,p}^{(1)}(u)w_i\right)+\left(N_{i,p}(u)w_i\sum_{i=0}^{n} N_{i,p}^{(2)}(u)w_i\right)}{\left(\sum_{i=0}^{n} N_{i,p}(u)w_i\right)^2}+\frac{2\cdot N_{i,p}(u)w_i\left(\sum_{i=0}^{n} N_{i,p}^{(1)}(u)w_i\right)^2}{\left(\sum_{i=0}^{n} N_{i,p}(u)w_i\right)^3} \tag{2.8}$$

The first and the second derivatives of basis functions, $N_{i,p}^{(1)}(u)$ and $N_{i,p}^{(2)}(u)$, are given by applying the k^{th} derivatives of basis functions as shown in Eqn. (2.9) [4][12].

$$N_{i,p}^{(k)}(u)=p\cdot\left(\frac{N_{i,p-1}^{(k-1)}(u)}{u_{i+p}-u_i}-\frac{N_{i+1,p-1}^{(k-1)}(u)}{u_{i+p+1}-u_{i+1}}\right) \tag{2.9}$$

Eqn. (2.10) and Eqn. (2.11) represent the first and the second derivatives of basis functions, respectively.

$$N_{i,p}^{(1)}(u)=p\cdot\left(\frac{N_{i,p-1}(u)}{u_{i+p}-u_i}-\frac{N_{i+1,p-1}(u)}{u_{i+p+1}-u_{i+1}}\right) \tag{2.10}$$

$$N_{j,q}^{(1)}(v)=q\cdot\left(\frac{N_{j,q-1}(v)}{v_{j+q}-v_j}-\frac{N_{j+1,q-1}(v)}{v_{j+q+1}-v_{j+1}}\right) \tag{2.11}$$

3. DESIGN OF A FAST SCHEME FOR INTERPOLATING NURBS CURVES

3.1 The Computation Structure of NURBS Curves

Referring to the formulas of NURBS curves, Eqn. (2.1) is rewritten as

$$W(u)\cdot P(u)=\sum_{i=0}^{n} N_{i,p}(u)w_iP_i \tag{3.1}$$

where the function $W(u)$ is defined by Eqn. (3.2).

$$W(u)=\sum_{i=0}^{n} N_{i,p}(u)w_i \tag{3.2}$$

By representing Eqn. (3.1) and Eqn. (3.2) together, new curve functions $r(u)$ and new control points r_i are derived by Eqns. (3.3)-(3.4).

$$r(u)=\begin{bmatrix} W(u)\cdot P(u) \\ W(u) \end{bmatrix}=\begin{bmatrix} \sum_{i=0}^{n} N_{i,p}(u)w_iP_i \\ \sum_{i=0}^{n} N_{i,p}(u)w_i \end{bmatrix}=\sum_{i=0}^{n} N_{i,p}(u)\begin{bmatrix} w_iP_i \\ w_i \end{bmatrix} \tag{3.3}$$

$$r_i=\begin{bmatrix} w_iP_i \\ w_i \end{bmatrix} \tag{3.4}$$

Eqn. (3.5) obtained from Eqns. (3.3)-(3.4) denotes the formula of NURBS curves in extended space.

$$r(u)=\begin{bmatrix} W(u)\cdot P(u) \\ W(u) \end{bmatrix}=\sum_{i=0}^{n} N_{i,p}(u)r_i \tag{3.5}$$

Based on Eqn. (3.5), the derivatives of curve functions $r(u)$ are derived by Eqns. (3.6)-(3.7).

$$\frac{dr(u)}{du}=\begin{bmatrix} W(u)\frac{dP(u)}{du}+\frac{dW(u)}{du}P(u) \\ \frac{dW(u)}{du} \end{bmatrix}=\sum_{i=0}^{n}\frac{dN_{i,p}(u)}{du}r_i \tag{3.6}$$

$$\frac{d^2r(u)}{du^2}=\begin{bmatrix} W(u)\frac{d^2P(u)}{du^2}+2\frac{dW(u)}{du}\frac{dP(u)}{du}+\frac{d^2W(u)}{du^2}P(u) \\ \frac{d^2W(u)}{du^2} \end{bmatrix}=\sum_{i=0}^{n}\frac{d^2N_{i,p}(u)}{du^2}r_i \tag{3.7}$$

Eqns. (3.5)-(3.7) describe the structure for computing NURBS curves $P(u)$ and their derivatives $P^{(1)}(u)$ and $P^{(2)}(u)$. The computation structure is summarized as follows:

1) Compute basis functions $N_{i,p}(u)$ and their derivatives, $\frac{dN_{i,p}(u)}{du}$ and $\frac{d^2N_{i,p}(u)}{du^2}$.

2) Compute functions $r(u)$, $\frac{dr(u)}{du}$, and $\frac{d^2r(u)}{du^2}$ by

$$r(u)=\sum_{i=0}^{n} N_{i,p}(u)\begin{bmatrix} w_iP_i \\ w_i \end{bmatrix}\equiv\begin{bmatrix} r(u)_{11} \\ r(u)_{21} \end{bmatrix};$$

$$\frac{dr(u)}{du}=\sum_{i=0}^{n}\frac{dN_{i,p}(u)}{du}\begin{bmatrix} w_iP_i \\ w_i \end{bmatrix}\equiv\begin{bmatrix} \left(\frac{dr(u)}{du}\right)_{11} \\ \left(\frac{dr(u)}{du}\right)_{21} \end{bmatrix};\ \frac{d^2r(u)}{du^2}=\sum_{i=0}^{n}\frac{d^2N_{i,p}(u)}{du^2}\begin{bmatrix} w_iP_i \\ w_i \end{bmatrix}\equiv\begin{bmatrix} \left(\frac{d^2r(u)}{du^2}\right)_{11} \\ \left(\frac{d^2r(u)}{du^2}\right)_{21} \end{bmatrix}.$$

3) Compute NURBS curves $P(u)$ and their derivatives $P^{(1)}(u)$ and $P^{(2)}(u)$ by

$$P(u)=\frac{r(u)_{11}}{r(u)_{21}};\ P^{(1)}(u)=\frac{dP(u)}{du}=\frac{\left(\frac{dr(u)}{du}\right)_{11}-\left(\frac{dr(u)}{du}\right)_{21}\cdot P(u)}{r(u)_{21}};$$

$$P^{(2)}(u)=\frac{d^2P(u)}{du^2}=\frac{\left(\frac{d^2r(u)}{du^2}\right)_{11}-\left(\frac{d^2r(u)}{du^2}\right)_{21}\cdot P(u)-2\cdot\left(\frac{dr(u)}{du}\right)_{21}\cdot\frac{dP(u)}{du}}{r(u)_{21}}.$$

Clearly, comparing with the existing approach as shown in Eqns. (2.1)-(2.2) and Eqns. (2.5)-(2.8), the computation structure shown in Eqns. (3.5)-(3.7) provides an efficient method for computing NURBS curves and their derivatives.

3.2 The Extended Branch Structure of Basis Functions

The derivatives of NURBS curves are obtained by applying the computation structure as shown in Eqns. (3.5)-(3.7). Clearly, the derivatives of basis functions are required for computing derivatives of NURBS curves. Based on the definition of basis functions, the first and the second derivatives of basis functions are derived in Eqn. (3.8) and Eqn. (3.9), respectively.

$$\frac{dN_{i,p}(u)}{du} = \frac{u-u_i}{u_{i+p}-u_i}\cdot\frac{dN_{i,p-1}(u)}{du} + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}\cdot\frac{dN_{i+1,p-1}(u)}{du} + \frac{1}{u_{i+p}-u_i}\cdot N_{i,p-1}(u) + \frac{-1}{u_{i+p+1}-u_{i+1}}\cdot N_{i+1,p-1}(u) \quad (3.8)$$

$$\begin{aligned}\frac{d^2N_{i,p}(u)}{du^2} &= \frac{u-u_i}{u_{i+p}-u_i}\cdot\frac{d^2N_{i,p-1}(u)}{du^2} + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}\cdot\frac{d^2N_{i+1,p-1}(u)}{du^2}\\ &+ \frac{2}{u_{i+p}-u_i}\cdot\frac{dN_{i,p-1}(u)}{du} + \frac{-2}{u_{i+p+1}-u_{i+1}}\cdot\frac{dN_{i+1,p-1}(u)}{du}\end{aligned} \quad (3.9)$$

For concisely representing the derivative functions, Eqn. (3.8) and Eqn. (3.9) are further rewritten in Eqn. (3.10) and Eqn. (3.11), respectively.

$$\frac{dN_{i,p}(u)}{du} = \alpha(i,p)\cdot\frac{dN_{i,p-1}(u)}{du} + \beta(i,p)\cdot\frac{dN_{i+1,p-1}(u)}{du} + \gamma(i,p)\cdot N_{i,p-1}(u) + \Omega(i,p)\cdot N_{i+1,p-1}(u) \quad (3.10)$$

$$\frac{d^2N_{i,p}(u)}{du^2} = \alpha(i,p)\cdot\frac{d^2N_{i,p-1}(u)}{du^2} + \beta(i,p)\cdot\frac{d^2N_{i+1,p-1}(u)}{du^2} + 2\cdot\gamma(i,p)\cdot\frac{dN_{i,p-1}(u)}{du} + 2\cdot\Omega(i,p)\cdot\frac{dN_{i+1,p-1}(u)}{du} \quad (3.11)$$

where rational functions $\alpha(i,p)$, $\beta(i,p)$, $\gamma(i,p)$, and $\Omega(i,p)$ are defined in Eqn. (3.12).

$$\alpha(i,p) = \frac{u-u_i}{u_{i+p}-u_i};\ \beta(i,p) = \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}};\ \gamma(i,p) = \frac{1}{u_{i+p}-u_i};\ \Omega(i,p) = \frac{-1}{u_{i+p+1}-u_{i+1}} \quad (3.12)$$

Comparing with Eqn. (2.3), the recurrence formulas for computing derivatives, $\frac{dN_{i,p}(u)}{du}$ and $\frac{d^2N_{i,p}(u)}{du^2}$, are similar to the recurrence formula for computing basis functions $N_{i,p}(u)$. Thus, the branch structure composed of Eqn. (2.3), Eqn. (3.8), and Eqn. (3.9) defines an extended branch structure in this paper. Accordingly, an extended basis function with the extended branch structure is applied to efficiently and simultaneously compute basis functions $N_{i,p}(u)$ and their derivatives, $\frac{dN_{i,p}(u)}{du}$ and $\frac{d^2N_{i,p}(u)}{du^2}$.

Fig. 3 shows the extended branch structure. Comparing with the existing branch structure as shown in Fig. 2, the extended branch structure simultaneously computes basis functions and their derivatives by using common rational functions $\alpha(i,p)$, $\beta(i,p)$, $\gamma(i,p)$ and $\Omega(i,p)$.

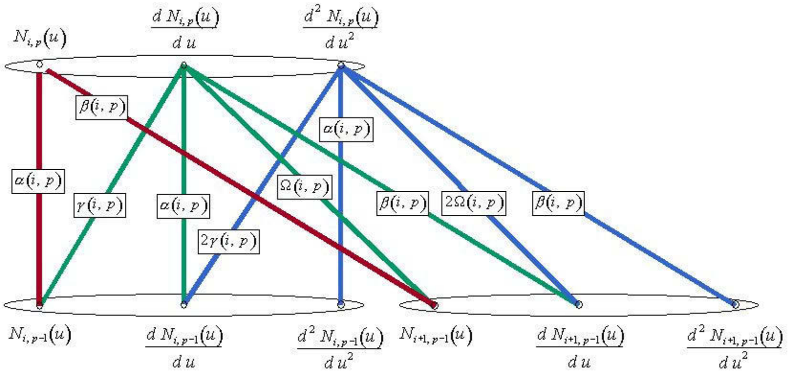

Fig. 3: The extended branch structure.

3.3 The Moving Data Method

The computation structure as shown in Eqns. (3.5)-(3.7) and the extended branch structure as shown in Fig. 3 are applied to efficiently compute NURBS curves and their derivatives. However, the tree structure of basis functions usually spends much computation time in interpolation processes. In this paper, a moving data method with common data registers is proposed by analyzing the tree structures of basis functions with adjacent index-of-sum.

Considering the tree structure as shown in Fig. 4., the basis functions $N_{i,4}(u)$ are affected over the parameter range from u_i to u_{i+4+1}. For instance, the basis function $N_{0,4}(u)$ is affected over the parameter range from u_0 to u_5 and the basis function $N_{1,4}(u)$ is affected over the parameter range from u_1 to u_6. For conveniently analyzing the computation of basis functions $N_{i,4}(u)$ with different parameter ranges, the representation functions K_i^j are defined as the basis functions with parameter u within the range $[u_{j+i}, u_{j+i+1})$. For instance, K_i^0 denote the representation functions of $N_{i,4}(u)$ with parameter u within the range $[u_i, u_{i+1})$; K_i^1 denote the representation functions of $N_{i,4}(u)$ with parameter u within the range $[u_{i+1}, u_{i+2})$; K_i^j denote the representation functions of $N_{i,4}(u)$ with parameter u within the range $[u_{i+j}, u_{i+j+1})$. The representation functions K_i^j with different index-of-sum are arranged in Fig. 5.

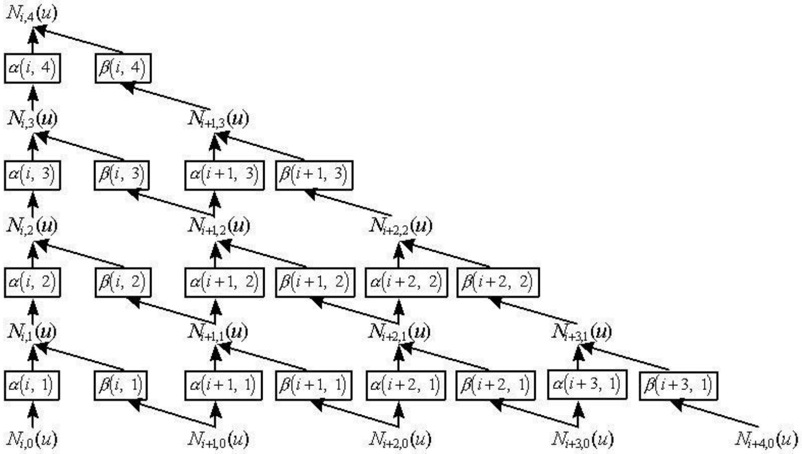

Fig. 4: The tree structure of basis functions $N_{i,4}(u)$ with p=4 and n=8.

	u_0	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_{11}	u_{12}	u_{13}
$N_{0,4}(u)$	K_0^0	K_0^1	K_0^2	K_0^3	K_0^4									
$N_{1,4}(u)$		K_1^0	K_1^1	K_1^2	K_1^3	K_1^4								
$N_{2,4}(u)$			K_2^0	K_2^1	K_2^2	K_2^3	K_2^4							
$N_{3,4}(u)$				K_3^0	K_3^1	K_3^2	K_3^3	K_3^4						
$N_{4,4}(u)$					K_4^0	K_4^1	K_4^2	K_4^3	K_4^4					
$N_{5,4}(u)$						K_5^0	K_5^1	K_5^2	K_5^3	K_5^4				
$N_{6,4}(u)$							K_6^0	K_6^1	K_6^2	K_6^3	K_6^4			
$N_{7,4}(u)$								K_7^0	K_7^1	K_7^2	K_7^3	K_7^4		
$N_{8,4}(u)$									K_8^0	K_8^1	K_8^2	K_8^3	K_8^4	

Fig. 5: The representation functions with different index-of-sum (p=4 and n=8).

As shown in Fig. 5, some conclusions are obtained as follows:

- The basis functions $N_{i,p}(u)$ are affected over the parameter range from u_i to u_{i+p+1}.

- The basis functions $N_{i,p}(u)$ are affected by (p+1) representation functions.
- The computation of basis functions $N_{i,p}(u)$ can be divided into three categories as
  - Category 1: The parameter u is in the range $\lfloor u_0, u_p)$. For each range $[u_i, u_{i+1})$ with $i < p$, there are (i+1) basis functions, $N_{0,p}(u), ..., N_{i,p}(u)$, that must be concerned for computing NURBS curves.
  - Category 2: The parameter u is in the range $\lfloor u_p, u_{n+1})$. For each range $[u_i, u_{i+1})$ with $p \le i < n+1$, there are (p+1) basis functions, $N_{i,p}(u), ..., N_{i+p,p}(u)$, that must be concerned for computing NURBS curves.
  - Category 3: The parameter u is in the range $\lfloor u_{n+1}, u_{n+p+1})$. For each range $[u_i, u_{i+1})$ with $n+1 \le i < n+p+1$, there are (n+p+1-i) basis functions, $N_{i-p,p}(u), ..., N_{n,p}(u)$, that must be concerned for computing NURBS curves.

Therefore, instead of (n+1) basis functions in Eqn. (2.1), (p+1) basis functions at most are required for interpolating NURBS curves.

Fig. 6 shows the tree structure of basis functions $N_{i+1,4}(u)$. Comparing with the tree structure of basis functions $N_{i,4}(u)$ as shown in Fig. 4, some data in registers can be reused for computing basis functions $N_{i+1,4}(u)$. Some registers in gray area of Fig. 6 denote the common data registers for computing basis functions $N_{i,4}(u)$ and $N_{i+1,4}(u)$. Note that $N_{i,4}(u)$ and $N_{i+1,4}(u)$ are basis functions with adjacent index-of-sum. Therefore, for efficiently computing basis functions $N_{i+1,4}(u)$, common data registers are moved from the tree structure of basis functions $N_{i,4}(u)$ to the tree structure of basis functions $N_{i+1,4}(u)$. The method by moving common data registers is defined as the moving data method in this paper.

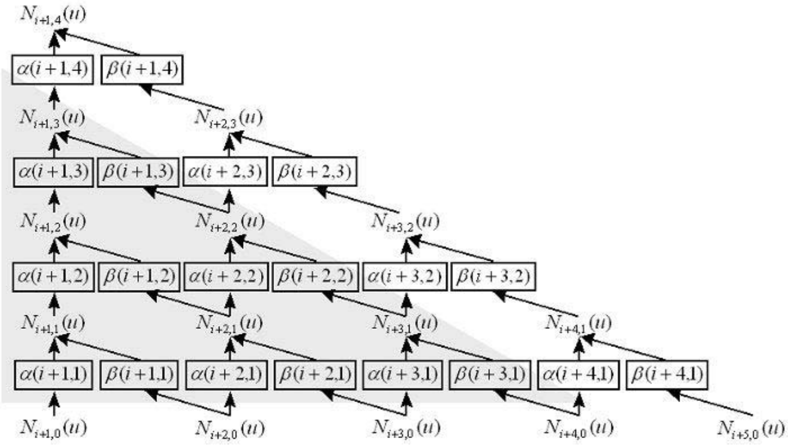

Fig. 6: The tree structure of basis functions $N_{i+1,4}(u)$ with p=4 and n=8.

4. SIMULATIONS AND MACHINING TESTS

4.1 Interpolation Time for Interpolating a NURBS Curve Point

A personal computer with DOS operating system and a Pentium 200 MHz CPU is employed to verify the computation efficiency of different approaches. The recorded interpolation time is the average value of the time for computing 10,000 interpolating points. A NURBS curve with different degrees, p, is applied in simulations. Tab. 1. shows the interpolation time compared with different approaches. Clearly, the conventional approach provides the longest interpolation time and the existing approach provides shorter interpolation time than the conventional approach. The proposed approach with a new computation structure, an extended basis function, and a moving data method provides the shortest interpolation time. Usually, the degree of NURBS curves is closely related to the interpolation time. Tab. 1 shows that raising the degree substantially increases the interpolation time of applying conventional and existing approaches. The increasing rates of conventional and existing approaches are 18.7 and 13.8, respectively. By applying the proposed approach, raising the degree slightly increases the interpolation time. The increasing rate is 4.3. Therefore, the proposed approach is more suitable for interpolating NURBS curves with higher degrees. Also, the simulation results indicate that the proposed approach is applicable for implementing real-time NURBS curve interpolators on CNC machines.

	Degree, *p*					Increasing rate
	1	2	3	4	5	
Conventional approach [12]	22 (μsec)	49 (μsec)	110 (μsec)	214 (μsec)	434 (μsec)	$\frac{434-22}{22} \cong 18.7$
Existing approach [4]	20 (μsec)	47 (μsec)	95 (μsec)	143 (μsec)	296 (μsec)	$\frac{296-20}{20} \cong 13.8$
Proposed approach	4 (μsec)	8 (μsec)	11 (μsec)	16 (μsec)	21 (μsec)	$\frac{21-4}{4} \cong 4.3$

Tab. 1: The interpolation time compared with different approaches.

4.2 NURBS Curve Interpolation Test

A circular NURBS curve with diameter 20 millimeters is applied to test the interpolation accuracy of the proposed approach. Fig. 7 shows the errors of interpolation points around the applied NURBS curve. The maximum error is 1.7×10^{-3} μm that is smaller than the least unit of 1 μm in conventional CNC machines.

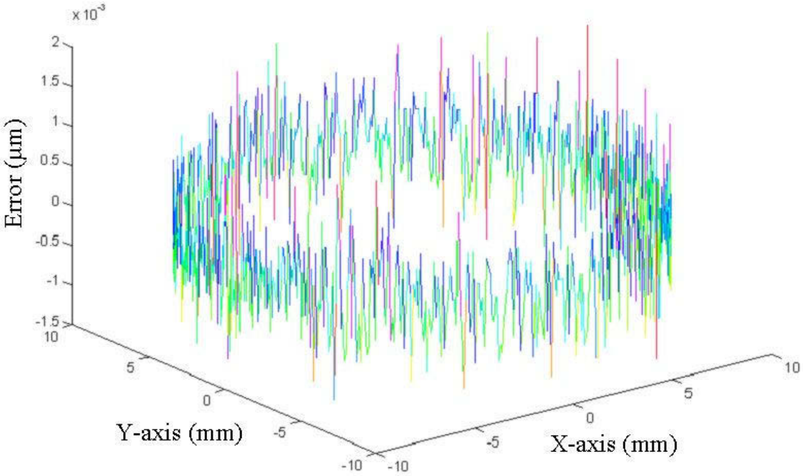

Fig. 7: The absolute errors of interpolation points.

4.3 NURBS Curve Machining Test

The proposed scheme is applied to a LEADWELL MCV-OP CNC machining center composed of an industrial PC, a DSP-based motion control card, and a mechanical system with three 3-phase Panasonic AC servo packs. Fig. 8 shows the applied LEADWELL MCV-OP CNC machining center. The 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, logical controller for controlling I/O devices, the motion generator for generating motion commands, and the central processor for handling machining procedures. The DSP-based motion control card with a high-performance TI TMS320C32 digital signal processor (DSP) is applied to interpolate motion commands and to control three AC servo packs at a sampling period of 1 *ms*. Some functions are also implemented on the DSP-based motion control card for accessing and controlling external hardware. The industrial PC sends motion commands to the DSP-based motion control card via dual port RAM. For instance, the motion command of a NURBS curve includes the upper index-of-sum, degree, control points, weights, and knot vector.

Fig. 8: LEADWELL MCV-OP CNC machining center.

Fig. 9 shows the machining results of circle commands with different interpolation approaches. Fig. 9(a), Fig. 9(b), and Fig. 9(d) are machining results of applying conventional interpolation method. Fig. 9(c) is the machining result of applying the proposed interpolation scheme. Tab. 2 tabulates the machining tests with different interpolation approaches. For conventional approaches, increasing the feedrate command usually increases the roundness of machining results. The roundness is increased from 16 μm to 24 μm when the feedrate command is increased from 1,000 $^{mm}/_{min}$ to 5,000 $^{mm}/_{min}$. However, by applying the proposed approach in this paper, the roundness is maintained at 14 μm even though the feedrate command is given as 3,000 $^{mm}/_{min}$. Although some factors, such as mechanical factors and control factors of motion systems, may significantly affect machining results, the results shown in Fig. 9 and Tab. 2 indicate that the proposed scheme provides good interpolation results and is feasible for implementation on CNC machines.

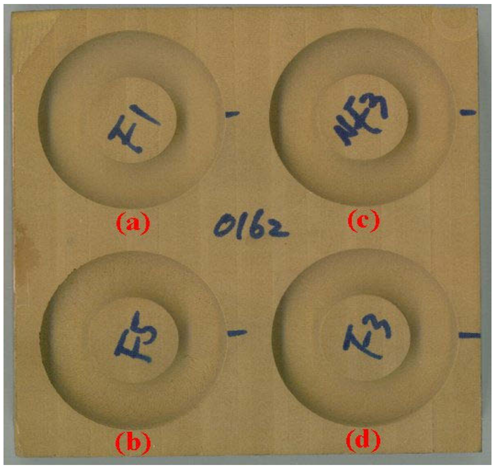

Fig. 9: NURBS curve machining results.

	Fig. 9(a)	Fig. 9(b)	Fig. 9(c)	Fig. 9(d)
Feedrate ($^{mm}/_{min}$)	1,000	5,000	3,000	3,000
Measured radius (mm)	19.439	19.423	19.438	19.434
Roundness (μm)	16	24	14	20

Tab. 2: The machining tests with different interpolation approaches.

5. CONCLUSIONS

In this paper, a fast scheme with a new computation structure, an extended basis function, and a moving data method is proposed to reduce the time for interpolating NURBS curves. The computation structure considering the formula in extended space provides an efficient method for computing NURBS curves and their derivatives. By applying the extended branch structure, the extended basis function provides a timesaving approach for simultaneously computing basis functions and derivatives. The moving data method significantly reduces the interpolation time by using common data registers and the tree structures of basis functions with adjacent index-of-sum. Some simulation and machining results indicate that the proposed scheme provides good computation efficiency and good interpolation results for interpolating NURBS curves on CNC machines.

6. ACKNOWLEDGEMENTS

The authors would like to thank Mr. Hsin-Chuan Su and Mr. Mao-Feng Tu at the Mechanical and Systems Laboratories of Industrial Technology Research Institute for their contributions in preparing the experimental setup. This work was supported in part by the Industrial Technology Research Institute of Taiwan, and National Science Council of the Republic of China under Contract NSC 95-2218-E-027-026.

7. REFERENCES

[1] Bahr, B.; Xiao, X.; Krishnan, K.: A real-time scheme of cubic parametric curve interpolations for CNC systems, Computers in Industry, 45(3), 2001, 309-317.

[2] Chang, Y. F.; Hong, R. C.: Parametric curve machining of a CNC milling EDM, International Journal of Machine Tools and Manufacture, 45(7-8), 2005, 941-948.

[3] Cheng, C. W.; Tsai, M. C.; Maciejowski, J.: Accurate feedrate control of CNC machine tools along NURBS curves, Proceedings of the IEEE Conference on Decision and Control, 3, 2004, 3053-3058.

[4] Cheng, C. W.; Tsai, M. C.: Real-time variable feed rate NURBS curve interpolator for CNC machining, International Journal of Advanced Manufacturing Technology, 23(11-12), 2004, 865-873.

[5] Choi, I. H.; Yang, M. Y.; Hong, W. P.; Jung, T. S.: Curve interpolation with variable feedrate for surface requirement, International Journal of Advanced Manufacturing Technology, 25(3-4), 2005, 325-333.

[6] Chuang, S. H.; Kao, C. Z.: One-sided arc approximation of B-spline curves for interference-free offsetting, Computer Aided Design, 31(2), 1999, 111-118.

[7] Jee, S.; Koo, T.: Tool-path generation for NURBS surface machining, Proceedings of the American Control Conference, 3, 2003, 2614-2619.

[8] Meek, D. S.; Walton, D. J.: Approximating quadratic NURBS curves by arc splines, Computer Aided Design, 25(6), 1993, 371–376.

[9] Ong, C. J.; Wong, Y. S.; Loh, H. T.; Hong, X. G.: Optimization approach for biarc curve-fitting of B-spline curves, Computer Aided Design, 28(12), 1996, 951-959.

[10] Park, H.: Error-bounded biarc approximation of planar curves, Computer Aided Design, 36(12), 2004, 1241-1251.

[11] Park, J.; Nam, S.; Yang, M.: Development of a real-time trajectory generator for NURBS interpolation based on the two-stage interpolation method, International Journal of Advanced Manufacturing Technology, 26(4), 2005, 359-365.

[12] Piegl, L.; Tiller, W.; The NURBS Book, Springer, New York, 1997.

[13] Piegl, L. A.; Tiller, W.: Biarc approximation of NURBS curves, Computer Aided Design, 34(11), 2002, 807-814.

[14] Tsai, M. C.; Cheng, C. W.; Cheng, M. Y.: A real-time NURBS surface interpolator for precision three-axis CNC machining, International Journal of Machine Tools and Manufacture, 43(12), 2003, 1217-1227.

[15] Yang, M. Y.; Park, J. H.: A study on an open architecture CNC system with a NURBS interpolator for WEDM, International Journal of Advanced Manufacturing Technology, 19(9), 2002, 664-668.

[16] Yeh, S. S.; Hsu, P. L.: Adaptive-feedrate interpolation for parametric curves with a confined chord error, Computer Aided Design, 34(3), 2002, 229-237.

[17] Yeh, S. S.; Hsu, P. L.: Speed-controlled interpolator for machining parametric curves, Computer Aided Design, 31(5), 1999, 349-357.