

# Development of Machinable Ellipses by NURBS Curves

Yuan L. Lai, Jian H. Chen, and Jui P. Hung

**Abstract**—Owing to the high-speed feed rate and ultra spindle speed have been used in modern machine tools, the tool-path generation plays a key role in the successful application of a High-Speed Machining (HSM) system. Because of its importance in both high-speed machining and tool-path generation, approximating a contour by NURBS format is a potential function in CAD/CAM/CNC systems. It is much more convenient to represent an ellipse by parametric form than to connect points laboriously determined in a CNC system. A new approximating method based on optimum processes and NURBS curves of any degree to the ellipses is presented in this study. Such operations can be the foundation of tool-radius compensation interpolator of NURBS curves in CNC system. All operating processes for a CAD tool is presented and demonstrated by practical models.

**Keywords**—Ellipse, Approximation, NURBS, Optimum.

## I. INTRODUCTION

**D**UE to the widespread use of NURBS-based curve representation and motion control, the demand of High-Speed Machining (HSM) is increasing. In the application of HSM, one of the important components which are often neglected is the tool-path formed strategy. The successful application of HSM can only be realized in associating with proper CAM functions. There are many methods of drawing an ellipse, including foci, trammel, concentric-circle, oblique-circle, parallelogram [1], two-circle and four-center [2] methods. The ellipses may be regarded as a distortion of a circle by unequal scale factors in two directions. It is simple but not efficient for traditional tools to deal with, like scale and offset. It is much more convenient to represent an ellipse by parametric form than to connect points laboriously determined in a CNC system. Recently, Rosin [3] has made an impressive survey and comparison on four-arc approximation to ellipses. Qians [4] proposed an analytical function to find optimum arcs to approximate an ellipse by maximum error function. It has been widely used in CAD field for many years and is gradually applied in CAM area with the prevalence of NURBS

interpolators equipped in CNC controllers. With NURBS interpolation, the machining of sculptured surfaces is carried out on a series of NURBS curves rather than a large number of short lines or arcs, which is the case of CNC controllers equipped with linear and circular interpolators only. This results in more fluid tool paths, makes the tool change its moving direction more smoothly and maintains a higher average feed rate.

Although approximations to circular arcs and full circles by parametric forms have been investigated [5-9], there has been very little work published on ellipses approximation using NURBS format. A real elliptic part can be made by the corresponding cutting paths guided by tool-radius compensation interpolator. In addition, these operations based on parametric curves offset algorithms [10-16].

Fig. 1 illustrates the cutting task of NURBS-based CNC machining [17]. In modern CAD/CAM systems, more and more profiles are represented in parametric forms to meet the requirement of HSM. Many researchers have developed real time parametric interpolators for curve generation using NURBS basis function [18-22]. Some major commercial controller manufacturers have brought NURBS operations into industry, such as Fanuc 15M/16M [23] and Siemens 840D [24]. Once the spline format is sent to the controller, the processor directly interpolates the segments at extremely tiny intervals. The architecture of the controller has look-ahead features that will change the feed-rate dynamically to adapt the spindle to rapid changes in direction.

This report tells a method based on optimum processes and NURBS curves of any degree to represent ellipses. By implementing the proposed algorithm on approximations for ellipses and ellipses offset, the results show accurate performances evidently. It can be made as comprehensive as one can image, the larger the number of control points is, the higher the quality of effectiveness is.

The organization of the chapter is as follows. In section 2 the procedures for acquiring initial conditions are described. Section 3 discusses and builds global error functions and section 4 optimizes locations and weights of all control points. Examples and comparisons are illustrated in section 5. In section 6, the generated tool paths are verified through NURBS machining and close this study.

Yuan L. Lai is with the Department of Mechanical and Automation Engineering, Da-yeh University, Changhua, Taiwan, R.O.C. (phone: +886-4-8511888; fax: 886-4-886-4-8511224, e-mail: yllai@mail.dyu.edu.tw)

Jian H. Chen is with the School of Physical Therapy, Chung Shan Medical University, Taichung, Taiwan, R.O.C.

Jui P. Hung is with the Department of Automation Engineering, National Chin-Yi University of Technology, Taichung, Taiwan, R.O.C.

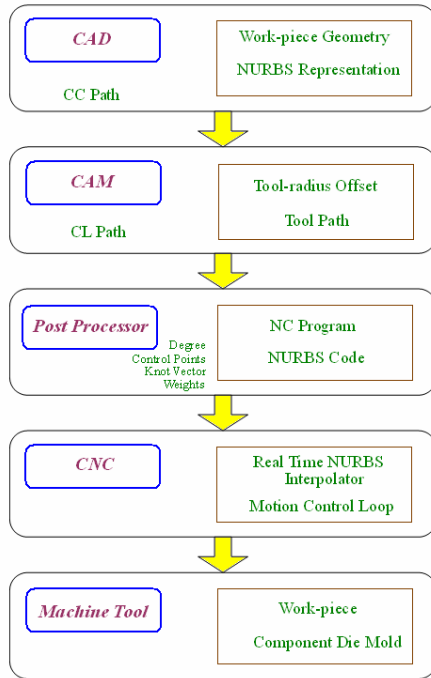


Fig. 1 Machining system for NURBS curves

## II. ESTABLISH THE INITIAL CONDITIONS

### A. Equivalent Approximating Curve

A  $n$  sides open shape consist of  $(n+1)$  control points, thus  $(n+1)$  control points can form a B-spline curve which is defined by the following equation:

$$C(t) = \sum_{i=0}^n N_{i,k}(t) P_i$$

A NURBS curve is defined by the following equation:

$$C(t) = \frac{\sum_{i=0}^n N_{i,k}(t) w_i P_i}{\sum_{j=0}^n N_{j,k}(t) w_j} = \sum_{i=0}^n R_{i,k}(t) P_i, \quad (1)$$

$$R_{i,k}(t) = \frac{N_{i,k}(t) w_i}{\sum_{j=0}^n N_{j,k}(t) w_j}$$

Where  $P_i$  are vectors composed of  $x$  and  $y$  coordinates of the control points,  $w_i$  are weights for each point,  $N_{i,k}(t)$  are B-spline basis functions.  $k$  ( $k = p+1$ ,  $p$  is the degree of basis function. For example, when order is 3, then rank in the definition of NURBS is 2, i.e. the NURBS expression is expressed by  $t^2$ ,  $t$  and constant.) is the order of a B-spline curve. The basis function is expressed by de Boor-Cox as follows:

$$N_{i,1}(t) = \begin{cases} 1, & \text{if } (t_i \leq t < t_{i+1}) \\ 0, & \text{otherwise} \end{cases}$$

$$N_{i,k}(t) = \frac{t-t_i}{t_{i+k+1}-t_i} N_{i,k-1}(t) + \frac{t_{i+k}-t}{t_{i+k}-t_{i+1}} N_{i+1,k-1}(t) \quad (2)$$

$[t_i], i = 0 \sim (n+k), t_i \leq t_{i+1}$ , is the knot vector.

The derivative of a B-spline curve is:

$$C'(t) = \sum_{i=0}^n N'_{i,k}(t) P_i \quad (3)$$

$$N'_{i,1}(t) = 0$$

$$N'_{i,k}(t) = \frac{N_{i,k-1}(t) + (t-t_i)N'_{i,k-1}(t)}{t_{i+k+1}-t_i} + \frac{(t_{i+k}-t)N'_{i+1,k-1}(t) - N_{i+1,k-1}(t)}{t_{i+k}-t_{i+1}}$$

The derivative of a NURBS curve is:

$$C'(t) = \sum_{i=0}^n R'_{i,k}(t) P_i, \quad w(t) = \sum_{j=0}^n N_{j,k}(t) w_j,$$

$$A_i(t) = N_{i,k}(t) w_i \quad (4)$$

$$R'_{i,k}(t) = \frac{A_i'(t)}{w(t)}, \quad R'_{i,k}(t) = \frac{w(t)A_i'(t) - w'(t)A_i(t)}{w(t)^2}$$

If a series of control points exist, one can refine the shape of such a curve in the following different ways. They offer a wide range of tools to design and analyze shape deformation. They are:

Moving of the control points  $P_i$ .

Increasing or decreasing the number of control points.

Multiple locating of the control points.

Changing the order  $k$  of the basis functions.

Modifying the weights  $w_i$  for each control point.

Replacing the basis function. (closed uniform, open uniform, non-uniform)

Rearranging the knot vector  $[t_i]$ .

Using multiple knot values in knot vector.

Changing the relative spacing of the knots.

An ellipse can be represented parametrically by the equations

$$\begin{cases} x = a \cos \theta \\ y = b \sin \theta \end{cases} \quad (5)$$

Where  $a$  and  $b$  are major and minor axes,  $a > b$ .  $x$  and  $y$  are the rectangular coordinates of any point on the ellipse, and the parameter  $\theta$  is the angle at the center measured from the  $x$ -axis anticlockwise. First, the initial point-sequenced curve  $C(t)$  is sampled based on equation (5). The proposed method is shown in the following steps. By using this method, one can find the initial curve control points.

The procedure using the proposed algorithm to obtain the initial condition is as follows:

1. Form a matrix equation to represent the curve using the NURBS format.

$$\begin{bmatrix} D_1(t_1) \\ D_2(t_2) \\ D_3(t_3) \\ \dots \\ D_j(t_j) \end{bmatrix} = \begin{bmatrix} N_{1,k}(t_1) & N_{2,k}(t_1) & N_{3,k}(t_1) & \dots & N_{i,k}(t_1) \\ N_{1,k}(t_2) & N_{2,k}(t_2) & N_{3,k}(t_2) & \dots & N_{i,k}(t_2) \\ N_{1,k}(t_3) & N_{2,k}(t_3) & N_{3,k}(t_3) & \dots & N_{i,k}(t_3) \\ \dots & \dots & \dots & \dots & \dots \\ N_{1,k}(t_j) & N_{2,k}(t_j) & N_{3,k}(t_j) & \dots & N_{i,k}(t_j) \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ \dots \\ P_i \end{bmatrix},$$

$$2 \leq k \leq n+1 \leq j$$

$$[D] = [N][P] \tag{6}$$

2. From step 1, if [N] is a Square Matrix then

$$[P] = [N]^{-1}[D] \tag{7}$$

3. If [N] is a Non-Square Matrix then

$$[D] = [N][P]$$

$$[P] = [[N]^T[N]]^{-1}[N]^T[D] \tag{8}$$

[P]: unknown, [D]: known.

4. In order to get [N], determine the parameters  $t_i$ ,

$$t_i = \frac{\sum_{s=2}^i |D_s - D_{s-1}|}{\sum_{s=2}^l |D_s - D_{s-1}|}, \quad l \geq 2$$

$$t_i = \frac{\sum_{s=2}^i |D_s - D_{s-1}|}{\sum_{s=2}^l |D_s - D_{s-1}|} t_{\max}, \quad l \geq 2 \tag{9}$$

5. Solve the above equations to find [P]

Repeat the above process, two NURBS curves  $C_a(t)$  and  $C_b(t)$  are obtained by specifying two different numbers of control points. In order to acquire a precise approximating curve, a bigger control point number is provided to  $C_a(t)$ . The other one is provided a reasonable control point number to acquire a implementing curve  $C_b(t)$ , which is expected to store fewer control points. This control point number depend on the *major/minor* ratio. Two NURBS curves are obtained to be the initial conditions. If there is no pressing reason for doing otherwise, here B-Spline should be defined as: No multiple control points. Open uniform knot vector. The format can be accepted by every CAD/CAM/CNC system presently in use. All default weights  $w_i$  are assigned one to each control point of the NURBS curve, which is the same as B-spline curve. By

using the evaluating bound error function, the implementing curve  $C_b(t)$  will be modified to optimum conditions in section 3.

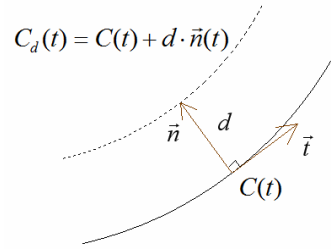


Fig. 2 Illustration of tangent and normal vectors of  $C(t)$

### B. Approximating Offset Curve

Second, the initial offset point-sequenced curve is sampled based on normal direction of the original curve, as shown in Fig. 2. Sampled points can be obtained by

$$C_d(t) = C(t) + d \cdot \vec{n}(t) \tag{10}$$

where  $\vec{n}(t)$  is the unit normal vector of  $C(t)$ ,  $d$  is the offset distance.

$$\vec{n}(t) = \frac{(y'(t), -x'(t))}{\sqrt{x'(t)^2 + y'(t)^2}}, \quad (x'(t), y'(t)) \neq (0,0) \tag{11}$$

Repeat the same procedures as described in previous section, an approximating offset curve  $C_d(t)$  can be obtained. A corresponding NURBS curve  $C_e(t)$  is formed by  $C_d(t)$ . This offset NURBS curve also can be acquired by equation (4). Same as above, by using the evaluating bound error function, the implementing curve  $C_e(t)$  will be modified to optimum conditions in section 3. It seems the simplest way is to set parameter  $[t_i]$  spread the curve regularly.

## III. BUILD EVALUATING BOUND ERROR FUNCTIONS

### Equivalent Approximating Curves

There were two popular approximating error measures at finite sample points along  $C(t)$  by computing equations (13) and (15) in earlier studies.

$$\varepsilon(t) = \|C_b(t) - C_a(t)\| \tag{12}$$

Minimize maximum Euclidean distance:

$$E_1 = \max(\varepsilon(t)) = \max(\|C_b(t) - C_a(t)\|) \tag{13}$$

or

$$\varepsilon(t) = \|C_b(t) - C_a(t)\|^2 \tag{14}$$

Minimize maximum squares distance:

$$E_2 = \max(\varepsilon(t)) = \max(\|C_b(t) - C_a(t)\|^2) \tag{15}$$

The above measuring errors represent local variations. They need to be transformed to measure global variations throughout

the curve  $C_a(t)$ . Assuming two parametric curves  $C_a(t_i)$  and  $C_b(t_j)$  with similar orientation knot vectors  $[t_i]$ , one can normalize these two knot vectors  $[t_i]$  and  $[t_j]$  into one knot vector, i.e.  $\min[t_i]=0$ ,  $\max[t_i]=1$ ,  $\min[t_j]=0$  and  $\max[t_j]=1$ , then  $t_i = t_j = t \cdot (t_i, t_j \in [0,1])$

Extend the Euclidean distance function (13) and (15) between  $C_a(t)$  and  $C_b(t)$  to global difference functions:

$$\varepsilon_1(t) = \|C_b(t) - C_a(t)\|$$

$$E_1 = \int_a^b \varepsilon_1(t) dt \approx E_1 = \sum_{i=1}^m \varepsilon_1(t_i) = \sum_{i=1}^m \|C_b(t_i) - C_a(t_i)\| \quad (16)$$

$$\varepsilon_2(t) = \|C_b(t) - C_a(t)\|^2$$

$$E_2 = \int_a^b \varepsilon_2(t) dt \approx E_2 = \sum_{i=1}^m \varepsilon_2(t_i) = \sum_{i=1}^m \|C_b(t_i) - C_a(t_i)\|^2 \quad (17)$$

where  $m$  is the sampling number.

The above global error functions are perhaps the simplest ways to measure error for the approximate curves  $C_b(t)$  to set  $t_i = t_j = t$ . After comparing these two error functions,  $E_1$  performs better than  $E_2$ . This article uses  $E_1$  as the EBE function.

### B. Approximating Offset Curves

In this section, a very different method to previous ones is used to offset a NURBS curve. The following are the major steps:

1. Build an Evaluating Bound Error function.
2. Sample an offset point-sequenced curve based on the original curve's first derivatives.
3. Use sampling points and pre-specified parameters (such as number of control points, degree iteration number) to form a new offset curve as described in the previous section.
4. Relocate the first and last control points to the exact positions, i.e., constraints of endpoint  $C_0$  continuity.
5. During each iteration, optimize the other control points by using a web-like search algorithm.

In order to find a better solution, build an estimating function is indispensable to this work to measure errors. Earlier studies measured approximate errors, in general, at finite sample points along  $C_a(t)$  by computing:

$$\varepsilon_1(t) = \text{abs}(\|C_e(t) - C_a(t)\| - d)$$

$$E_1 = \max(\varepsilon_1(t)) = \max(\text{abs}(\|C_e(t) - C_a(t)\| - d)) \quad (18)$$

where  $\varepsilon_1(t)$  is the offset distance variation. Elber and Cohen [25] proposed another measuring criterion to improve the offset performance by computing:

$$\varepsilon_2(t) = \text{abs}(\|C_e(t) - C_a(t)\|^2 - d^2)$$

$$E_2 = \max(\varepsilon_2(t)) = \max(\text{abs}(\|C_e(t) - C_a(t)\|^2 - d^2)) \quad (19)$$

The above measuring error represents local variations. They need to be transformed to measure global variations throughout the curve  $C_a(t)$ . Assuming two parametric curves  $C_a(t_i)$  and  $C_e(t_j)$  with similar orientation knot vectors  $[t_i]$ , one can normalize these two knot vectors  $[t_i]$  and  $[t_j]$  into one knot vector, i.e.  $\min[t_i]=0$ ,  $\max[t_i]=1$ ,  $\min[t_j]=0$  and  $\max[t_j]=1$ , then  $t_i = t_j = t \cdot (t_i, t_j \in [0,1])$

Extend the offset distance variation function to area difference function:

$$\varepsilon_1(t) = \text{abs}(\|C_e(t) - C_a(t)\| - d)$$

$$E_1 = \int_a^b \varepsilon_1(t) dt \approx \sum_{i=1}^m \varepsilon_1(t_i) = \sum_{i=1}^m \text{abs}(\|C_e(t_i) - C_a(t_i)\| - d) \quad (20)$$

where  $m$  is the sampling number.

$$\varepsilon_2(t) = \text{abs}(\|C_e(t) - C_a(t)\|^2 - d^2)$$

$$E_2 = \int_a^b \varepsilon_2(t) dt \approx \sum_{i=1}^m \varepsilon_2(t_i) = \sum_{i=1}^m \text{abs}(\|C_e(t_i) - C_a(t_i)\|^2 - d^2) \quad (21)$$

The above area error functions are perhaps the simplest ways to measure error for the approximate offset curve  $C_e(t)$  to set  $t_i = t_j = t$ , and compare distances between points on equal parameters value with the offset distance  $d$ . However, the point  $C_e(t)$  may not lie on the normal direction at the point  $C_a(t)$  [26]. In order to conquer this directional error-prone, in this paper, we proposed a new criterion to evaluate variation throughout the curve. After comparing these two error functions,  $E_1$  perform better then  $E_2$ . This article uses  $E_1$  as our error function. This Evaluating Bound Error (EBE) function consists of offset distance  $d$  and angle variation  $\delta$ , it was defined as:

$$\theta(t) = \cos^{-1}\left(\frac{a \cdot b}{|a||b|}\right)$$

$$a = C'_a(t), b = C_e(t) - C_a(t)$$

$$\delta(t) = \text{abs}\left(\theta(t) - \frac{\pi}{2}\right), (\text{ or } b = C'_e(t), \delta(t) = \text{abs}(\theta(t)) )$$

$$f(t) = \frac{1}{\cos(\delta(t))}$$

$$E_3 = \sum_{i=1}^m \varepsilon(t_i) + f(t) \times d \quad (22)$$

where  $m$  is the sampling number.

The average Evaluating Bound Error per unit length percentage is defined as:

$$EBE = \frac{E_3 - (m \times d)}{m \times d} \times 100\% \quad (23)$$

$EBE$  equals to zero if orthogonality and exact distance are preserved.

One purpose of this  $EBE$  function tries to maintain a constant distance  $d$  between offset and base curves. The other more significant purpose can rectify these two curves approaching in the same direction.

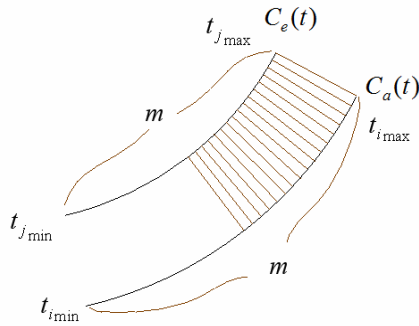


Fig. 3 Regularize the parameters of  $t_i$  and  $t_j$

The average Evaluating Bound Error  $E_{EBE}$  equals to zero if orthogonality and exact distance are preserved. Due to the convenience of comparing both base and offset curves, this study regularizes the parameters  $t_i$  uniformly at finite  $m$  segments. The illustration of related curve's phenomenon is shown in Fig. 3. There are two intentions of this error function, as shown in Fig. 3, one purpose of this  $EBE$  function tries to maintain a constant distance  $d$  between offset and base curves. The other more significant purpose can rectify these two curves approaching same direction.

#### IV. LOCATIONS AND WEIGHTS OPTIMIZATION

This study used a radiating web-like search algorithm to detect optimum positions. In this method, optimum parameters should be defined first, i.e. radius step  $r_{step}$ , angle step  $\theta_{step}$  and convergence  $d_{min}$ .

A program algorithm is offered below to help current control points to find the current optimum locations during each iteration.

```

-----
Begin
Compute initial  $E_{ini}$ 
Set  $E_{current} = E_{ini}$ 
Set Searching_step  $d_{step} = 1$ 
 $r_{now} = d_{step}$ 
Do until  $d_{step} < d_{min}$ 
    Do  $i = 1$  to  $n_{ang}$  ( $n_{ang} = \frac{2\pi}{\theta_{step}}$ )
        Set new location for current control point  $p_i$ 
         $p_{ix} = p_x + r_{now} \cos(\theta_i)$ ,  $p_{iy} = p_y + r_{now} \sin(\theta_i)$ 
        Compute  $E_i$ 
    
```

```

        If  $i=1$  or  $E_{now} > E_i$  then  $E_{now} = E_i$  and  $p_{now} = p_i$ 
    Loop  $i$ 
    If  $E_{current} > E_{now}$  then
         $E_{current} = E_{now}$  and  $r_{now} = r_{now} + d_{step}$  and
         $p_{current} = p_{now}$ 
    Else
         $r_{now} = r_{now} - d_{step}$  and  $d_{step} = d_{step} / 5$ 
    Endif
Loop
End
-----

```

The default initial search-radius step  $r_{step} = 1$ . From the above algorithm, the  $d_{step}$  is getting smaller and smaller, when it is less than convergence  $d_{min}$ , i.e.,  $r_{step} < d_{min}$ , the search process will stop on this control point. During each evaluating iteration, we perturb the control points  $p_1 \sim p_n$  to evaluate the area error. Each control point change will affect the GBE function and influence the next searching iteration stage. Thus it may change the final convergence situation of the optimum solution. In order to meet  $G_0$  continuity, the first and last control points were fixed at the exact positions. The second and  $(n)^{th}$  control points are restricted to locate at the tangent vectors of the start and end points of an ellipse, thus ensure NURBS curves can meet the requirement of  $G_1$  continuity.

A program algorithm is offered below to help current control points to find the current optimum weights during each iteration. This subroutine needs to be treated twice, one is positive direction, and the other is negative direction.

```

-----
Begin
Compute initial error  $E_{ini}$ 
Set  $no\_max = 5$ ,  $no\_err =$ 
Set  $E_{current} = E_{ini}$ 
Set Searching_step  $d_{step} = 1$ 
Do until  $no\_err > no\_max$ 
     $w_i = w_i + d_{step}$ 
    Compute  $E_{now}$ 
    If  $E_{now} < E_{current}$  then
         $E_{current} = E_{now}$ 
         $no\_err = 0$ 
    Else
         $w_i = w_i - d_{step}$ 
         $d_{step} = d_{step} / 5$ 
         $no\_err = no\_err + 1$ 
    Endif
Loop
End
-----

```

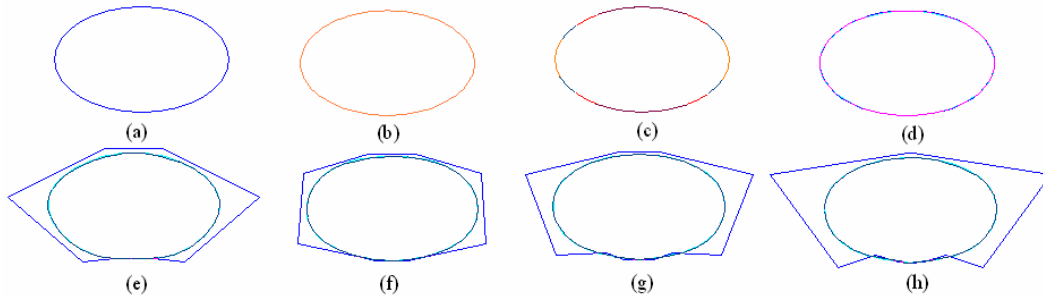


Fig. 4 (a) True ellipse with  $a=5$  and  $b=3$ . (b) Approximation by 96 straight lines. (c) Approximation by 12 circular arcs. (d) Approximation by a NURBS curve of degree 4 and 48 control points. (e) Approximation by a NURBS curve of degree 4 and 10 control points. (f) Approximation by a NURBS curve of degree 4 and 12 control points. (g) Approximation by a NURBS curve of degree 5 and 12 control points. (h) Approximation by a NURBS curve of degree 6 and 12 control points

## V. ILLUSTRATION AND APPLICATIONS

Fig. 4(a) shows a true ellipse with major radius  $a=5$  and minor radius  $b=3$ . Fig. 4(b) shows an approximation to Fig. 4(a) by 96 straight lines. Fig. 4(c) shows an approximation to Fig. 4(a) by 12 circular arcs. Fig. 4(d) shows an approximation to Fig. 4(a) by a NURBS curve of degree 4 and 48 control points without optimum processes. Fig. 4(e) shows an approximation to Fig. 4(a) by a NURBS curve of degree 4, uniform knot vector and 10 control points. Fig. 4(f) shows an approximation to Fig. 4(a) by a NURBS curve of degree 4, uniform knot vector and 12 control points. Fig. 4(g) shows an approximation to Fig. 4(a) by a NURBS curve of degree 5, uniform knot vector and 12 control points. Fig. 4(h) shows an approximation to Fig. 4(a) by a NURBS curve of degree 6, uniform knot vector and 12 control points. Fig. 4(e)-(g) were obtained with NURBS curves and treated by the proposed optimum processes. An error comparison is shown in Fig. 5. The error when using 10 control points is less than 0.16. When using 12 control points, all errors can control under 0.1. The experiment results show that no matter what degree the NURBS curve is more control points performs better than fewer control points.

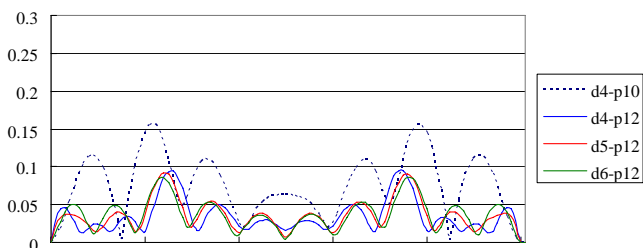


Fig. 5 An error comparison

An approximation to the above ellipse offset with a distance of 1 is shown in Fig. 6. It shows an approximation to Fig. 4(a) by a NURBS curve of degree 5, uniform knot vector and 12 control points. An error comparison is shown in Fig. 7. The error after using optimum processes can control under 0.0355.

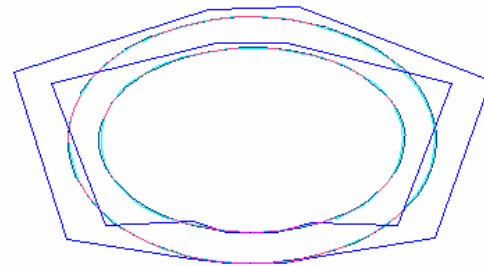


Fig. 6 An approximation to an ellipse offset

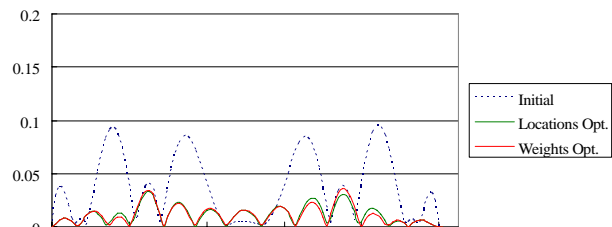


Fig. 7 An error comparison

The application of current method to engineering problems is firstly shown in Fig. 8, where outer and inner profiles are approximated by NURBS curves. Fig. 9 shows a machining process of ellipses by a machine center with FANUC 16M controller, NURBS-based tool path, high precision contour control (HPCC) and linear motors [3]. Fig. 10 shows an elliptic box (major diameter is 180mm and minor diameter is 120mm) was acquired by assembling three pieces of elliptic MDF-boards (Medium Density Fiberboard). This study presents an optimum procedure to acquire a NURBS curve to approximate an ellipse. The NURBS-format machining codes of FANUC controllers, listed in Table I, are adopted in this

study, where code "G05 P10000" and code "G05 P0" are used to activate and turn off the HPCC function, respectively. In addition, "G06.2" is the cutting instruction,  $P$  is the order of NURBS,  $K$  represents knots and  $R$  represents their associated weights.

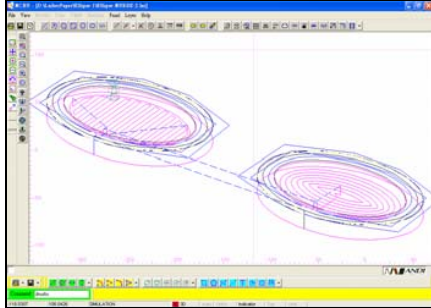


Fig. 8 Tool-path generation of ellipses



Fig. 9 MDF-board (Medium Density Fiberboard) machining of ellipses

TABLE I  
MACHINING CODES FOR THE ELLIPSES USING NURBS

Outer NURBS tool-path	Inner NURBS tool-path
G05 P10000	G05 P10000
P5 K0 X0. Y-66.005 R1. F30000	P5 K0 X0. Y-53.995 R1. F30000
K0 X-17.706 Y-66.002 R1.	K0 X-15.546 Y-53.974 R1.
K0 X-43.677 Y-60.753 R1.047	K0 X-36.107 Y-49.614 R1.068
K0 X-103.174 Y-40.73 R1.019	K0 X-90.574 Y-34.851 R1.028
K0 X-99.535 Y40.486 R1.027	K0 X-86.291 Y34.496 R1.039
K1 X-29.327 Y68.157 R1.027	K1 X-24.236 Y55.54 R1.038
K2 X29.323 Y68.146 R1.027	K2 X24.219 Y55.523 R1.039
K3 X99.547 Y40.499 R1.027	K3 X86.344 Y34.518 R1.038
K4 X103.151 Y-40.746 R1.019	K4 X90.476 Y-34.876 R1.027
K5 X43.7 Y-60.745 R1.047	K5 X36.19 Y-49.601 R1.066
K6 X17.694 Y-66.003 R1.	K6 X15.516 Y-53.971 R1.
K7 X0. Y-66.005 R1.	K7 X0. Y-53.995 R1.
K8.	K8.
K8.	K8.
K8.	K8.
K8.	K8.
K8.	K8.
G05 P0	G05 P0



Fig. 10 A MDF-board elliptic box

## VI. CONCLUSION

This study introduces an accurate algorithm for ellipse approximation, an algorithm based on optimum processes and NURBS curves of any degree. In such a way, the original data used to define NURBS curves, which includes control points, weights and knots, will be transmitted into controllers as G-codes. The entire curve which was once described by many blocks of short lines or arcs now requires a single program block only. As a result, the size of NC programs can be greatly reduced. The size reduction is up to 1/10 to 1/100 relative to a linear-interpolation part program. This partially solved one of the disadvantages of applying linear and circular interpolations in HSM - large size of NC programs for complicated geometries. NURBS interpolation is one type of curve interpolation. To take advantage of the capability, one requirement is a CAM system capable of outputting NURBS tool paths. This method is serviceable in engineering applications. The proposed method detects optimum locations for each control point by perturbing them with a radiating web-like search algorithm. These control points form an accurate ellipse with a uniform knot vector which can be accepted by any modern CAD/CAM/CNC system. By implementing the similar processes but different error functions, an offset curve can be obtained with NURBS representation. Next stage of this research, by refining the searching correlation between control point positions and corresponding weights, the proposed operating model possibly work on approximation to general curves with NURBS curves. Such operations can be the foundation of tool-radius compensation interpolator of NURBS curves in CNC system.

## REFERENCES

- [1] F.E. Giesecke, A. Mitchell, H.C. Spencer, I.L. Hill, R.O. Loving, J.T. Dygdon, Engineering Graphics, Macmillan, 1985.
- [2] C. Jensen, and J.D. Hesel, Engineering Drawing and Design, McGraw-Hill, 1985.
- [3] P.L. Rosin, A survey and comparison of traditional piecewise circular approximations to the ellipse, *Computer Aided Geometric Design*, 16, 1999, pp.269-286.
- [4] W.H. Qian, and K. Qian, Optimising the four-arc approximation to ellipses, *Computer Aided Geometric Design*, 18, 2001, pp.1-19.
- [5] C. Blanc, and C. Schlick, Accurate Parametrization of Conics by NURBS, *IEEE Computer Graphics and Applications*, 16(3), 1996, pp.64-71.
- [6] J.M. Carnicer, E. Mainar, J.M. Pena, Representing circles with five control points, *Computer Aided Geometric Design*, 20, 2003, pp.501-511.
- [7] J.J. Chou, Higher order Bezier circles, *Computer-Aided Design*, 27, 1995, pp.303-309.

- [8] L.A. Piegl, and W. Tiller, A Menagerie of Rational B-Spline Circles, *IEEE Computer Graphics and Applications*, 9(5), 1989, pp.48-56.
- [9] L.A. Piegl, and W. Tiller, Circle approximation using integral B-splines, *Computer-Aided Design*, 35, 2003, pp.601-607.
- [10] S.H.F. Chuang, and C.Z. Kao, One-sided arc approximation of B-spline curves for interference-free offsetting, *Computer-Aided Design*, 31, 1999, pp.111-118.
- [11] G. Elber, I.K. Lee, M.S. Kim, Comparing offset curve approximation methods, *IEEE Computer Graphics and Applications*, 17(3), 1997, pp.62-71.
- [12] I.K. Lee, M.S. Kim, G. Elber, Planar curve offset based on circle approximation, *Computer-Aided Design*, 28, 1996, pp.617-630.
- [13] Y.M. Li, V.Y. Hsu, Curve offsetting based on Legendre series, *Computer Aided Geometric Design*, 15, 1998, pp.711-720.
- [14] T. Maekawa, An overview of offset curves and surface, *Computer-Aided Design*, 31, 1999, pp.165-173.
- [15] J.B.S. Oliveira, and L.H. Figueiredo, Robust Approximation of Offsets and Bisectors of Plane Curves, *IEEE Computer Graphics and Image Processing*, 2000, pp.139-145. 1
- [16] L.A. Piegl, and W. Tiller, Computing offsets of NURBS curves and surfaces, *Computer-Aided Design*, 31, 1999, pp.147-156.
- [17] M.Y. Cheng, M.C. Tsai, J.C. Kuo, Real-time NURBS command generators for CNC servo controllers, *International Journal of Machine Tools and Manufacture*, 42, 2002, pp.801-813.
- [18] M. Tikhon, T.J. Ko, S.H. Lee, H.S. Kim, NURBS interpolator for constant material removal rate in open NC machine tools, *International Journal of Machine Tools and Manufacture*, 44, 2004, pp.237-245.
- [19] 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, pp.1217-1227.
- [20] S.S. Yeh, and P.L. Hsu, Adaptive-feedrate interpolation for parametric curves with a confined chord error, *Computer-Aided Design*, 34, 2002, pp.229-237.
- [21] T. Yong, and R. Narayanaswami, A parametric interpolator with confined chord errors, acceleration and deceleration for NC machining, *Computer-Aided Design*, 35, 2003, pp.1249-1259.
- [22] Q.G. Zhang, and B.R. Greenway, Development and implementation of a NURBS curve motion interpolator, *Robotics and Computer-Integrated Manufacturing*, 14, 1998, pp.27-36.
- [23] FANUC LTD., FANUC Series 16i/160i/160is-MB Operator's Manual, 2001.
- [24] Siemens AG, Sinumerik 840D/810D/FM-NC Programming Guide: Advanced, 1997.
- [25] G. Elber, and E. Cohen, Error Bounded Variable Distance Offset Operator for Free Form Curves and Surfaces, *International Journal of Computational Geometry and Application*, 1(1), 1991, pp.67-78.
- [26] R.T. Farouki, Conic Approximation of Conic Offsets, *Journal of Symbolic Computation*, 23, 1997, pp.301-313.