

O. Masory

Research Engineer,
Gould Laboratories,
Rolling Meadow, Ill. 60008,
Assoc. Mem. ASME

Y. Koren

Goebel Professor,
Department of Mechanical
Engineering,
Center for Robotics
and Integrated Manufacturing,
University of Michigan,
Ann Arbor, Mich. 48109
Mem. ASME

Reference-Word Circular Interpolators for CNC Systems

Interpolation techniques for CNC manufacturing systems are either of the Reference-Pulse or Reference-Word type. Reference-Pulse interpolators have been discussed in a previous paper [2]. The present paper provides an analysis of a selected number of Reference-Word circular interpolation techniques that are useful in CNC systems. These interpolators function in an on-line iterative mode and generate binary words which are supplied as references to the control loops in Sampled-Data CNC systems. The various methods are evaluated in terms of accuracy, maximum permitted radius, and number of required iterations to complete a circular arc.

Introduction

The present paper is the last in a series of three papers dealing with the design concepts of computer numerical control (CNC) systems [1, 2]. The first paper [1] presented two principal structures of control systems denoted as Reference-Pulse and Sampled-Data systems. With the first technique the computer produces a sequence of reference pulses for each axis of motion, each pulse generating a motion of one basic length-unit (BLU) of axis travel. These pulses can actuate a stepping motor in an open-loop system, or be fed as a reference to a closed-loop hardware system. With the Sampled-Data technique, the control loop is closed by software through the computer itself, as shown in Fig. 1. The control program compares a reference binary word with the feedback signal to determine the position error. The error is fed at fixed time intervals to a digital-to-analog converter (DAC), which in turn supplies a voltage proportional to the required axis velocity.

Both of these techniques require distinct interpolation routines to generate their corresponding references: pulses or binary words. The interpolator routine coordinates the motion along the machine axes, which are separately driven, to generate the required machining path. The machining path is usually obtained from a combination of linear and circular segments, and accordingly the control program contains both linear and circular interpolation subroutines. The interpolator routine functions in an on-line mode and is usually written in assembly language in order to improve computing speed. Although linear interpolators can be easily written in assembly language, difficulties arise in assembly language programming of circular interpolators which necessitates the on-line solution of a second order equation.

The previous paper [2] in this series was concerned with the evaluation of circular interpolators for Reference-Pulse CNC systems. All Reference-Pulse interpolators are based on an iterative technique controlled by an interrupt clock. At each

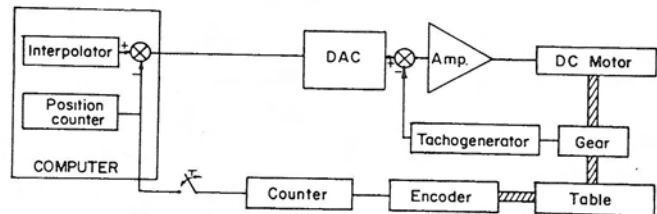


Fig. 1 A single-axis sampled-data system

interrupt, a single iteration of the interpolation routine is executed, which in turn can provide an output pulse that advances the corresponding machine axis by one BLU. Therefore, the maximum attainable feedrate, or axis velocity in BLUs/s, is inversely proportional to the execution time of a single iteration. It has been shown that the appropriate Reference-Pulse interpolation technique for machine tool systems is the software DDA [3], since the uniformity of the feedrate along the circular path is obtained only with this method. Using a PDP-11 as the CNC controller, an interpolation iteration time of $52 \mu\text{s}$ has been obtained [2], which allows a maximum velocity of 19,230 BLUs/s. A BLU might be on the order of $10 \mu\text{m}$ in a typical machine tool system, resulting in a maximum feedrate of about 11 m/min, which is adequate for machining applications. However, modern CNC systems employ a single axial drive motor for both idle motions and machining, and in these systems a velocity of 11 m/min is considered to be too low for idle motion.

In contrast, the maximum velocity in Sampled-Data systems is not limited by the computer. In these systems the circular interpolation is based upon approximating a circle with straight line segments. For each segment the interpolator generates a reference word proportional to the local axial velocities which are transmitted to the corresponding software loop-comparators of the control axes. The present paper evaluates Reference-Word circular interpolator techniques which can be applied to manufacturing systems permitting the simultaneous operation of two axes-of-motion.

The Concept of Reference-Word Interpolators

In circular interpolation, the simultaneous motion of two

Contributed by the Production Engineering Division for publication in the JOURNAL OF ENGINEERING FOR INDUSTRY and presented at the 1981 Winter Annual Meeting, November 15-17, 1981, Washington, D.C., of the AMERICAN SOCIETY OF MECHANICAL ENGINEERS. Revised manuscript received at ASME Headquarters June 15, 1982.

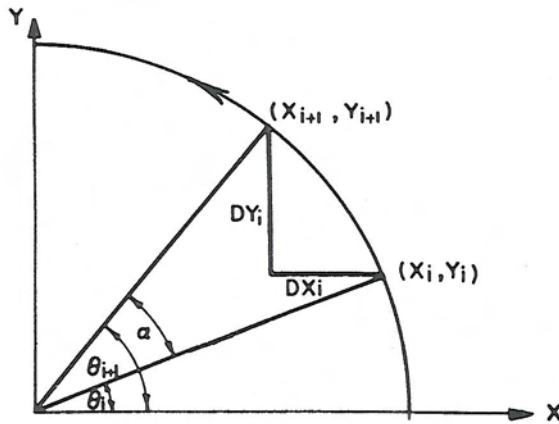


Fig. 2 Definitions for the reference-word technique

axes generates a circular arc at a constant tangential velocity, or feedrate, V_0 . The axial velocities satisfy the following equations:

$$\begin{aligned} V_x(t) &= V_0 \sin \theta(t) \\ V_y(t) &= V_0 \cos \theta(t) \end{aligned} \quad (1)$$

where

$$\theta(t) = (V_0/R)t$$

and R is the radius of the circular arc.

The velocity components V_x and V_y are computed by the circular interpolator and are supplied as reference inputs to the computer closed-loops. The circle generated in this case is actually comprised of straight line segments. At the beginning of each segment the references are supplied by the interpolator and the end of the segment is located with the aid of a feedback signal. Increasing the number of these segments improves the accuracy of the generated circle, but increases the number of iterations, thus requiring more computer time. The optimal number of segments is the smallest one which maintains the path error within the required limit of one BLU.

Each iteration of the algorithm corresponds to an angle α , as illustrated in Fig. 2. The choice of the angle α depends on the interpolation method. All of these methods employ the difference equation:

$$\begin{aligned} \cos \theta(i+1) &= A \cos \theta(i) - B \sin \theta(i) \\ \sin \theta(i+1) &= A \sin \theta(i) + B \cos \theta(i) \end{aligned} \quad (2)$$

where the coefficients A and B are given by:

$$A = \cos \alpha \quad B = \sin \alpha \quad (3)$$

and

$$\theta(i+1) = \theta(i) + \alpha \quad (4)$$

The corresponding segment is terminated at the point $X(i+1)$, $Y(i+1)$ which is approximated by:

$$\begin{aligned} X(i+1) &= R(i) \cos \theta(i+1) \\ Y(i+1) &= R(i) \sin \theta(i+1) \end{aligned} \quad (5)$$

Substituting equation (2) into (5) yields:

$$\begin{aligned} X(i+1) &= AX(i) - BY(i) \\ Y(i+1) &= AY(i) + BX(i) \end{aligned} \quad (6)$$

Equation (6) is the basic relationship which permits the calculation of a successive point based on the present one.

The main differences between the various interpolation methods are in the determination of α and in approximation of the coefficients A and B in equation (3). Once the angle α is chosen, the interpolator routine proceeds, for each iteration, as follows:

1 At each point $X(i)$, $Y(i)$ the interpolator calculates the

coordinates of the successive point $X(i+1)$, $Y(i+1)$ according to equation (6). The segment lengths are:

$$\begin{aligned} DX(i) &= X(i+1) - X(i) = (A-1)X(i) - BY(i) \\ DY(i) &= Y(i+1) - Y(i) = (A-1)Y(i) + BX(i) \end{aligned} \quad (7)$$

and the corresponding velocities are:

$$\begin{aligned} V_x(i) &= V_0 DX(i) / DS(i) \\ V_y(i) &= V_0 DY(i) / DS(i) \end{aligned} \quad (8)$$

where

$$DS(i) = \sqrt{DX^2(i) + DY^2(i)}$$

2 The values obtained from equation (7) are the incremental positions, and those of equation (8) are the velocities, or the reference-words, for the present segment. These values are supplied to the software control loops.

3 Upon completion of the segment, the routine increments the coordinates $X(i)$ and $Y(i)$.

Since the angle α is relatively small, the chord length DS can be approximated by its arc length $R\alpha$ and the calculation of the velocities V_x and V_y can be simplified to:

$$\begin{aligned} V_x(i) &= K DX(i) \\ V_y(i) &= K DY(i) \end{aligned} \quad (9)$$

where $K = V_0/R\alpha$. The parameter K is a constant, which is calculated only once for each circle, and consequently the reference words are actually proportional to the segment lengths.

Approximation of a circle by straight line segments causes two types of errors as illustrated in Fig. 3:

(a) Radial error ER , due to the truncation effects:

$$ER(i) = R(i) - R = \sqrt{X^2(i) + Y^2(i)} - R \quad (10)$$

where R is the required radius.

(b) Chord height error EH defined by:

$$EH(i) = R - R(i) \cos(\alpha/2) \quad (11)$$

Equation (10) is explicitly obtained using the approximated values of A and B , which causes the truncation error in the radius. Since the algorithms use an iterative technique, this error is accumulated with time. The effect of the truncation error ER at the i th iteration can be approximated by (see Appendix 1):

$$ER(i) = i(C-1)R \quad (12)$$

where

$$C = \sqrt{A^2 + B^2}$$

Equation (12) shows how the truncation error increases with the number of iterations. The largest circular arc produced by a single instruction in NC systems is a quarter of a circle. The compatible number of iterations is:

$$N = \pi/2\alpha \quad (13)$$

Upon completing this number of steps the error ER reaches its maximum value:

$$ER_{\max} = (\pi/2\alpha)(C-1)R \quad (14)$$

By contrast, the error EH is not cumulative. Using the formula

$$\cos(\alpha/2) = \sqrt{(1 + \cos \alpha)/2} = \sqrt{(1 + A)/2} \quad (15)$$

in equation (11), yields:

$$EH(i) = R - R(i) \sqrt{(1 + A)/2} \quad (16)$$

The angle α is selected so that either the error ER or EH

reaches a maximum value of one BLU. In addition to the angle α , the interpolator is given the coordinates of the initial point $[X(0), Y(0)]$ and the required feedrate V_0 . These parameters are obtained from the part-program for each circular arc.

Several interpolators applying the reference-word method are now discussed and their relative features are compared.

Euler Method

The Euler method is based upon a first-order Taylor series expansion to equation (3) which provides the simplest approximation:

$$A=1 \quad ; \quad B=\alpha \quad (17)$$

The basic equations of the algorithm are obtained by substituting for A and B in equation (7):

$$\begin{aligned} DX(i) &= -\alpha Y(i) \\ DY(i) &= \alpha X(i) \end{aligned} \quad (18)$$

Substitution of equation (18) in (9) gives the required velocities:

$$\begin{aligned} V_x(i) &= -(V_0/R)Y(i) \\ V_y(i) &= (V_0/R)X(i) \end{aligned} \quad (19)$$

Since the series expansion is truncated after the first term, the accuracy of the Euler method depends mainly on this truncation error ER rather than on the chord error EH. The maximum error is obtained by substituting equation (17) into (14):

$$ER_{\max} = (\pi/2\alpha)(\sqrt{1+\alpha^2} - 1)R \quad (20)$$

For small α , equation (20) can be approximated by:

$$ER_{\max} = (\pi/4)\alpha R \quad (21)$$

which allows for calculating the value of α . To satisfy the accuracy requirement, the maximum permitted error is one BLU and the corresponding angle is:

$$\alpha = 4/(\pi R) \quad (22)$$

Consequently, the number of iterations required to complete a quarter of a circle is obtained from equation (13):

$$N = \pi^2 R/8 \quad (23)$$

Improved Euler Method

The improved Euler method (IEM) [3] also uses the approximation (17) but it applies the calculated value of $X(i+1)$ to determine the present value of $Y(i+1)$. With this modification, equation (6) becomes:

$$\begin{aligned} X(i+1) &= AX(i) - BY(i) \\ Y(i+1) &= AY(i) + BX(i+1) \end{aligned} \quad (24)$$

and with substitution of A and B from (17) the main equations of the (IEM) algorithm are obtained:

$$\begin{aligned} X(i+1) &= X(i) - \alpha Y(i) \\ Y(i+1) &= Y(i) + \alpha X(i+1) \end{aligned} \quad (25)$$

The segment lengths as defined in equation (7) are:

$$\begin{aligned} DX(i) &= -\alpha Y(i) \\ DY(i) &= \alpha X(i+1) \end{aligned} \quad (26)$$

The improvement as compared with the previous method is illustrated by deriving the explicit expression for $Y(i+1)$ in the second equation of (25):

$$\begin{aligned} X(i+1) &= X(i) - \alpha Y(i) \\ Y(i+1) &= (1 - \alpha^2)Y(i) + \alpha X(i) \end{aligned} \quad (27)$$

Table 1 Maximum radius in IEM

n [bits]	q	p	R_m [BLUs]
15	8	7	255
23	13	10	4095
31	16	15	32767

The actual average A which is used in equation (27) is given by:

$$\frac{1}{2} [1 + (1 - \alpha^2)] = 1 - \frac{1}{2} \alpha^2 \quad (28)$$

rather than the value of 1 which is used for approximation of $\cos\alpha$ in the Euler method. Therefore, a better approximation of the coefficient A is achieved with the IEM without any additional computational effort.

The error ER reaches its maximum value for particular iteration that satisfies the condition (see Appendix 2):

$$X(i) = (1 + \alpha)Y(i) \quad (29)$$

or at angle of $\theta \approx \pi/4$ and the corresponding error is:

$$ER_{\max} = \frac{\alpha}{4} R \quad (30)$$

Since ER should not exceed one unit

$$\alpha = 4/R \quad (31)$$

The number of iterations required to produce a quarter of a circle as obtained from equation (13) is:

$$N = \pi R/8 \quad (32)$$

which is less than with the Euler method by a factor of π (see equation (23)).

Although the algorithms of the Euler and Improved Euler methods are relatively simple, they cannot be implemented with basic assembler instructions such as ADD and SHIFT. The computational difficulties arise in calculating the angle α and in performing multiplications αX and αY . With the Euler method, part of the difficulty in calculating the angle α is with using the factor π (equation (22)), which is not needed with the IEM (equation (31)). The second difficulty can be overcome by expressing the angle α in powers of 2, i.e., $\alpha = 2^{-m}$, thereby permitting the use of SHIFT operations rather than multiplications. In this case the angle α is calculated by:

$$\alpha = 2^{-m} \leq 4/R$$

or

$$2^{m+2} \geq R \quad (33)$$

The power m is the smallest integer which satisfies equation (33) and is obtained by SHIFT operations. Thus, a complete implementation of the IEM algorithm is achieved by basic assembler instructions, without the necessity for either a floating-point unit (FPU) or multiplication operations.

In spite of the attractive features of the IEM, special care is needed to avoid introducing a new truncation error that can arise from the right-shifts used in αX and αY . To avoid the truncation error, the n -bit word is divided into two parts. One part is q bits long and stores the radius. The other part is p bits and stores the error. In the IEM, the maximum ER (of one unit) occurs at $\theta = \pi/4$ or when $1/2N$ iterations have been completed. Thus, the condition:

$$\frac{1}{2} N = 2^p \quad (34)$$

guarantees that the truncation error is less than one unit. The relationship between the maximum permitted radius R_m and the word length n can be determined through the equation:

$$n = q + p = \log_2 R_m + \log_2 \left(\frac{1}{2} N \right) \quad (35)$$

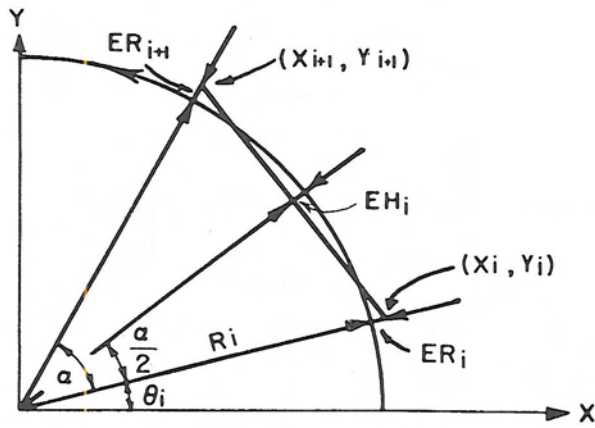


Fig. 3 Errors in the generated circle

Substituting for N from equation (32) gives:

$$n = 2\log_2 R_m + \log_2 \pi - 4 \quad (36)$$

and since $R_m = 2^q$

$$q = \frac{1}{2} n + 1.2 \quad (37)$$

Table 1 gives the maximum allowable radius in BLU's for typical length of computer words (one bit is reserved as a sign bit):

In practice only the last line in Table 1 would give sufficiently large radii in CNC systems. Therefore, a double- or even triple-precision word technique will be required in the control computer. Note that since in the Euler method the number of iterations are greater than in the IEM and the maximum error appears at the end of the arc, the equivalent formula to (37) is:

$$q = \frac{1}{2} n - 0.15 \quad (38)$$

which means that the maximum permitted radius in practice is even smaller, and the Euler method is useless for CNC applications.

Taylor Method

For the Taylor method [4] the coefficients A and B in equation (3) are evaluated by the truncated series:

$$A = 1 - \frac{1}{2} \alpha^2; B = \alpha \quad (39)$$

Using these values, the basic equations of the algorithm are derived. The corresponding segment lengths are obtained from equation (7):

$$DX(i) = -\frac{1}{2} \alpha^2 X(i) - \alpha Y(i) \quad (40)$$

$$DY(i) = -\frac{1}{2} \alpha^2 Y(i) + \alpha X(i)$$

and the velocities obtained from equation (9):

$$V_x(i) = -(V_0/R) \left[\frac{1}{2} \alpha X(i) + Y(i) \right] \quad (41)$$

$$V_y(i) = (V_0/R) \left[-\frac{1}{2} \alpha Y(i) + X(i) \right]$$

The maximum value of the error ER is calculated according to equation (14):

$$ER_{\max} = \frac{\pi}{2\alpha} \left(\sqrt{1 + \frac{1}{4} \alpha^4} - 1 \right) R \approx \pi R \alpha^3 / 16 \quad (42)$$

Table 2 Maximum radius in Taylor method

n [bits]	q	p	R_m [BLUs]
15	10	5	1023
23	15	8	32767
31	21	10	2097151

The error EH in equation (11) is obtained by substitution for A , which yields:

$$EH(i) = R - R(i) \sqrt{1 - \frac{1}{4} \alpha^2} = R - R(i) + (\alpha^2/8)R(i) \quad (43)$$

Since the error caused by the truncated series (39) leads to an increased radius with each step, the maximum value of EH is obtained at the first iteration, for $R(i) = R$, which yields:

$$EH_{\max} = R \alpha^2 / 8 \quad (44)$$

A comparison of two errors in equations (42) and (44) shows that

$$ER_{\max} = \frac{\pi \alpha}{2} EH_{\max} \quad (45)$$

which means that for an angle α which satisfies $\alpha < 2/\pi$, the error EH is the dominant one. In this case the angle α is determined from equation (44). For EH_{\max} equal to one BLU

$$\alpha = \sqrt{8/R} \quad (46)$$

In this case $\alpha < 2/\pi$ is satisfied for $R > 20$ BLUs, which covers virtually all practical systems (e.g., if 1BLU = 10 μ m, $R > 0.2$ mm). The number of iterations to complete a quarter of a circle is:

$$N = \pi/2\alpha = (\pi/4)\sqrt{R/2} \quad (47)$$

The algorithm given by equations (40) and (41) requires a determination of α and α^2 . Expressing these values in power of 2 avoids the use of floating-point unit [5]:

$$\alpha = 2^{-m} < \sqrt{8/R} \quad (48)$$

or

$$2^{2m+3} \geq R \quad (49)$$

Again, the power m is determined by SHIFT operations, and all multiplication instructions in the algorithm are executed in the same way. Therefore, the Taylor method can be implemented by using basic assembler instructions.

The relationship between the maximum radius and the word-length for the Taylor method is:

$$n = q + p = \log_2 R_m + \log_2 N = \log_2 R_m + \log_2 [(\pi/4)\sqrt{R_m/2}] \quad (50)$$

and since $R = 2^q$:

$$q = \frac{2}{3} n + 0.56 \quad (51)$$

Examples illustrating the dependence of the maximum radius on the word length are given in Table 2. The last two cases in Table 2 are practical for CNC systems.

In reference [4] the Taylor method algorithm was applied to a CNC system controlled by a PDP-8 computer using a triple-precision word. Successive points along the circular arc were computed at the rate of 1.4 millisecond per point. The interpolator occupied 600 (12 bit) words of core storage.

Tustin Method

The Tustin method [6] is based on an approximate

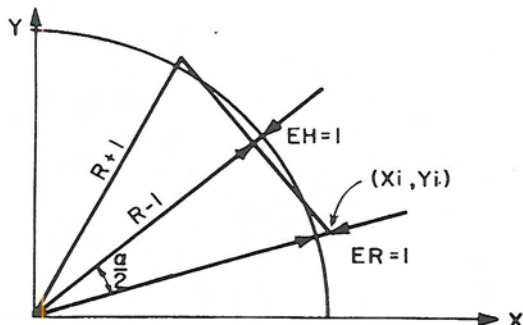


Fig. 4 Circle generation by the Improved Tustin Method

relationship between the derivative operator s and the discrete variable z :

$$s = \frac{2}{T} \left(\frac{z-1}{z+1} \right) \quad (52)$$

Using the Tustin approximation yields the following relations for $\cos \alpha$ and $\sin \alpha$, or A and B :

$$A = \frac{1 - (\alpha/2)^2}{1 + (\alpha/2)^2}; \quad B = \frac{\alpha}{1 + (\alpha/2)^2} \quad (53)$$

These approximations can be used to generate circular interpolation for CNC systems. Substituting these values of A and B into equations (7) and (9) gives the basic equations of the proposed algorithm:

$$DX(i) = -\frac{1}{1 + (\alpha/2)^2} \left[\frac{\alpha^2}{2} X(i) + \alpha Y(i) \right] \quad (54)$$

$$DY(i) = \frac{1}{1 + (\alpha/2)^2} \left[-\frac{\alpha^2}{2} Y(i) + \alpha X(i) \right]$$

$$V_x(i) = -\frac{V_0}{R[1 + (\alpha/2)^2]} \left[\frac{\alpha}{2} X(i) + Y(i) \right] \quad (55)$$

$$V_y(i) = \frac{V_0}{R[1 + (\alpha/2)^2]} \left[-\frac{\alpha}{2} Y(i) + X(i) \right]$$

The truncation error is found by substituting equation (53) into (12) which shows that:

$$ER(i) = 0 \quad (56)$$

Since the error ER is identically zero, the angle α is prescribed by the error EH given in equation (16). For $R(i) = R$:

$$EH = R - \frac{R}{\sqrt{1 + (\alpha/2)^2}} \quad (57)$$

which can be approximated for small α by:

$$EH = \frac{\alpha^2}{\alpha^2 + 8} R \quad (58)$$

The angle α is derived by equating the error EH to one unit:

$$\alpha = \sqrt{8/(R-1)} \approx \sqrt{8/R} \quad (59)$$

which is identical to equation (46).

Improved Tustin Method

The Tustin method yields an ER of zero. From a practical point of view in manufacturing systems we choose the maximum error equal to one BLU. Therefore the method can be improved by increasing the angle α [7]. From Fig. 4 the angle α is evaluated using equation (15):

$$\frac{R+1}{R-1} = \sqrt{2/(1+A)} = \sqrt{1 + (\alpha/2)^2} = 1 + \frac{\alpha^2}{8} \quad (61)$$

Table 3 Simulation results for $R = 10,000$ BLU's

Method	N	Simulation		Practical	
		ER _{max}	EH _{max}	ER _{max}	EH _{max}
Euler	12331	1.008	1.008	1.500	1.500
IEM	3928	1.002	1.002	1.652	1.652
Taylor	56	0.071	0.988	0.633	1.428
Tustin	56	1.016	1.016	0.681	1.681
ITM	40	1.029	0.999	1.662	1.405

Table 4 Simulation results for $R = 250$ BLUs

Method	Type	N	ER _{max}	EH _{max}
IEM	FPU	99	1.557	1.549
IEM	ASL	102	1.103	1.103
Taylor	FPU	9	0.529	0.999
Taylor	ASL	13	0.953	0.953

Table 5 Features of reference-word interpolators

Method	Programming	α	N
Euler	FPU	$4/\pi R$	$\pi^2 R/8$
IEM	FPU or ASL	$4/R$	$\pi R/8$
Taylor	FPU or ASL	$\sqrt{8/R}$	$\frac{\pi}{4} \sqrt{R/2}$
Tustin	FPU	$\sqrt{8/R}$	$\frac{\pi}{4} \sqrt{R/2}$
ITM	FPU	$4/\sqrt{R}$	$\frac{\pi}{8} \sqrt{R}$

so that the angle α is:

$$\alpha = \sqrt{16/(R-1)} = 4/\sqrt{R} \quad (62)$$

Therefore, the angle α is increased by a factor of $\sqrt{2}$ decreasing the number of iterations to:

$$N = \frac{\pi}{8} \sqrt{R} \quad (63)$$

Obviously the Tustin methods require considerably more calculations than the previous methods. Basic assembler instructions are insufficient to handle the routine and an FPU is a necessity.

Comparison and Discussion

The interpolation methods requiring an FPU were simulated for $R = 10000$ BLU units to produce a quarter of a circle. The results are presented in Table 3, where N is the number of iterations and the errors are given in BLUs.

The errors ER_{max} and EH_{max} , calculated during the simulation of the methods, are listed in the third and fourth columns of Table 3. In a real system, however, the DACs and the position counters can only use integer numbers, so the reference word must be an integer. This means that DX , DY , V_x , and V_y must be rounded off to the closest integer value which increases the error. The maximum errors ER_{max} and EH_{max} for a real system are listed in the fifth and sixth columns of Table 3. In this example, for real systems, the interpolation methods generate errors in the range of 1 to 2 BLUs. In order to limit the error to 1 BLU the angle α should be decreased. It can be seen that the Improved Tustin Method will generate an arc with the fewest number of iterations.

In order to compare assembly language (ASL) with the floating-point unit (FPU) methods, a simulation was performed for producing a 90 degree arc with a radius of 250 units. The results shown in Table 4 indicate that for an

algorithm programmed with assembly instructions, the Taylor method is the best. The number of iterations required to complete the arc by this method is considerably smaller than the number needed with IEM, while the errors are always within the required tolerance of one BLU.

Finally, the main features of the various algorithms have been summarized in Table 5.

The interpolators of Table 5 which require an FPU will require more computer storage and have reduced computer speed than those implemented with ASL. From Table 5 it can be seen that if an FPU is available, the Improved Tustin Method (ITM), which has been introduced in the present paper, is the favorite. This method produces a circle using the fewest iterations and guarantees the required accuracy. The computation of the parameters A and B in equation (53) does not cause any difficulties since an FPU is used.

If an FPU is not available, or if implementation of assembly language is preferred, then the Taylor method [4, 5] should be applied. With this method the maximum allowable radius is much larger than with the IEM (see Tables 1 and 2), and the number of iterations required to complete the arc is smaller by a factor of $\sqrt{R/2}$.

Acknowledgment

The authors would like to thank Professors S. Malkin and G. Ulsoy for their helpful comments.

References

- 1 Koren, Y., "Design of Computer Control for Manufacturing Systems," *ASME JOURNAL OF ENGINEERING FOR INDUSTRY*, Vol. 101, No. 3, Aug. 1979, pp. 326-332.
- 2 Koren, Y., and Masory, O., "Reference-Pulse Circular Interpolators for CNC Systems," *ASME JOURNAL OF ENGINEERING FOR INDUSTRY*, Vol. 103, No. 1, Feb. 1981, pp. 131-136.
- 3 Koren, Y., "Interpolator for a CNC System," *IEEE Trans. on Computers*, Vol. 25, No. 1, Jan. 1976, pp. 32-37.
- 4 Bergren, C., "A Simple Algorithm for Circular Interpolation," *Control Engineering*, Sept. 1971, pp. 57-59.
- 5 Musse, J., Veron, M., Lepage, F., and Drapier, J., "High Performance Tool Path Interpolator for CNC System," *CIRP Manufacturing Systems*, Vol. 7, No. 1, 1978. (Presented in the 9th CIRP Int. Sem. on Manufacturing Systems.)
- 6 Cadzow, J. A., and Martens, H. R., *Discrete-Time and Computer Control Systems*, Chapter 9, Prentice-Hall, New Jersey, 1970.
- 7 Koren, Y., *Computer Control of Manufacturing Systems*, Chapter 5, McGraw-Hill, New York, 1983.

APPENDIX 1

Summing the squares of both sides of equation (6) gives:

$$R^2(i+1) = C^2 R^2(i) \quad (\text{A1-1})$$

where

$$C^2 = A^2 + B^2 \quad (\text{A1-2})$$

By successively using equation (A1-1), as the interpolation proceeds, assuming that the initial point lies on the circle [$R(0) = R$], one obtains:

$$R(i) = C^i R \quad (\text{A1-3})$$

The radial error ER in the i th step is determined from equation (10) and equation (A1-3):

$$\text{ER}(i) = (C^i - 1)R \quad (\text{A1-4})$$

Since the value of C is close to 1, equation (A1-4) is approximated by:

$$\text{ER}(i) = i(C - 1)R \quad (\text{A1-5})$$

APPENDIX 2

Defining:

$$\begin{aligned} \Delta R(i) &= R(i+1) - R(i) \\ &= \sqrt{X^2(i+1) + Y^2(i+1)} - \sqrt{X^2(i) + Y^2(i)} \end{aligned} \quad (\text{A2-1})$$

The radial error ER at the j th step is derived by summing the incremental errors $\Delta R(i)$, or by integrating along the arc s :

$$\text{ER}(j) = \sum_{i=0}^j \Delta R(i) = \int \Delta R \cdot ds \quad (\text{A2-2})$$

The maximum error ER_{\max} is obtained by equating the first derivative of equation (A2-2) to zero:

$$\frac{d\text{ER}}{ds} = \Delta R(i) = 0 \quad (\text{A2-3})$$

Namely, at $R(i+1) = R(i)$

Using the definition of equation (A2-1) we see that the maximum radial error occurs when

$$X^2(i+1) + Y^2(i+1) = X^2(i) + Y^2(i) \quad (\text{A2-4})$$

Substitution of equation (25) into (A2-4) yields

$$2[X(i) - X(i+1)]Y(i)/\alpha = X^2(i+1) + Y^2(i) \quad (\text{A2-5})$$

Substituting for $X(i+1)$ from equation (24) into the left-side term of equation (A2-5) gives $Y(i) = X(i+1)$, or

$$X(i) = Y(i) + \alpha Y(i) \quad (\text{A2-6})$$