Sensors & Transducers

© 2014 by IFSA Publishing, S. L.
http://www.sensorsportal.com

An Improved Minimum Error Interpolator of CNC for General Curves Based on FPGA

[1, 2]Jiye HUANG, [1] Jiaxiang LOU, [1] Zhanxiong WU, [1] Mingyu GAO
[1] Hangzhou Dianzi University, Hangzhou, 310018, China
[2] E-mail: hjynet@hdu.edu.cn

Received: 10 January 2014 /Accepted: 30 April 2014 /Published: 31 May 2014

Abstract: This paper presents an improved minimum error interpolation algorithm for general curves generation in computer numerical control (CNC). Compared with the conventional interpolation algorithms such as the By-Point Comparison method, the Minimum-Error method and the Digital Differential Analyzer (DDA) method, the proposed improved Minimum-Error interpolation algorithm can find a balance between accuracy and efficiency. The new algorithm is applicable for the curves of linear, circular, elliptical and parabolic. The proposed algorithm is realized on a field programmable gate array (FPGA) with Verilog HDL language, and simulated by the ModelSim software, and finally verified on a two-axis CNC lathe. The algorithm has the following advantages: firstly, the maximum interpolation error is only half of the minimum step-size; and secondly the computing time is only two clock cycles of the FPGA. Simulations and actual tests have proved that the high accuracy and efficiency of the algorithm, which shows that it is highly suited for real-time applications. *Copyright © 2014 IFSA Publishing, S. L.*

Keywords: Improved Minimum-Error Interpolation, FPGA, CNC.

1. Introduction

In computer numerical control (CNC) systems [1], the motion trajectory of a machining tool is achieved through interpolations, and the interpolation algorithm directly affects the processing speed and accuracy. The task of the interpolation algorithm is to calculate the coordinates of intermediate points between the start point and the end point of the outline, making the motion trajectory same as the given interpolation curve [2-5]. Because the real-time of the interpolation directly affects the turning speed of the CNC system and the final geometric accuracy of the parts, it is the core technology which directly influences the performance of the whole CNC system [6-8].

In present, the widely used interpolation algorithms are the By-Point Comparison algorithm, the Minimum-Error algorithm [9], and the Digital Differential Analyzer (DDA) based algorithm [10].

As to By-Point Comparison method, the current coordinate is compared with the theoretical trajectory to determine the direction of the next step. If current point is at the bottom (inside) of Z-axis, then next step is to move to the top (outside) of the given trajectory; similarly, if current point is at the top (outside) of the given trajectory, then next step is to move to the bottom (inside) of the given trajectory. By-Point Comparison is simple for calculation and realization t, but the problem is that the maximum error is close to one min-step size.

As to Minimum-Error method, the output has three mobile ways: the direction of X-axis, the direction of Y-axis and the diagonal direction (both X-axis and Y-axis). So it is just need to judge which point is the closest one to the given trajectory, then this direction is selected. The maximum error is just half of min-step size, but the method is too complex for MCU system.

As to DDA method, each axis accumulates a value to a register and this value is related to current coordinate position. When the accumulator register value is bigger than a set value which we called overflow, the corresponding axis output a pulse, the accumulator register subtracts the set value and the remainder is used for the next calculation. When calculate non-linear interpolation, DDA need to change the accumulated value to current coordinate position, which makes unavoidable rounding error.

In conclusion, because of the defect of low accuracy or low efficiency in By-Point Comparison, Min-Error and DDA, this paper proposed the Slightest Min-Error interpolation algorithm of linear, circular, elliptical and parabolic for MCU system.

Field programmable gate array (FPGA) is an advanced product on the basis of programmable devices such as PAL \ GAL \ EPLD. Compared with ASICs and PLDs, FPGA has the characteristics of easy to be modified, low cost, short design cycle, etc. Compared with traditional MCUs, the FPGA has the characteristics of real-time control, high speed, reliability, parallelism. With these advantages, the FPGA is quite suited for reconfigurable systems such as computer numerical control (CNC) systems.

2. Theory of Slightest Min-error

As shown in Fig. 1, the coordinate of the current point is *P(x_i, y_i)*. There are three mobile ways: the diagonal direction (both X-axis and Y-axis), the direction of X-axis and the direction of Y-axis, denoted as point $A(x_i+1, y_i+1)$, point $B(x_i+1, y_i)$ and point $C(x_i, y_i+1)$. Because of the different mobile direction and different quadrants, we agree as follows: point A is the diagonal direction; point B is the direction as the bottom (inside or closer to the origin) of the curve and point C is another direction. The midpoint of the straight line AC marked as $M(x_i+\frac{1}{2}, y_i+1)$ and the midpoint of the straight line AB as $N(x_i+1, y_i+\frac{1}{2})$.

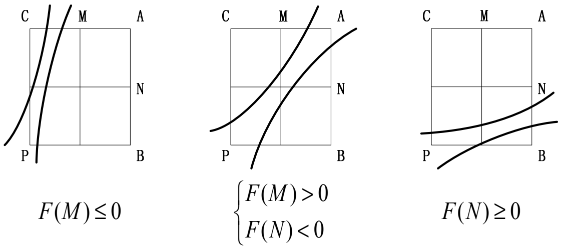


Fig. 1. Process of Slightest Min-Error Interpolation.

In Fig. 1, F is a deviation judgment formula, which is determined by the interpolation curve. Here this formula can be simply understood as the coordinates of the point error of plus or minus.

According to the positive or negative value of deviation judgment formula in the mid-point M and N, we judge the mobile direction:

When $F(M)\le 0$, point $C(x_i, y_i+1)$ is closer to the curve. So select the direction of Y-axis.

When $\begin{cases}F(M)>0\\F(N)<0\end{cases}$, point $A(x_i+1, y_i+1)$ is closer to the curve. So select diagonal direction (both X-axis and Y-axis).

When $F(N)\ge 0$, point $B(x_i+1, y_i)$ is closer to the curve. So select the direction of X-axis.

In every step the result of deviation judgment formula needs to be recalculated, which is used to determine the mobile direction of next coordinates.

3. Interpolation of General Curves

Next the new method was deduced to general curves, like line, circle, ellipse and parabola, which the most important is the deviation judgment formula.

3.1. Line Interpolation

3.1.1. Condition Description

Known conditions: the start point is the origin (0, 0); the end coordinates is (X_e, Y_e), which is in the first quadrant.

In the Fig. 2, the coordinate of different points as follows: $P(x_i, y_i)$, $A(x_i+1y_i+1)$, $B(x_i+1, y_i)$, $C(x_i, y_i+1)$, $M(x_i+\frac{1}{2}, y_i+1)$ and $N(x_i+1, y_i+\frac{1}{2})$.

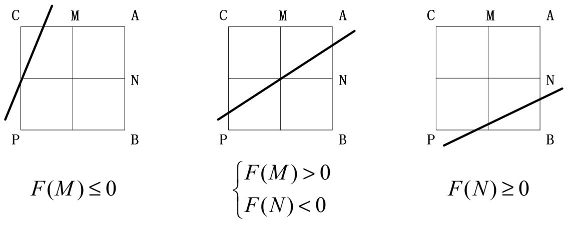


Fig. 2. Linear interpolation schematic.

3.1.2. Formula Derivation

Linear equation:

$$y=\frac{Y_e}{X_e}x, \tag{1}$$

The deviation judgment formula:

$$F = X_e y - Y_e x\,, \tag{2}$$

Substitute P , M and N into the formula:

$$F(\mathrm{P}) = F_i = X_e y_i - Y_e x_i\,, \tag{3}$$

$$\begin{aligned} F(M) &= X_e(y_i+1) - Y_e(x_i+\frac{1}{2}) \\ &= F_i + X_e - \frac{1}{2}Y_e \end{aligned} \tag{4}$$

$$\begin{aligned} F(N) &= X_e(y_i+\frac{1}{2}) - Y_e(x_i+1) \\ &= F_i + \frac{1}{2}X_e - Y_e \end{aligned} \tag{5}$$

Combined with Fig. 2 and the theory of slightest min-error, derive as follows:

When *F(M)≤0*, the point *C(x_i, y_i+1)* is closer to the curve. So the direction of Y-axis was selected. And the value of deviation judgment formula was updated:

$$F_{\mathrm{i}+1} = X_e(y_i+1) - Y_e x_i = F_{\mathrm{i}} + X_e \tag{6}$$

When $\begin{cases} F(M) > 0 \\ F(N) < 0 \end{cases}$, the point *A*($x_i$+1, y_i+1) is closer to the curve. So the diagonal direction was selected. And the value of deviation judgment formula was updated:

$$\begin{aligned} F_{\mathrm{i}+1} &= X_e(y_i+1) - Y_e(x_i+1) \\ &= F_{\mathrm{i}} + X_e - Y_e \end{aligned} \tag{7}$$

When *F(N)≥0*, the point *B*(x_i+1, y_i) is closer to the curve. So the direction of X-axis is selected. And the value of deviation judgment formula was updated:

$$F_{\mathrm{i}+1} = X_e y_{\mathrm{i}} - Y_e(x_i+1) = F_{\mathrm{i}} - Y_e \tag{8}$$

Finally, because the start point is on the line, obtain the initial value of deviation judgment formula:

$$F_0 = 0 \tag{9}$$

3.2. Circular Interpolation

3.2.1. Condition Description

Known conditions: the center of the circle is the point (0, 0); starting point is (0, *R*) (*R* is the radius of circular); and end point is (*R*, 0). And the interpolation direction is counterclockwise.

In the Fig. 3, the coordinate of different points was as follows: *P*(x_i, y_i), *A*(x_i+1, y_i-1), *B*(x_i, y_i-1), *C*(x_i+1, y_i), *M*(x_i+1, y_i-1/2) and *N*(x_i+1/2, y_i-1).

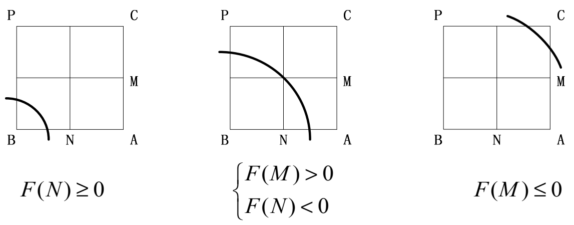


Fig. 3. Circular interpolation schematic.

3.2.2. Formula Derivation

Circle equation is:

$$x^2 + y^2 = R^2 \tag{10}$$

The deviation judgment formula is:

$$F = x^2 + y^2 - R^2 \tag{11}$$

Substitute P , M and N into the formula:

$$F(\mathrm{P}) = F_i = x_i^2 + y_i^2 - R^2 \tag{12}$$

$$\begin{aligned} F(M) &= (x_i+1)^2 + (y_i-\frac{1}{2})^2 - R^2 \\ &= F_i + 2x_i - y_i + \frac{5}{4} \end{aligned} \tag{13}$$

$$\begin{aligned} F(N) &= (x_i+\frac{1}{2})^2 + (y_i-1)^2 - R^2 \\ &= F_i + x_i - 2y_i + \frac{5}{4} \end{aligned} \tag{14}$$

Due to the value 5/4, which is not conductive for calculation, multiply both sides of the equation with the value 4.

$$4F(M) = 4F_i + 8x_i - 4y_i + 5 \tag{15}$$

$$4F(N) = 4F_i + 4x_i - 8y_i + 5 \tag{16}$$

As to the actual calculation, we use formula 4*F*(*M*) and 4*F*(*N*), because whether multiplied by 4 or not, they are the same sign. Combined with Fig. 3 and the theory of slightest min-error, derive as follows:

When *F(M)≤0*, point *C*(x_i+1, y_i) is closer to the curve. So select the direction of Y-axis. And recalculate the value of deviation judgment formula:

$$\begin{aligned} F_{\mathrm{i}+1} &= (x_i+1)^2 + y_i^2 - R^2 \\ &= F_{\mathrm{i}} + 2x_i + 1 \end{aligned} \tag{17}$$

When $\begin{cases} F(M) > 0 \\ F(N) < 0 \end{cases}$, point $A(x_i+1, y_i-1)$ is closer to the curve. So select diagonal direction (both X-axis and Y-axis). And recalculate the value of deviation judgment formula:

$$\begin{aligned} F_{\mathrm{i}+1} &= (x_i+1)^2 + (y_i-1)^2 - R^2 \\ &= F_{\mathrm{i}} + 2x_i - 2y_i + 2 \end{aligned} \tag{18}$$

When $F(N)\geq 0$, point $B(x_i, y_i-1)$, point is closer to the curve. So select the direction of X-axis. And recalculate the value of deviation judgment formula:

$$F_{\mathrm{i}+1} = x_i^2 + (y_i-1)^2 - R^2 = F_{\mathrm{i}} - 2y_i + 1 \tag{19}$$

Finally, because the start point is on the circle, obtain the initial value of deviation judgment formula:

$$F_0 = 0 \tag{20}$$

3.3. Elliptical Interpolation

3.3.1. Condition Description

Known conditions: the center of the circle is the origin (0, 0); starting point is (0, *b*); end coordinates is (*a*, 0), where *a* and *b* is major and minor axis of an Ellipse. And the interpolation direction is counterclockwise.

In the Fig. 4, the coordinate of different points as follows:

$P(x_i, y_i)$, $A(x_i+1, y_i-1)$, $B(x_i, y_i-1)$, $C(x_i+1, y_i)$, $M(x_i+1, y_i-1/2)$ and $N(x_i+1/2, y_i-1)$.

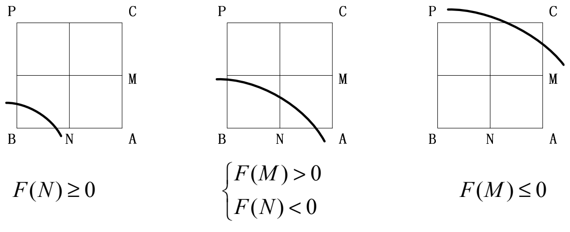


Fig. 4. Elliptical interpolation schematic.

3.3.2. Formula Derivation

Elliptical equation:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{21}$$

The deviation judgment formula:

$$F = b^2x^2 + a^2y^2 - a^2b^2 \tag{22}$$

Define two intermediate variables *s* and *t*:

$$s = b^2x^2 \tag{23}$$

$$t = a^2y^2 \tag{24}$$

Substitute *P, M* and *N* into the formula:

$$F(\mathrm{P}) = F_i = x_i^2 + y_i^2 - R^2 \tag{25}$$

$$s_i = b^2x_i^2 \tag{26}$$

$$t_i = a^2y_i^2 \tag{27}$$

$$\begin{aligned} F(M) &= b^2(x_i+1)^2 + a^2(y_i - \frac{1}{2})^2 - a^2b^2 \\ &= F_i + 2b^2x_i - a^2y_i + \frac{1}{4}a^2 + b^2 \\ &= F_i + 2s_i - t_i + \frac{1}{4}a^2 + b^2 \end{aligned} \tag{28}$$

$$\begin{aligned} F(N) &= b^2(x_i + \frac{1}{2})^2 + a^2(y_i-1)^2 - a^2b^2 \\ &= F_i + b^2x_i - 2a^2y_i + a^2 + \frac{1}{4}b^2 \\ &= F_i + s_i - 2t_i + a^2 + \frac{1}{4}b^2 \end{aligned} \tag{29}$$

Here the operations divided by 4 is no need to multiply with 4, because the value of a^2 and b^2 are so large that divided by 4 makes no effect. Combined with Fig. 4 and the theory of slightest min-error, derive as follows:

When $F(M)\leq 0$, point $C(x_i+1, y_i)$ is closer to the curve. So select the direction of Y-axis. And recalculate the value of deviation judgment formula:

$$\begin{aligned} F_{\mathrm{i}+1} &= b^2(x_i+1)^2 + a^2y_i^2 - a^2b^2 \\ &= F_{\mathrm{i}} + 2b^2x_i + b^2 \\ &= F_{\mathrm{i}} + 2s_i + b^2 \end{aligned} \tag{30}$$

$$s_{i+1} = s_i + b^2, \tag{31}$$

When $\begin{cases} F(M) > 0 \\ F(N) < 0 \end{cases}$, point $A(x_i+1, y_i-1)$ is closer to the curve. So select diagonal direction (both X-axis and Y-axis). And recalculate the value of deviation judgment formula:

$$\begin{aligned} F_{\mathrm{i}+1} &= b^2(x_i+1)^2 + a^2(y_i-1)^2 - a^2b^2 \\ &= F_{\mathrm{i}} + 2b^2x_i - 2a^2y_i + a^2 + b^2 \\ &= F_{\mathrm{i}} + 2s_i - 2t_i + a^2 + b^2 \end{aligned} \tag{32}$$

$$s_{i+1} = s_i + b^2 \tag{33}$$

$$t_{i+1} = t_i - a^2 \tag{34}$$

When $F(N) \geq 0$, point $B(x_i, y_i\text{-}1)$ is closer to the curve. So select the direction of X-axis. And recalculate the value of deviation judgment formula:

$$\begin{aligned} F_{\mathrm{i}+1} &= b^2 x_i^{\,2} + a^2(y_i - 1)^2 - a^2b^2 \\ &= F_{\mathrm{i}} - 2a^2 y_i + a^2 \\ &= F_{\mathrm{i}} - 2t_i + a^2 \end{aligned} \tag{35}$$

$$t_{i+1} = t_i - a^2 \tag{36}$$

Finally, because the start point is on the elliptical, obtain the initial value of deviation judgment formula:

$$F_0 = 0 \tag{37}$$

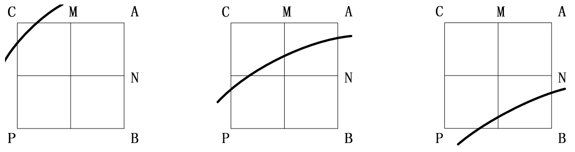
3.4. Parabola Interpolation

3.4.1. Condition Description

Known conditions: the vertex of the parabola is the origin (0, 0); starting point is the origin(0, 0); end coordinates is (2*p*,2*p*), where 2*p* is the coefficient of parabola. And the symmetry axis of the parabola is X-axis.

In the Fig. 5, the coordinate of different points as follows:

$P(x_i, y_i)$, $A(x_i{+}1, y_i{+}1)$, $B(x_i{+}1, y_i)$, $C(x_i, y_i{+}1)$, $M(x_i{+}1/2, y_i{+}1)$ and $N(x_i{+}1, y_i{+}1/2)$.



$F(M) \leq 0$ $\qquad$ $\begin{cases} F(M) > 0 \\ F(N) < 0 \end{cases}$ $\qquad$ $F(N) \geq 0$

Fig. 5. Parabola interpolation schematic.

3.4.2. Formula Derivation

Parabola equation:

$$y^2 = 2px \tag{38}$$

The deviation judgment formula:

$$F = y^2 - 2px \tag{39}$$

Substitute *P, M* and *N* into the formula:

$$F(\mathrm{P}) = F_i = y_i^{\,2} - 2px_i \tag{40}$$

$$\begin{aligned} F(M) &= (y_i + 1)^2 - 2p(x_i + \frac{1}{2}) \\ &= F_i + 2y_i - p + 1 \end{aligned} \tag{41}$$

$$\begin{aligned} F(N) &= (y_i + \frac{1}{2})^2 - 2p(x_i + 1)^2 \\ &= F_i + y_i - 2p + \frac{1}{4} \end{aligned} \tag{42}$$

Due to the value 5/4, which is not conductive for calculation, multiply both sides of the equation with the value 4.

$$4F(N) = 4F_i + 4y_i - 8p + 1 \tag{43}$$

As to the actual calculation, we use formula 4*F*(*N*), because whether multiplied by 4 or not, they are the same sign. Combined with Fig. 5 and the theory of slightest min-error, the follows were derived:

When $F(M) \leq 0$, point $C(x_i, y_i{+}1)$ is closer to the curve. So select the direction of Y-axis. And recalculate the value of deviation judgment formula:

$$F_{\mathrm{i}+1} = (y_i + 1)^2 - 2px_i = F_{\mathrm{i}} + 2y_i + 1 \tag{44}$$

When $\begin{cases} F(M) > 0 \\ F(N) < 0 \end{cases}$, point $A(x_i{+}1, y_i{+}1)$ is closer to the curve. So select diagonal direction (both X-axis and Y-axis). And recalculate the value of deviation judgment formula:

$$\begin{aligned} F_{\mathrm{i}+1} &= (y_i + 1)^2 - 2p(x_i + 1) \\ &= F_{\mathrm{i}} + 2y_i - 2p + 1 \end{aligned} \tag{45}$$

When $F(N) \geq 0$, point $B(x_i{+}1, y_i)$ is closer to the curve. So select the direction of X-axis. And recalculate the value of deviation judgment formula:

$$F_{\mathrm{i}+1} = y_i^{\,2} - 2p(x_i + 1) = F_{\mathrm{i}} - 2p \tag{46}$$

Finally, because the start point is on the Parabola, obtain the initial value of deviation judgment formula:

$$F_0 = 0 \tag{47}$$

4. Software Simulation

Slightest Min-Error interpolation algorithm is realized based on FPGA hardware platform by Verilog HDL. Here shows the simulation waveforms in ModelSim software, which displays 4 kinds of general cures.

Here are the generic signals in the simulation waveform, as follows:

Input signal: *clk* is the clock signal; *rst* is the reset signal; *Start_X*0, *Start_Z0, End_X*1 and *End_Z*1 represent the starting point and end point coordinates; *Syn* is a synchronization start signal;

Output signal: *X_Pwm_Out* and *Z_Pwm_Out* is the interpolation pulse output; *Busy* indicates when the interpolation is finish;

Intermediate signal: *Val_x* and Val_z show the current coordinate values; *Val_m*, *Val_n* and *Val_f* represent the deviation judgment formula $F(M)$, $F(N)$ and F_i.

4.1. Line Interpolation

Simulation waveform for linear interpolation modules is shown in Fig. 6. The starting point is the origin (0, 0) and the end coordinates is point $(3,5)$.

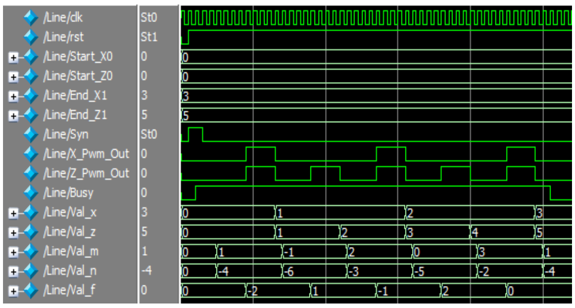


Fig. 6. Linear interpolation waveform.

4.2. Circular Interpolation

Simulation waveform for circular interpolation module is shown in Fig. 7. The starting point is point (0, 5) and the end coordinates is point (5, 0).

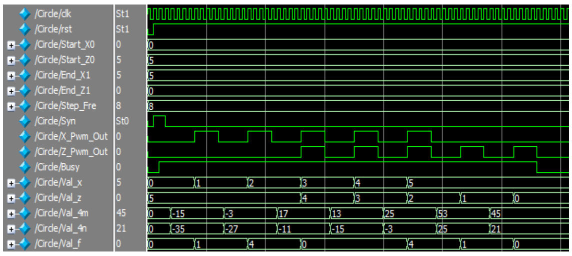


Fig. 7. Circular interpolation waveform.

Intermediate signal: *Val_4m* and *Val_4n* represent the deviation judgment formula $4F(M)$ and $4F(N)$.

4.3. Circular Interpolation

Simulation waveform for linear interpolation modules is shown in Fig. 8. The starting point is point (0, 4) and the end coordinates is point (6, 0).

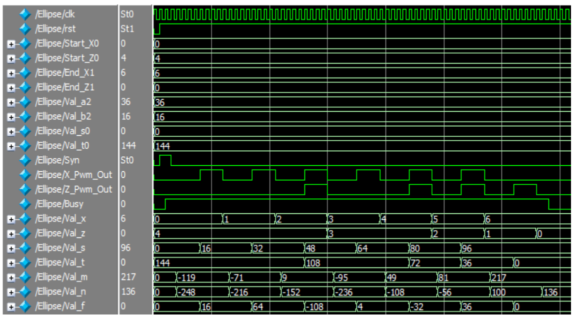


Fig. 8. Elliptical interpolation waveform.

Here are some intermediate signals which are not mentioned in the Fig. 8.

Input signal: *Val_a2* and *Val_b2* represent a^2 and b^2; *Val_s*0 and *Val_t*0 represent the initial value of s_0 and t_0;

Intermediate signal: *Val_s* and *Val_t* represent s_i and t_i.

4.4. Parabolic Interpolation

Simulation waveform for linear interpolation modules is shown in Fig. 10. The starting point is the origin (0, 0) and the end coordinates is point (4, 4).

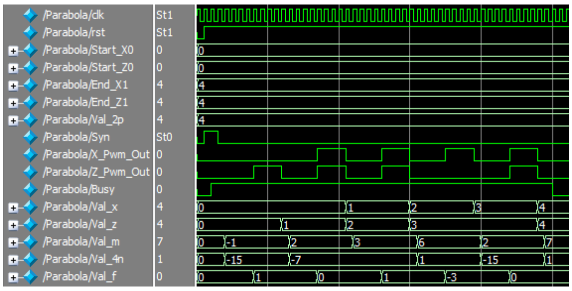


Fig. 10. Parabola interpolation waveform.

Here are some intermediate signals which are not mentioned in the Fig. 10.

Input signal: *Val_a2* and *Val_b2* represent a^2 and b^2; *Val_s*0 and *Val_t*0 represent the initial value of s_0 and t_0;

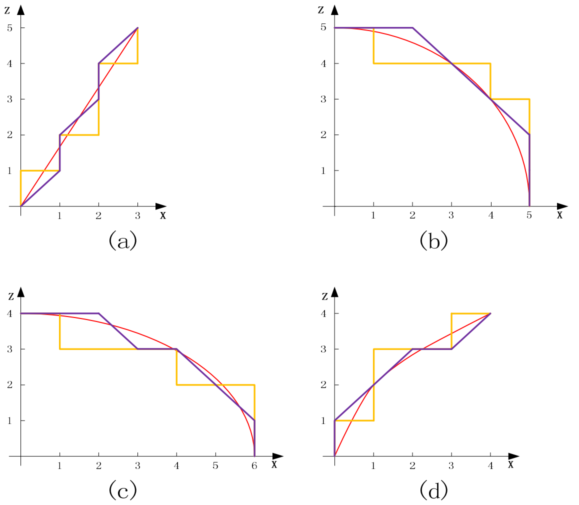
Intermediate signal: *Val_4n* represent the deviation judgment formula $4F(N)$.

5. The Analysis of Speed and Accuracy

In theory, the maximum error is only the half of the min-step size. In order to compare the effect of interpolation, we compared it with the result of By-Point Comparison method.

In the Fig. 11, (a), (b), (c) and (d) represent the interpolation trajectories of line, circle, ellipse and parabola; the red one represents ideal curve; the yellow one represents trajectories of By-Point

Comparison; the purple one represents trajectories of Slightest Min-Error. Using the min-step size as the reference standard, speed can be represented by total number of steps and accuracy can be represented by normalized value. The interpolation parameters of the four curves were shown in Table 1. Slightest Min-Error has higher speed and accuracy.



(a) (b)

(c) (d)

Fig. 11. General curve interpolation trajectory.

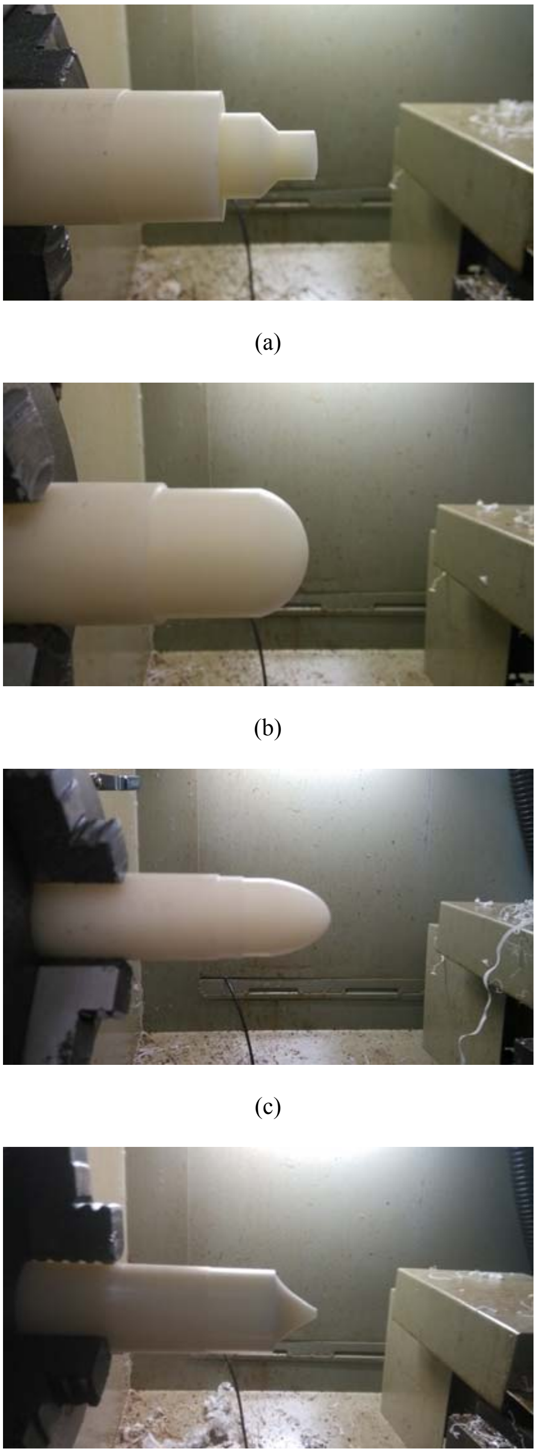
Table 1. Speed and accuracy of the two interpolation algorithms.

General Curves	Line	Circle	Ellipse	Para bola
Start point	(0,0)	(0,5)	(0,4)	(0,0)
End point	(3,5)	(5,0)	(6,0)	(4,4)
Speed of By-Point Comparison	8	10	10	8
Speed of Slightest Min-Error	5	7	7	5
maximum error of By-Point Comparison	0.686	0.877	0.937	0.743
maximum error of Slightest Min-Error	0.343	0.385	0.431	0.400

6. Verification

The actual two-axis CNC system controller is composed of MCU STM32F407 and FPGA EP2C8Q208C8. MCU Controller is responsible for human-computer interaction, G-code analysis and communication with FPGA. EP2C8Q208C8 is responsible for hardware interpolation, IO output and signal detection. The kernel software of the FPGA is developed in Quartus II. Four kinds of general curves interpolation picture is shown below.

In the Fig. 12, (a), (b), (c) and (d) represent the interpolation trajectories of line, circle, ellipse and parabola. The min-step size of the servo controller is at the magnitude of 1 um.



(a)

(b)

(c)

(d)

Fig. 12. General curve interpolation picture.

7. Conclusion

A general curve interpolation algorithm based on FPGA is presented, which is realized based on FPGA hardware platform, programmed by Verilog HDL, simulated by software ModelSim and finally verified on a two-axis CNC lathe. Because the proposed Slightest Min-Error interpolation algorithm has high mobile speed and machining accuracy, whose maximum error is less than half of the min-step size, it can be widely used in precision CNC systems and industrial robots. Because the FPGA has the characteristics of simple program, highly integrated, real-time, parallel and low prices, it has a high application value and economic benefits.

References

[1]. K. Erwinski, M. Paprocki, L. Grzesiak, et al., Application of Ethernet Powerlink for communication in a Linux RTAI open CNC control system, *IEEE Transactions on Industrial Electronics,* Vol. 60, Issue 2, 2013, pp. 628 - 636.

[2]. R. E. Breaz, G. S. Racz, O. C. Bologa, et al., Motion Control of medium size CNC Machine-Tools–A Hands-on Approach, in *Proceedings of the Conference on Industrial Electronics and Applications (ICIEA),* 2012, pp. 2212-2217.

[3]. P. Sofer, J. Skuta, J. Gebauer, Drilling of circuit boards using CNC controlled by programmable logic controller, in *Proceedings of the 13th International Carpathian Control Conference (ICCC),* 2012, pp. 711 - 714.

[4]. S. Tarnovean, R. E. Breaz, C. Biris, et al., Simulation tools for studying the behavior of CNC laser cutting machines in different working regimes, in *Proceedings of the 16th International Conference on System Theory, Control and Computing (ICSTCC)*, 2012, pp. 1-6.

[5]. P. Kaewsaiha, N. Dejdumrong, A Combined Linear and Circular-Arc Approximation of Curves for Feedrate Smoothing of CNC Machining, in *Proceedings of the 10th International Conference Computer Graphics, Imaging and Visualization (CGIV),* 2013, pp. 23-26.

[6]. Suk-Hwan Suh, Seong-Kyoon Kang, Dae-Hyuk Chung, Ian Stroud, Theory and Design of CNC Systems, *Springer-Verlag London Limited*, 2008.

[7]. J. Balic, Model of automated computer aided NC machine tools programming, *Journal of Achievements in Materials and Manufacturing Engineering*, Vol. 17, Issue 1-2, 2006, pp.309-312.

[8]. M. Tikhon, NURBS interpolator for constant material removal rate in open NC machine tools, *International Journal of Machine Tools & Manufacture*, Vol. 44, 2004, pp. 27-36.

[9]. D. Ungureanu-Anghel, D.-I. Belci, E. Raduca, M. Raduca, Minimum vicinity interpolation algorithms for calculation of trajectory processing for a model of CNC machine tool, in *Proceedings of the International Joint Conference on Computational Cybernetics and Technical Informatics (ICCC-CONTI),* 2010. pp. 367-372.

[10]. T. Masood, A. Mubashar, M. A. Jawad Khan, Techniques used in the simulation and control of PROTEC computer numerically controlled (CNC) machine software Industrial Technology, in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT)*, 2002, pp. 504-509.