Sensors & Transducers, Vol. 176, Issue 8, August 2014, pp. 296-305

Sensors & Transducers

An Improved Rotary Interpolation Based on FPGA

*** Mingyu Gao, Jiaxiang Lou, Zhanxiong Wu, Jilong Ye**
[1] Hangzhou Dianzi University, Hangzhou, 310018, China
[2] E-mail: mackgao@hdu.edu.cn

Received: 21 May 2014 /Accepted: 31 July 2014 /Published: 31 August 2014

Abstract: This paper presents an improved rotary interpolation algorithm, which consists of a standard curve interpolation module and a rotary process module. Compared to the conventional rotary interpolation algorithms, the proposed rotary interpolation algorithm is simpler and more efficient. The proposed algorithm was realized on a FPGA with Verilog HDL language, and simulated by the ModelSim software, and finally verified on a two-axis CNC lathe, which uses rotary ellipse and rotary parabolic as an example. According to the theoretical analysis and practical process validation, the algorithm has the following advantages: firstly, less arithmetic items is conducive for interpolation operation; 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.

Keywords: Rotation, 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.

The conventional rotary interpolation algorithms [6-7] directly substitute the rotated coordinate conversion formula into a standard interpolation curve, get s rotary equation in the original coordinates, and use the By-Point Comparison algorithm [8], the Digital Differential Analyzer (DDA) based algorithm [9] or the Minimum-Error algorithm [10] to complete interpolation calculation. Because the rotated coordinate conversion formula use value and to substitute (or) and most general rotation curves have squared terms, there will be more other items in the expanded equations.

Thus, this expanded items lead to a more complex computing problem. And there is a proportional relationship between accuracy and calculation complexity, which is showed as the higher accuracy, the more complex calculation.

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. Improved Rotary Interpolation Theory

The rotated coordinate conversion formula of mathematical sense is shown in Fig. 1. Point P rotates a counterclockwise angle of θ around the origin, φ is the angle between line OP and X-axis and r is the distance from the origin to the point P.

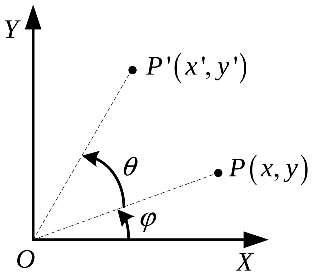

Fig. 1. Rotation conversion.

The polar form of the point P is:

$$\begin{cases} x = r\cos\theta \\ y = r\sin\theta \end{cases}, \tag{1}$$

So P' derive:

$$\begin{cases} x' = r\cos(\theta+\varphi) \\ = r\cos\varphi\cos\theta - r\sin\varphi\sin\theta \\ = x\cos\theta - y\sin\theta \\ y' = r\sin(\theta+\varphi) \\ = r\cos\varphi\sin\theta + r\sin\varphi\cos\theta \\ = x\sin\theta + y\cos\theta \end{cases}, \tag{2}$$

Unlike conventional one-step rotary interpolation algorithm, this improved rotary interpolation algorithm is divided into two parts: 1) Standard curve interpolation output, using the algorithm such as By-Point Comparison, DDA or Minimum-Error; 2) Standard curve interpolation output rotary process. Because standard curve interpolation is not the focus of this paper, just use the conclusion. The improved rotary process is similar to DDA method. When the standard curve interpolation output, accumulate the value and overflow output respectively. There are two kinds of direction case in the standard curve interpolation: same and opposite. And the relationship between improved rotary interpolation and the direction of standard curve interpolation is shown below.

2.1. In the Same Direction

The same direction means that both X-axis and Y-axis are positive or negative, which lead to the same rotary result. Just using positive X-axis and Y-axis as an example in Fig. 2, there are three possible output cases in the standard curve interpolation: (a) represent only the X-axis output two min-step size; (b) represent only the Y-axis output two min-step size; (c) represent both the X-axis and the Y-axis output two min-step size. Line Color Description: the black one and the coordinate scale represent the interpolation direction and the min-step size; the purple one (OP_1 and OP_2) represents the standard curve interpolation output; the dark blue one (OQ_1 and OQ_2) represents the purple one rotating an angle of θ; the light blue one (OM_1, OM_2, ON_1 and ON_2) represents the projection value of the dark blue one on the direction axis; the red one (OX_1, OX_2, OY_1 and OY_2) represents the output of rotation result.

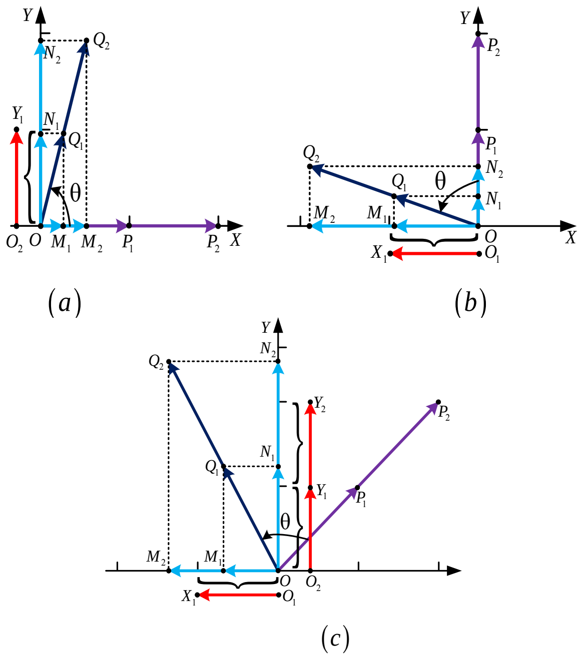

Fig. 2. Rotary schematic of same direction.

According to Fig. 2, derive the cumulative formula of X and Y axes:

(a) Only the X-axis output one min-step size, derive the cumulative formula:

$$\begin{cases} Fx_{i+1} = Fx_i + \cos\theta \\ Fy_{i+1} = Fy_i + \sin\theta \end{cases}, \tag{3}$$

(b) Only the Y-axis output one min-step size, derive the cumulative formula:

$$\begin{cases} Fx_{i+1} = Fx_i - \sin\theta \\ Fy_{i+1} = Fy_i + \cos\theta \end{cases}, \tag{4}$$

(c) Both X and Y axes output one min-step size, derive the cumulative formula:

$$\begin{cases} Fx_{i+1} = Fx_i - sin\theta + \cos\theta \\ Fy_{i+1} = Fy_i + \sin\theta + \cos\theta \end{cases}, \tag{5}$$

Judge the cumulative value, after the end of accumulation:

1) When $Fx_{i+1} \geq 1$ (or $Fy_{i+1} \geq 1$), the cumulative value of X (or Y) axis is overflow, output one min-step size and recalculate the cumulative value:

$$Fx_{i+1} = Fx_{i+1} - 1 \text{ (or } Fy_{i+1} = Fy_{i+1} - 1\text{)}, \tag{6}$$

2) When $Fx_{i+1} \leq -1$ (or $Fy_{i+1} \leq -1$), the cumulative value of X (or Y) axis is overflow, output one min-step size and recalculate the cumulative value:

$$Fx_{i+1} = Fx_{i+1} + 1 \text{ (or } Fy_{i+1} = Fy_{i+1} + 1\text{)}, \tag{7}$$

In order to make a more complete rotation curve, when the entire standard curve interpolation is complete and $0.5 \leq |Fx_{i+1}| < 1$ (or $0.5 \leq |Fy_{i+1}| < 1$), Still output one min-step size, before ending rotary interpolation.

Finally, give the initial conditions of cumulative value:

$$\begin{cases} Fx_0 = 0 \\ Fy_0 = 0 \end{cases}, \tag{8}$$

2.2. Opposite Direction

Opposite direction means that positive X-axis, negative Y-axis or negative X-axis, positive Y-axis, which lead to the same rotary result. Just using positive X-axis and negative Y-axis as an example in Fig. 3, there are three possible output cases in the standard curve interpolation: (a) represent only the X-axis output 2 min-step size; (b) represent only the Y-axis output 2 min-step size; (c) represent both the X-axis and the Y-axis output 2 min-step size. Definitions of line color can refer to Fig. 2.

According to Fig. 3, derive the cumulative formula of X and Y axes:

a) Only the X-axis output one min-step size, derive the cumulative formula:

$$\begin{cases} Fx_{i+1} = Fx_i + \cos\theta \\ Fy_{i+1} = Fy_i + \sin\theta \end{cases}, \tag{9}$$

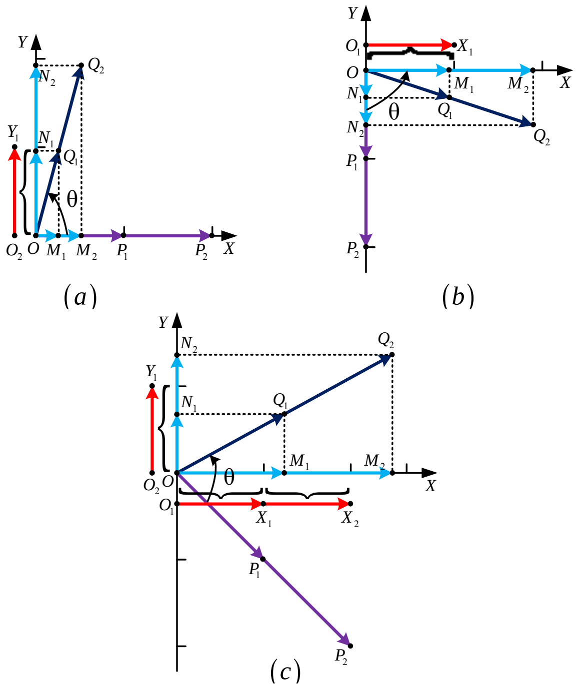

Fig. 3. Rotary schematic of opposite direction.

b) Only the Y-axis output one min-step size, derive the cumulative formula:

$$\begin{cases} Fx_{i+1} = Fx_i + \sin\theta \\ Fy_{i+1} = Fy_i - \cos\theta \end{cases}, \tag{10}$$

c) Both X and Y axes output one min-step size, derive the cumulative formula:

$$\begin{cases} Fx_{i+1} = Fx_i + \sin\theta + \cos\theta \\ Fy_{i+1} = Fy_i + \sin\theta - \cos\theta \end{cases}, \tag{11}$$

Judge the cumulative value, after the end of accumulation:

When $Fx_{i+1} \geq 1$ (or $Fy_{i+1} \geq 1$), the cumulative value of X (or Y) axis is overflow, output one min-step size and recalculates the cumulative value:

$$Fx_{i+1} = Fx_{i+1} - 1 \text{ (or } Fy_{i+1} = Fy_{i+1} - 1\text{)}, \tag{12}$$

2) When $Fx_{i+1} \leq -1$ (or $Fy_{i+1} \leq -1$), the cumulative value of X (or Y) axis is overflow, output one min-step size and recalculates the cumulative value:

$$Fx_{i+1} = Fx_{i+1} + 1 \text{ (or } Fy_{i+1} = Fy_{i+1} + 1\text{)}, \tag{13}$$

In order to make a more complete rotation curve, when the entire standard curve interpolation is complete and $0.5 \leq |Fx_{i+1}| < 1$ (or $0.5 \leq |Fy_{i+1}| < 1$), Still output one min-step size, before ending rotary interpolation.

Finally, give the initial conditions of cumulative value:

$$\begin{cases} Fx_0 = 0 \\ Fy_0 = 0 \end{cases}, \qquad (14)$$

3. Monotonic Interval Division

Because the track of rotated regular curves, like ellipse and parabola, are not monotonic anymore in every quadrant, this brings great difficulties to program. Therefore, redefining every monotonous interval is necessary for interpolation.

3.1. Interval of Ellipse

Shown in Fig. 4, there are two coordinate systems: the standard coordinate system XOY and the rotary coordinate system $X'OY'$. Line Color Description: the black one represents two coordinate systems and the angle between them is θ; the light blue one (denoted as I) represents an ellipse in the standard coordinate system XOY; the dark blue (denoted as II) one represents an ellipse in the rotary coordinate system $X'OY'$ and it can be obtained from the light blue ellipse rotating an counterclockwise angle of θ; Points $H'(x_1', y_1')$, $I'(x_2', y_2')$, $J'(x_3', y_3')$, $K'(x_4', y_4')$ represent the extreme points of the dark blue ellipse and the dark red line (denoted as III) represents the tangent of this extreme points; the light red one (denoted as IV) is obtained from the dark red line rotating an counterclockwise angle of θ and intersects the standard ellipse at points $H(x_1, y_1)$, $I(x_2, y_2)$, $J(x_3, y_3)$, $K(x_4, y_4)$, which is just the extreme points coordinate of the unrotated ellipse.

In Fig. 4, the rotary ellipse is divided into four monotonic intervals by the tangent points H', I', J' and K'. Because the improved rotary interpolation use the standard ellipse interpolation as a rotary object, just need to calculate the coordinate of points H, I, J, K in the standard ellipse.

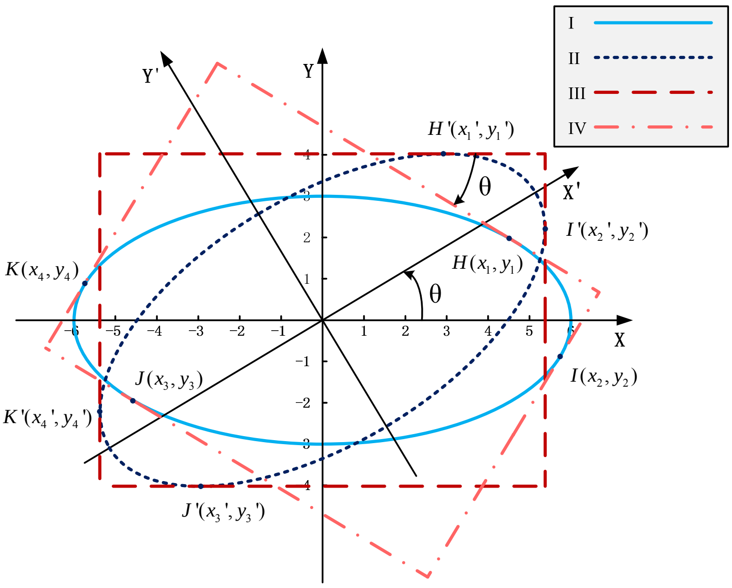

Fig. 4. Schematic interval of rotary ellipse.

The ellipse equations in the standard coordinate system XOY is

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \qquad (15)$$

Demand the x derivative:

$$\frac{dy}{dx} = \frac{-b^2 x}{a^2 y}, \qquad (16)$$

Because the angle between two tangent lines and the X axis is $(\pi - \theta)$, obtain:

$$\frac{dy}{dx} = \frac{-b^2 x}{a^2 y} = -\tan\theta, \qquad (17)$$

So:

$$H(x_1, y_1) = \left(\frac{a^2 \tan\theta}{\sqrt{a^2 \tan^2\theta + b^2}}, \frac{b^2}{\sqrt{a^2 \tan^2\theta + b^2}} \right), \qquad (18)$$

$$J(x_3, y_3) = \left(\frac{-a^2 \tan\theta}{\sqrt{a^2 \tan^2\theta + b^2}}, \frac{-b^2}{\sqrt{a^2 \tan^2\theta + b^2}} \right), \qquad (19)$$

The angle between another two tangent lines and the X axis is $(\pi/2-\theta)$, obtain:

$$\frac{dy}{dx}=\frac{-b^2x}{a^2y}=\frac{1}{\tan\theta}, \tag{20}$$

So:

$$I(x_2,y_2)=\left(\frac{a^2}{\sqrt{a^2+b^2\tan^2\theta}},\frac{-b^2\tan\theta}{\sqrt{a^2+b^2\tan^2\theta}}\right), \tag{21}$$

$$K(x_4,y_4)=\left(\frac{-a^2}{\sqrt{a^2+b^2\tan^2\theta}},\frac{b^2\tan\theta}{\sqrt{a^2+b^2\tan^2\theta}}\right), \tag{22}$$

So the ellipse is divided into four monotonic intervals by the points H, I, J and K. In every monotonic interval, use cumulative rotation processing method mentioned in the previous chapter to complete the entire elliptical rotary interpolation.

3.2. Interval of Parabola

Similar to the ellipse, shown in Fig. 5, points $M'(x_1',y_1')$, $N'(x_2',y_2')$ represent the extreme points of the rotary parabola and points $M(x_1,y_1)$, $N(x_2,y_2)$ represent the extreme points coordinate of the unrotated parabola.

In Fig. 5, the rotary parabola is divided into three monotonic intervals by the tangent points M' and N'. Because the improved rotary interpolation uses the standard parabola interpolation as a rotary object, just need to calculate the coordinate of points M and N in the standard parabola.

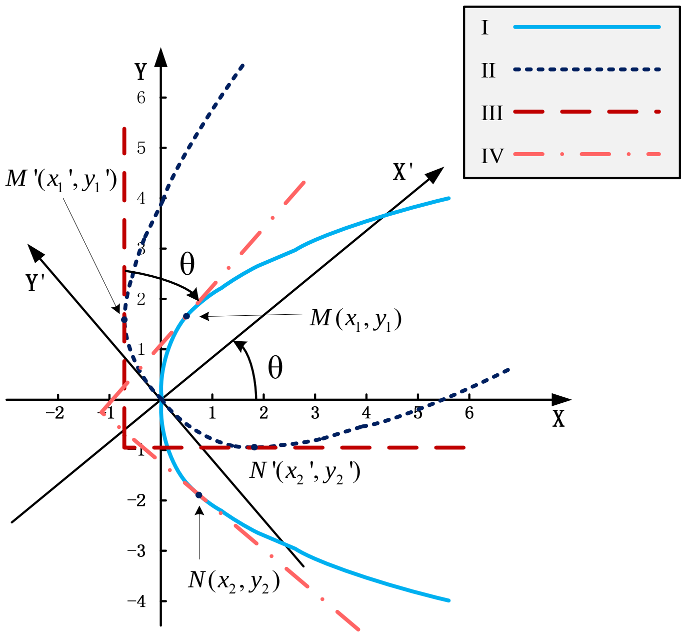

Fig. 5. Schematic interval of rotary parabola.

The parabola equations in the standard coordinate system XOY is

$$y^2=2px, \tag{23}$$

Demand the x derivative:

$$\frac{dy}{dx}=\frac{p}{y}, \tag{24}$$

Because the angle between one tangent line and the X axis is $(\pi/2-\theta)$, obtain:

$$\frac{dy}{dx}=\frac{p}{y}=\frac{1}{\tan\theta}, \tag{25}$$

So:

$$M(x_1,y_1)=\left(\frac{p\tan^2\theta}{2},p\tan\theta\right), \tag{26}$$

The angle between another two tangent lines and the X axis is $(\pi-\theta)$, obtain:

$$\frac{dy}{dx}=\frac{p}{y}=-\tan\theta, \tag{27}$$

So:

$$N(x_2, y_2) = \left(\frac{p}{2\tan^2\theta}, \frac{-p}{\tan\theta} \right), \tag{28}$$

So the parabola is divided into three monotonic intervals by the points M and N. In every monotonic interval, use cumulative rotation processing method mentioned in the previous chapter to complete the entire parabola rotary interpolation.

4. Software Simulation

The improved rotary interpolation algorithm is realized based on FPGA hardware platform by Verilog HDL. Here shows the simulation waveforms in ModelSim software, which displays rotary ellipse and rotary parabolic interpolation.

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

1) Input signal: clk is the clock signal; rst is the reset signal; $Start\_X0$, $Start\_Z0$, End_X1 and End_Z1 represent the starting point and end point coordinates of the standard interpolation curve; Val_a2 and Val_b2 are the Elliptic coefficients a^2 and b^2. Val_2p is the Parabola coefficient $2p$. Val_Sin and Val_Cos are the angle value $\sin\theta$ and $\cos\theta$ which are amplified 100000000 times; Syn is a synchronization start signal; X_Dir_In and Z_Dir_In are the direction output of rotary interpolation;

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

3) Intermediate signal: Val_m, Val_n, Val_4n and Val_f represent the deviation judgment formula which is used to calculate standard ellipse or parabola interpolation; $X\_Pwm\_Out\_Std$ and $Z\_Pwm\_Out\_Std$ are the output of standard ellipse or parabola interpolation; Val_XTemp and Val_ZTemp are the cumulative value that stand for Fx_i and Fy_i; Val_x and Val_z show the current coordinate values.

4.1. Waveform of Rotary Ellipse

Simulation waveform for rotary ellipse interpolation modules is shown in Fig. 6. The starting coordinate of the standard ellipse is point (0, 8); the end coordinate of the standard ellipse is point (16, 0) and the rotary angle is π/6.

4.2. Waveform of Rotary Parabola

Simulation waveform for rotary parabola interpolation modules is shown in Fig. 7. The starting point of the standard parabola is the origin (0, 0); the end coordinate of the standard parabola is point (8, 8) and the rotary angle is π/6.

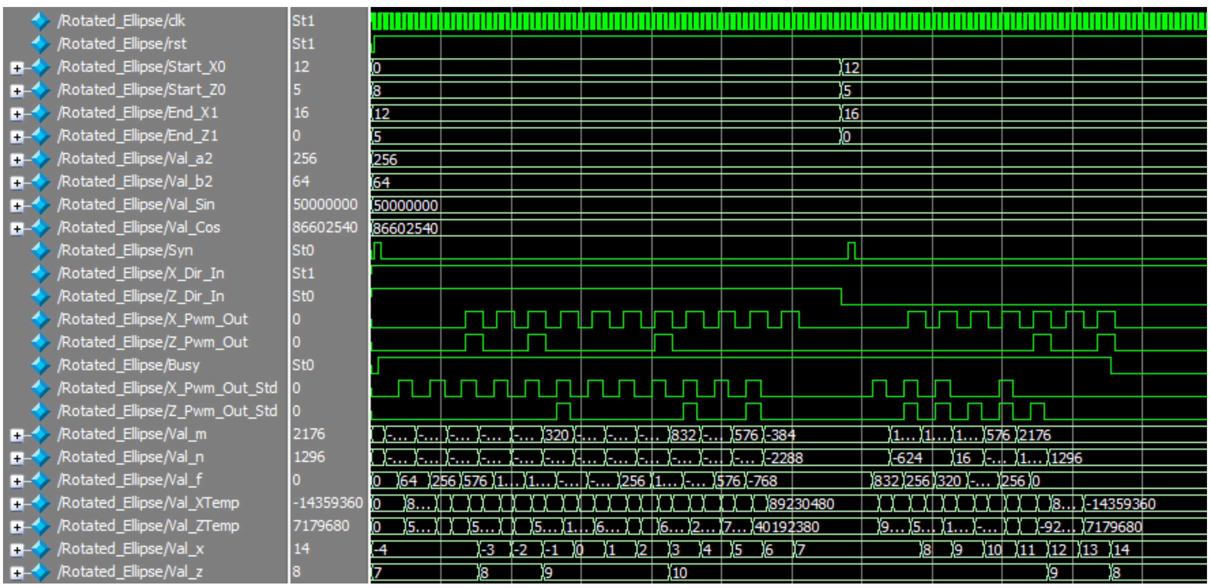

Fig. 6. Interpolation waveform of rotary ellipse.

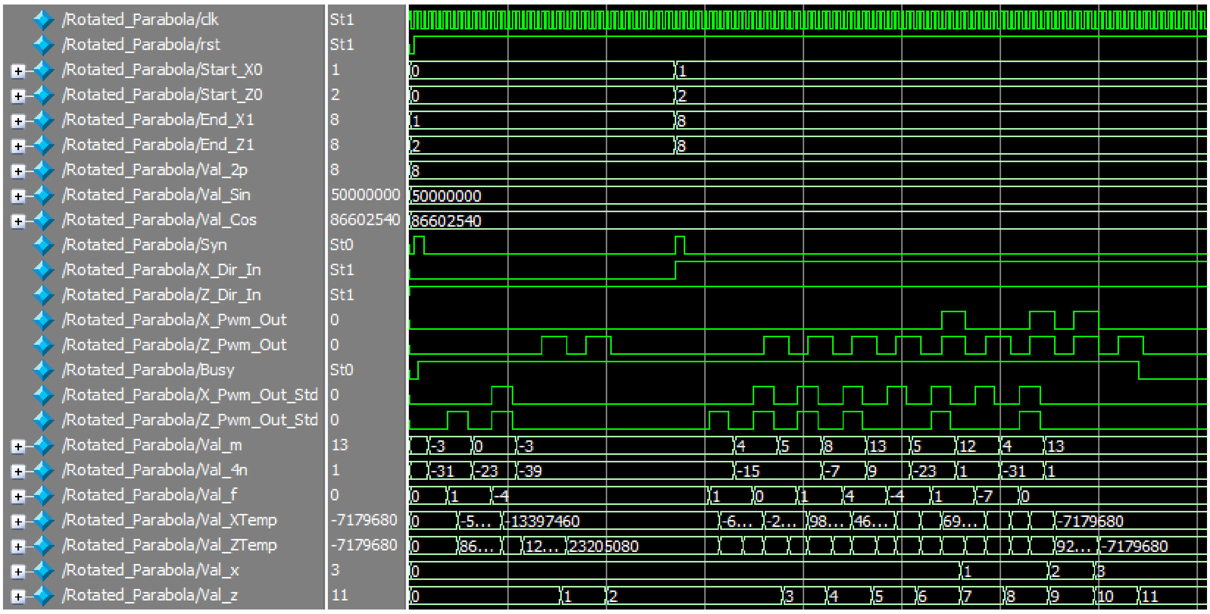

Fig. 7. Interpolation waveform of rotary parabola.

5. Analysis of Accuracy and Complexity

The improved rotary interpolation algorithm is divided into two parts: standard curve interpolation and rotary process. Using the FPGA's characteristics of parallel, this two parts work just like the assembly line and the computing time is only two clock cycles, which is only decided by standard curve interpolation. Different with the conventional rotary interpolation algorithm, which uses the rotary coordinate as input variables, the improved rotary interpolation algorithm use the standard curve coordinate as input variables. Because the input must be integer values, there are some rounding errors between the rotary coordinate and the standard curve coordinate. Here draw the interpolation track and list the rounding error and the maximum error. By default, all values in figure and table are normalized.

5.1. Accuracy and Complexity of Rotary Ellipse

Shown in Fig. 8, the light blue one(denoted as I) represents a standard ellipse; the dark blue one (denoted as II) is obtained from the light blue ellipse rotating an counterclockwise angle of $\pi/6$; the purple one (denoted as III) represents the standard ellipse interpolation output by Minimum-Error interpolation algorithm; the dark red one (denoted as IV) represents the rotary ellipse output by the improved rotary interpolation algorithm; the light red (denoted as V) one is obtained from the dark red one rotating an clockwise angle of $\pi/6$.

Because there are rounding errors between the rotary coordinate and the standard curve coordinate, list the starting point, highest point and end point to understand the errors better in Table 1. And there is a big rounding error at highest point about 0.441.

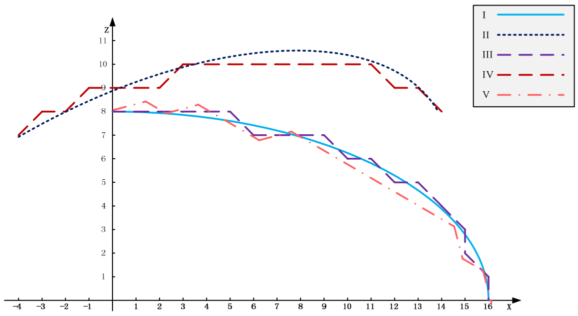

Fig. 8. Interpolation track of rotary ellipse.

Table 1. Rounding error of rotary ellipse.

Rounding Error	Starting point	Highest point	End point
Coordinates of standard ellipse	(0.000, 8.000)	(5.237, 12.095)	(16.000, 0.000)
Integer values and rounding error	(0,8) →0.000	(5,12) →0.255	(16,0) →0.000
Coordinates of rotary ellipse	(-4.000, 6.928)	(7.856, 10.583)	(13.856, 8.000)
Integer values and rounding error	(-4,7) →0.072	(8,11) →0.441	(14,8) →0.144

The maximum error of rotary ellipse track in Fig. 8 is shown in Table 2. The maximum error is 0.626, which is less than a minimum step size. Combined with Table 1, the accuracy of the improved rotary interpolation algorithm is similar to conventional rotary interpolation algorithm.

Table 2. Maximum error of rotary ellipse.

Maximum error	Interpolation coordinate	Error value
Maximum error of standard ellipse	(15,2)	0.432
Error of rotary elliptical starting point	(-4,7)	0.072
Error of rotary elliptical highest point	(8,10)	0.601
Error of rotary elliptical end point	(14,8)	0.144
Maximum error of rotary ellipse	(12,9)	0.626

Because the complexity is not like accuracy which can be measured in digit, here just list the rotary formula of conventional rotary interpolation algorithm, which is obtained from the standard elliptic equation by substituting into the coordinate conversion formula.

$$\left(\frac{x\cos\theta+y\sin\theta}{a}\right)^2+\left(\frac{y\cos\theta-x\sin\theta}{b}\right)^2=1, \quad (29)$$

Simplify,

$$Ax^2+By^2+Cxy+D=0, \quad (30)$$

where

$$\begin{cases} A=a^2\sin^2\theta+b^2\cos^2\theta \\ B=a^2\cos^2\theta+b^2\sin^2\theta \\ C=2\sin\theta\cos\theta(b^2-a^2) \\ D=-a^2b^2 \end{cases}$$

Compared with standard elliptic equation, the improved rotary interpolation algorithm don't have item xy.So the complexity is less than conventional rotary interpolation algorithm, but this is not obvious.

5.2. Accuracy and Complexity of Rotary Parabola

Shown in Fig. 9, the light blue one (denoted as I) represents a standard parabola; the dark blue one (denoted as II) is obtained from the light blue parabola rotating an counterclockwise angle of $\frac{\pi}{6}$; the purple one (denoted as III) represents the standard parabola interpolation output by Minimum-Error interpolation algorithm; the dark red one (denoted as IV) represents the rotary parabola output by the improved rotary interpolation algorithm; the light red one (denoted as V) is obtained from the dark red one rotating an clockwise angle of π/6.

Because there are rounding errors between the rotary coordinate and the standard curve coordinate, list the starting point, leftmost point and end point to understand the errors better in Table 3. And there is a big rounding error at leftmost point about 0.538.

The maximum error of rotary parabola track in Fig. 9 is shown in Table 4. The maximum error is 0.577, which is less than a minimum step size. Combined with Table 3, the accuracy of the improved rotary interpolation algorithm is similar to conventional rotary interpolation algorithm.

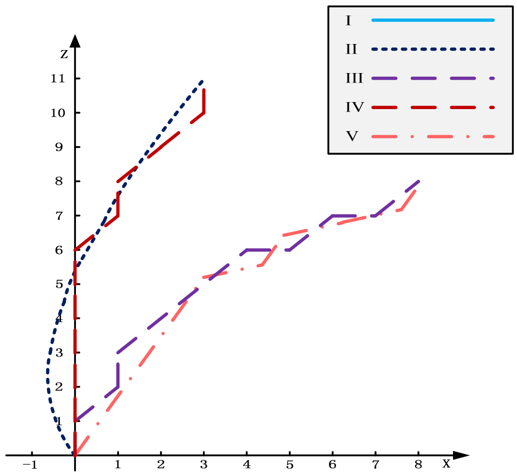

Fig. 9. Interpolation track of rotary parabola.

Table 3. Rounding error of rotary parabola.

Rounding Error	Starting point	Leftmost point	End point
Coordinates of standard parabola	(0.000, 0.000)	(0.667, 2.309)	(8.000, 8.000)
Integer values and rounding error	(0,0) →0.000	(1,2) →0.454	(8,8) →0.000
Coordinates of rotary parabola	(0.000, 0.000)	(-0.577, 2.333)	(2.938, 10.928)
Integer values and rounding error	(0,0) →0.000	(-1,2) →0.538	(3,11) →0.088

Table 4. Maximum error of rotary parabola.

Maximum error	Interpolation coordinate	Error value
Maximum error of standard parabola	(1,2)	0.443
Error of rotary parabola starting point	(0,0)	0.000
Error of rotary parabola highest point	(0,2)	0.577
Error of rotary parabola end point	(3,11)	0.144
Maximum error of rotary parabola	(0,2)	0.577

Because the complexity is not like accuracy which can be measured in digit, here just list the rotary formula of conventional rotary interpolation algorithm, which is obtained from the standard parabola equation by substituting into the coordinate conversion formula.

$$\left(y\cos\theta - x\sin\theta\right)^2 = 2p\left(x\cos\theta + y\sin\theta\right), \qquad (31)$$

Simplify,

$$Ax^2 + By^2 + Cxy + Dx + Ey = 0\,, \qquad (32)$$

where

$$\begin{cases} A = \sin^2\theta \\ B = \cos^2\theta \\ C = -2\sin\theta\cos\theta \\ D = -2p\cos\theta \\ E = 2p\sin\theta \end{cases}$$

Compared with standard parabola equation, the improved rotary interpolation algorithm don't have items x^2, xy and xy. So the complexity is much lower than conventional rotary interpolation algorithm and it at least reduces 30 %.

6. Picture of Rotary Products

The actual two-axis CNC system controller is composed of MCU STM32F407 [11] and FPGA EP2C8Q208C8 [12]. STM32F407 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. According to different rotary curve, here list the working drawings, tool trajectory and the picture of real products. The min-step size of the servo controller is at the magnitude of 1 um.

6.1. Process of Rotary Ellipse

Fig. 10 is the working drawing of rotary ellipse; Fig. 11 is the tool trajectory of rotary ellipse; Fig. 12 is the picture of real rotary products.

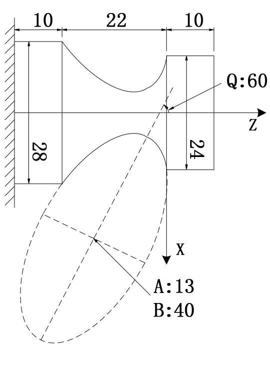

Fig. 10. Working drawing of rotary ellipse.

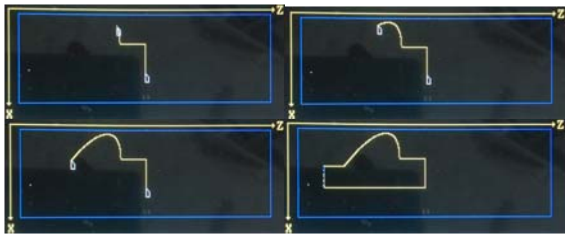

Fig. 11. Tool trajectory of rotary ellipse.

Fig. 12. Picture of real rotary products.

6.2. Process of Rotary Parabola

Fig. 13 is the working drawing of rotary parabola; Fig. 14 is the tool trajectory of rotary parabola; Fig. 15 is the picture of real rotary products.

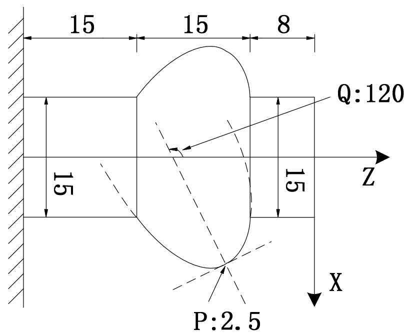

Fig. 13. Working drawing of rotary parabola.

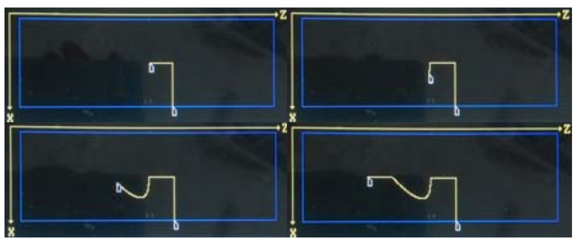

Fig. 14. Tool trajectory of rotary parabola.

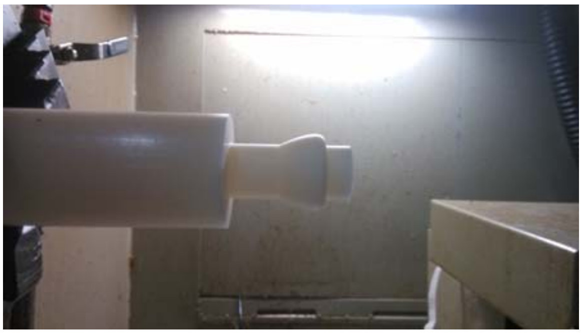

Fig. 15. Picture of real rotary products.

7. Conclusion

An improved rotary interpolation algorithm was presented, which was 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 improved rotary interpolation algorithm is a post-processing method, it can be used not only for ellipse and parabola interpolation, but also for other rotary curve like hyperbola, sinusoidal line and so on. And compared with the conventional rotary interpolation algorithms, it has the characteristics of less arithmetic items and simple programming, which means 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]. Chan-Hong Lee, Gyungho Khim, Reliability design of CNC system, in *Proceedings of the International Conference on Smart Manufacturing Application (ICSMA 2008)*, September 2011, pp. 1140-1142.

[2]. Yang Yong, Wen Shan, Distributed intelligent maintenance system for CNC machine tools based on Kansei engineering, in *Proceedings of the International Conference on Artificial Intelligence and Computational Intelligence (AICI'09)*, November 2009, pp. 467-471.

[3]. Ge-Yuan Ding, Zhen-Jiang Li, Wei-Tang Sun, Jian Xing, The design and implementation of CNC system based on the circular buffer, in *Proceedings of the International Conference on System Science and Engineering (ICSSE'2012)*, June 2012-July 2012, pp. 534-538.

[4]. Tao Yu, Junxi Bi, Lilan Liu, A study of intelligent integrated nano CNC system based on standard, , in *Proceedings of the 2nd IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, 13-16 August 2006, pp. 1-5.

[5]. F. Calabrese, G. Celentano, Design and realization of a STEP-NC compliant CNC embedded controller, in *Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2007)*, September 2007, pp. 1010- 1017.

[6]. Yong Zhang, Feihu Zhang, Guangyu Zhang, Dianrong Luan, Research on NC interpolation algorithm for rotary symmetric aspheric ultrasonic-magnetorheological compound finishing, *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures*, Vol. 27, Issue 3, May 2009, pp. 1509-1513.

[7]. Liang Quan, Wang Yongzhang, The studies of spline interpolation for five-axis machining, in *Proceedings of the 4th IEEE Conference on Industrial Electronics and Applications (ICIEA 2009)*, May 2009, pp. 2770-2777.

[8]. Huang Bin, Wu Chunxue, The research and realization of Circular interpolation track based on the Point-to-point comparison method of CNC in Scilab, in *Proceedings of the 2nd International Conference on Networking and Digital Society (ICNDS'2010)*, May 2010, pp. 267-269.

[9]. C.-Y. Chen, P.-S. Liao, C.-C. Cheng, G.-F. Jong, Design and implementation of real-time NURBS interpolator for motion control, in *Proceedings of the 2nd IEEE Conference on Industrial Electronics and Applications (ICIEA'2007)*, May 2007, pp. 426-431.

[10]. 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)*, May 2010, pp. 367-372.

[11]. STM32F407xx (http://www.st.com/web/en/resource/technical/document/datasheet/DM00037051.pdf).

[12]. Cyclone_II_Device_Handbook (http://www.st.com/web/en/resource/technical/document/datasheet/DM00037051.pdf).