

Elliptic Curved Component Macro-Programming and Its Application

Zhibo Yang, Junchen Hu, Kaiqiang Li, Shiyu Zhang and Aiju Liu

School of Mechanical and Power Engineering, Henan Polytechnic University, Jiaozuo 454000, China

E-mail: Yangzhibo2001@163.com

Abstract. Most conventional numerical control systems do not have the function of noncircular curve interpolation instruction. Manual programming is extremely challenging, automatic programming by computer-aided manufacturing software is highly sophisticated, and processing parameters cannot be easily modified. Therefore, macro-programs, which possess powerful parametric programming, are applied for the processing of noncircular curved components. The values of arguments were determined using transfer and loop statements (IF and WHILE), and elliptic curved macro-programs were achieved using normal and parameter equations in this study. The elliptic curved components were fitted using micro-sized segments or arcs. The numerical control machining tests verified the validity and viability of the macro-programs, and elliptic curved components were processed. The results indicated that the elliptic curved components processed using macro-programs met the design requirements.

1. Introduction

In FANUC numerical control systems, the assignment of macro-programs by arguments is achieved using arguments, and the versatility and applicability of the programs can be enhanced using macro programming. Currently, most FANUC numerical control systems employ Category B macro-programs for programming[1,2]. In conventional programming, designated constants are used, and the program is executed according to the designed sequence with no skip allowed. The macro-programs, however, employ arguments that can be assigned and operated, thus enabling skip. For the processing of series components (with size variation but no shape change), conventional programs require re-programming for each component, while macro-programs need changes in argument values only[3]. Using macro-programs, noncircular curved components, such as conic curved ones, can be fitted using micro-sized lines or arcs to meet the precision requirements[4].

2. Macro-programming for Conic Curved Components

2.1. Macro-programming Mechanism

In the numerical controlled processing of elliptic curved components, the elliptic curve was divided into several tiny segments, which were fitted using micro-sized line segments or arcs. In this case, the processing precision was directly related to the quantity of segments divided[5]. For components with sophisticated configurations, only mathematical equations and algorithms were keyed in manually, and node coordinates were obtained and processed by macro-programs, thus reducing efforts for programming.



2.2. Macro-programming Format

In the first format, conditional transfer instructions (IF statements) were used:

IF [conditional expression]
GOTO m;

In this case, execution came before judgment. In other words, the program went to the mth section if the condition was satisfied; otherwise, the program went to the next section.

In the second format, loop instructions (WHILE statements) were used:

WHILE [conditional expression]
DO m;
...
END m;

In this case, judgment came before execution. In other words, the section from DO m to END m was executed if the condition was satisfied; otherwise, the section after END m was executed[6,7]. Herein, m could only be 1, 2, or 3; otherwise, alert was triggered[8].

3. Mechanism of Macro-Programming for Elliptic Curved Components

In the lathe workpiece coordinate system xoz , the z - and the x -axes intercepts were assumed to be a and b , respectively, and coordinates of Point p on the ellipse as (z, x) , as shown in Figure 1:

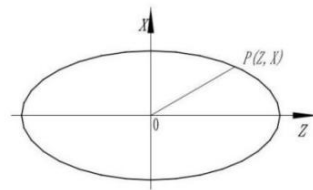


Figure 1. Elliptical mathematical model.

The standard equation of ellipse was as follows:

$$\frac{z^2}{a^2} + \frac{x^2}{b^2} = 1 \quad (1)$$

Where a and b were the semi-major axis and semi-minor axis, respectively, and $a > b > 0$.

The parameter equation of ellipse was as follows:

$$\begin{cases} z = a \cos \alpha \\ x = b \sin \alpha \end{cases} \quad (2)$$

Where α was the polar angle.

If the standard equation was used for the macro-programming of ellipse, coordinates of p_0, p_1, \dots, p_n with z or x as the argument were calculated, and the elliptic curve was processed by pointwise interpolation[9]. If the parameter equation was used for the macro-programming of ellipse, coordinates of p_0, p_1, \dots, p_n with angle α as the argument were calculated, as shown in Figure 2.

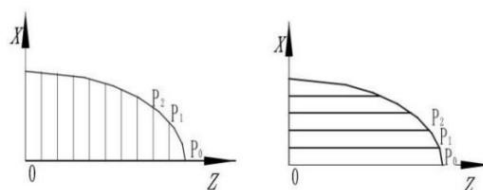


Figure 2. Elliptical Schematic of different processing method argument.

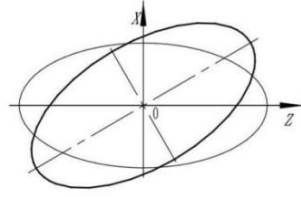


Figure 3. Mathematical model of inclined elliptic curve.

In the macro-program, arguments #24 and #26 were assumed to be x and z coordinates, respectively. The macro-program equation for the x coordinate in the standard equation was as follows:

$$\#24 = \frac{b}{a} * SQRT[a * a] - [\#26 * \#26] \quad (3)$$

Assuming argument #10 in the parameter equation as angle a , the macro-program equation was as follows:

$$\begin{cases} \#26 = a \cos[\#10] \\ \#24 = b \sin[\#10] \end{cases} \quad (4)$$

For the macro-programming of inclined elliptic components, coordinates were usually transformed[10], as shown in Figure 3.

As an inclined ellipse was obtained by the tilting of ellipse by θ , the ellipse equation could be transformed using the rotation transformation matrix:

$$\begin{cases} z' = z \cos \theta - x \sin \theta \\ x' = z \sin \theta + x \cos \theta \end{cases} \quad (5)$$

Where (x, z) and (x', z') were the coordinates before and after rotation, and θ was the rotation angle.

The simultaneous equations (5) and (1), the equation of inclined ellipse was as follows:

$$\begin{cases} z' = z \cos \theta - \frac{b}{a} \sqrt{a^2 - z^2} \sin \theta \\ x' = z \sin \theta + \frac{b}{a} \sqrt{a^2 - z^2} \cos \theta \end{cases} \quad (6)$$

With θ as an argument, the equation of inclined ellipse could be obtained from equation (2):

$$\begin{cases} z' = a \cos \alpha \cos \theta - b \sin \alpha \sin \theta \\ x' = a \cos \alpha \sin \theta + b \sin \alpha \cos \theta \end{cases} \quad (7)$$

In the macro-program, arguments #1 and #2 were set as x' and z' coordinates of Point P. The parameter equation of the macro-program was as follows:

$$\begin{cases} \#2 = \#26 * \cos[\theta] - \#24 * \sin[\theta] \\ \#1 = \#26 * \sin[\theta] + \#24 * \cos[\theta] \end{cases} \quad (8)$$

Where $\#24 = \frac{b}{a} * SQRT[a * a] - [\#26 * \#26]$.

4. Application of Macro-Programming for Elliptic Curved Component

The component shown in Figure 4 was processed by employing macro-programming. The blank is a round bar ($\Phi = 50 \times 100 \text{ mm}^2$), and the intersection point of its central axis with the right end face of

the elliptic curved component is defined as the origin for programming. The programs for rough and finish machining of elliptic curved components were developed based on the G73 fixed shape cutting cycle. Herein, the overall withdrawal amounts along x - and z -axes were both 2 mm (radius) and the cycle index was 14. T0101 was a rough turning tool with the main shaft speed = 500 r/min and feeding rate = 0.3 mm/r in the case of rough machining. During finish machining, T0202 was used and the allowances for finish were 0.5 mm and 0.1 mm along x -axis and z -axis, respectively. The feeding rate was 0.1 mm/r.

Two elliptic curves were observed on the component, and the origins from right to left were $\frac{z^2}{20^2} + \frac{x^2}{15^2} = 1$, $\frac{z^2}{20^2} + \frac{x^2}{14^2} = 1$. In the ellipse equation $\frac{z^2}{20^2} + \frac{x^2}{15^2} = 1$, the argument #26 was z in the ellipse equation, the initial value was #4 = 20, the final value was #5 = 0; the argument #24 was x in the ellipse equation, the initial value was 0, and the increment was #6 = 0.2. The processing lists for elliptic curve macro-arguments based on standard and parameter equations were shown in Table 1 and Table 2 respectively.

According to the coordinate system in the scheme and the coordinate system for programming, the central line of the ellipse and the central line of the component were located on 20 mm away from the program origin in the negative z direction. Hence, the coordinates of any point in the new system were x (2*#24), z (#26-20).

In the ellipse equation $\frac{z^2}{20^2} + \frac{x^2}{14^2} = 1$, the argument #26 was z in the ellipse equation, the initial value was #4 = 14, and the final value was #6 = -14; the argument #24 was x in the ellipse equation, and the increment was #6 = 0.2. The processing lists of macro-arguments in elliptic curves developed using standard and parameter equations was Table 3 and Table 4, respectively.

According to the coordinate system in the scheme and the coordinate system for programming, the central lines of the ellipse and component were parallel to each other. The origin in the ellipse coordinate system was 25 mm away from the program origin in the x direction and 34 mm away from the program origin in the negative z direction. Hence, the coordinates of any point in the new system were x (50-2*#24), z (#26-34).

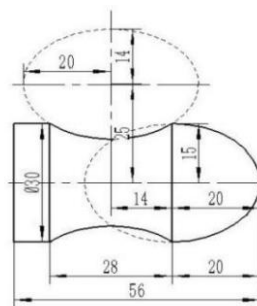


Figure 4. Elliptic Curved Component.

Table 1. Standard equation elliptic curve processing program variable table.

Variable selection	Variable representation	Macro variable	Macro variable
Independent variable	z	#26	#26
Domain	[20,0]	[#4, #5]	[#4, #5]
Dependent variable	$x = \frac{15}{20} \sqrt{20^2 - z^2}$	#24 = $\frac{15}{20} * SQRT[(20 * 20) - (\#26 * \#26)]$	

Table 2. Parameter equation elliptic curve processing program variable table.

Variable selection	Variable representation	Macro variable
Independent variable	α	#10
Domain	[0°, 90°]	[#4, #5]
Dependent variable	$\begin{cases} z = 20 \cos \alpha \\ x = 15 \sin \alpha \end{cases}$	$\begin{cases} \#26 = 20 \cos[\#10] \\ \#24 = 15 \sin[\#10] \end{cases}$

Table 3. Standard equation elliptic curve processing program variable table.

Variable selection	Variable representation	Macro variable
Independent variable	z	#26
Domain	[14, -14]	[#4, #5]
Dependent variable	$x = \frac{14}{20} \sqrt{20^2 - z^2}$	#24 = $\frac{14}{20} * SQRT[(20 * 20) - (\#26 * \#26)]$

Table 4. Parameter equation elliptic curve processing program variable table.

Variable selection	Variable representation	Macro variable
Independent variable	α	#10
Domain	[54.3°, -54.3°]	[#4, #5]
Dependent variable	$\begin{cases} z = 20 \cos \alpha \\ x = 14 \sin \alpha \end{cases}$	$\begin{cases} \#26 = 20 \cos[\#10] \\ \#24 = 14 \sin[\#10] \end{cases}$

Figure 5 shows the flow chart of macro-program for an elliptic curve. The program flow chart was developed according to the logical flow, followed by the development of processing procedures according to the program flow chart. The developed program was imported to the FANUC numerical control lathe. First, the developed program was simulated to avoid damages to lathe or cutting tools caused by program errors. Figure 6 shows the cutting tool route on the numerical control panel; the

validity of the developed program could be verified according to the cutting tool route shown. Figure 7 shows the component processed by the proposed approach. The specifications of this component were demonstrated to satisfy design requirements. In the processing of series components (with size variation but no shape change) by macro-programs, only changes in argument values were needed. In this way, standardization, modularization, and versatility of numerical control processing were realized. Using macro-programs, noncircular curved components, such as conic curved ones, could be fitted using micro-sized lines or arcs to meet the precision requirements. In the processing of elliptic curved components with varying parameters (e.g., initial and final points, angle, semi-major axis, and semi-minor axis), one general macro-program with these parameters as arguments allowed processing of various components without re-programming. Additionally, this study provided references for the process of other nonlinear curved components such as hyperbolic and parabolic curves.

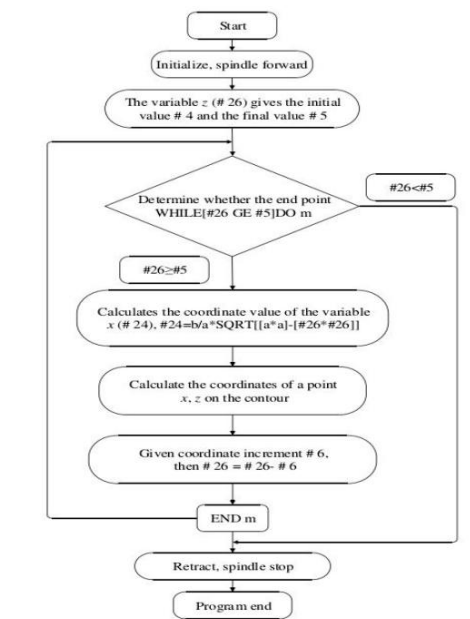


Figure 5. The flow chart of macro-program for an elliptic curve.



Figure 6. The cutting tool route on the numerical control panel.



Figure 7. The component processed by the proposed approach.

5. Conclusions

(1) Owing to its mathematical calculation function, loop statements, and conditional transfer instructions, macro-programming was employed for numerical control processing of elliptic curved components. The programming procedures were reduced, while the versatility and applicability of the programs were enhanced.

(2) The elliptic curves were fitted using micro-sized segments or arcs. Rough and finish machining of elliptic curved components were achieved using the G73 fixed shape cutting cycle instructions. The

argument was defined using IF and WHILE statements. Elliptic curved components were processed using standard or equation-based macro-programs to eliminate issues of ellipse configuration in processing by numerical control lathe. In this way, tedious node calculations were avoided, and manual programming work required was significantly reduced.

(3) Additionally, this approach provided references for macro-programming for other noncircular curves. Parameter By increasing the quantity of difference points, the elliptic curve was fitted by micro-sized segments, and the processing precision was significantly enhanced.

6.Acknowledgments

This research was supported by the financial sup-ports from the National Natural Science Funds Fund (U1204517), Natural Science Fund by the Education Department of Henan Province P.R. China (12B460011), and the Opening Project of Key Laboratory of Precision Manufacturing Technology and Engineering, Henan Polytechnic University.

7.References

- [1] Li P, Miao FX and Zhang M 2014 Application of fanuc cnc system macro-program in elliptic parts turning *Mech. Eng. Automat.* (3) 59-60
- [2] Xu H 2013 *Fanuc cnc system with my manual programming* (Beijing:Chemical Industry Press) pp 62-63
- [3] Wang MY 2015 Exploration into the macro programming of cnc lathes ellipses *Appl. Mech. Mater.* 727-728 823-826
- [4] Zhang M, Li P and Miao FX 2015 Applying macro programming to turn non elliptic curve parts by fanuc system *Mech. Eng. Automat.* (2) 196-197
- [5] Liu XC 2016 Study on the macro programming of nc machining convex ellipse and concave ellipse *Mech. Elecl. Eng. Technol.* 45 (3) 60-62
- [6] Huang QG 2011 Application of user macro-program processing non-circular curve parts in cnc turning process *Manuf. Automat.* 33 (9) 49-51
- [7] Liu YL, Jia T 2009 Research on how to write the macro program of ellipse parts and its application *Manuf. Automat.* 31 (7) 92-94
- [8] Ling WC 2012 CNC programing of the quadratic curve *Mech. Elecl. Eng. Technol.* 41 (7) 76-78
- [9] Li GY, Feng YH and Wang LL 2012 Macro programming skills and practice for ellipse part *Coal Mine Mach.* 33 (8) 137-138
- [10] Li YX, Li XS and Li XL 2015 Application of macro program processing non-circular curve profile using NC lathe *Henan Sci.* 33(2) 204-208