The possibility of cutting shapes on CNC machine tools

Piotr Paszta^{1,*}

¹Czestochowa University of Technology, Institute of Mechanical Technologies,
Al. Armii Krajowej 21, 42-201 Czestochowa, Poland

Abstract. The development of computer technology has enabled it to be introduced to machine tools. The widespread use of CNC machine tools has resulted in a rapid development of the processing industry. So far, obtaining the profile of the curve has required the use of a complicated machine tool kinematic design, or it has been impossible to implement it otherwise than by copying. The numerical control of machine tools is based on the mathematical description of motion. This provides the capability to form curves of the same shape but with different dimensions using parametric programs. The article describes the practical use of parameters in machining on an FYS milling machine with Mitsubishi control.

1 Introduction

Programming numerically controlled machine tools involves the notation of all motion trajectories and technological activities in a programming language understandable for the machine tool in a manner that allows the execution of an object of the desired shape, dimensions and surface quality [1- 2].

The job of numerical control is to convert numerical data carrying the information of the work program into quantities controlling machine tool operation. Numerical data describing the geometrical shape of an object, which can be obtained from the constructional drawing, and technological numerical data can be used directly for controlling the machine tool. In shape control, only some profile points are preset, while the controlling quantities, as the exact relationships of the path versus time, are determined by interpolation in the processor and the controls of motions in servo-drive axes. The generation of controlling displacement quantities takes place based on numerical computations.

The tool should move along a trajectory that can be resolved into straight and curvilinear segments.

An arbitrary contour is obtained through the cooperation of two or more advance motion drive motors. The majority of shape controls have the capability of combined linear and circular interpolation. Modern CNC machine tools used in the aircraft industry and the

* Corresponding author: paszta@itm.pcz.pl

manufacture of moulds and dies have the capability to use helical, parabolic, cubic and NURBS interpolations. In those milling machines, in addition of the displacement of the tool, also its axis direction in space changes [3].

2 The possibility of parametric programming

2.1 Milling a cam of a logarithmic spiral profile

The assumption in the development of a parametric program for a CNC machine tool is to demonstrate the machine tool's capability and operation principle and the possibility of creating parametric programs. As a profile to be machined, a cam has been assumed, which is composed of a semicircle and two curves of a logarithmic spiral profile. The program is created in two steps. The first step includes the development of the program and the verification of the correctness of its operation using the program CIMCO Edit. This simulator allows for checking for any possible errors occurred during programming and for any undesirable tool motions, which would prevent the operator from any collisions during machining on the machine. The next step is to make the part using the FYS numerically controlled milling machine. Figure 1 shows the profile of the cam.

Fig. 1. Theoretical cam profile.

The lower part of the cam is made by circular interpolation G03, while the logarithmic curves are made by linear interpolation G01. The variable curve radius r should be calculated from the formula describing the logarithmic spiral in polar coordinates, when the spiral pole coincides with the pole (origin) of the coordinate system.

Fig. 2. The logarithmic spiral created in the Wolfram CDF Player program.

The formula describing the logarithmic spiral in polar coordinates, when the spiral pole coincides with the pole (origin) of the coordinate system:

$$r = ae^{b\varphi} \quad (1)$$

where:

e - base of the natural logarithm, $a, b \in R$.

One „end” of the spiral winds onto the pole (for $\varphi \rightarrow -\infty$, the spiral approaches asymptotically the pole $r \rightarrow 0$), while the other „escapes” to infinity (for $\varphi \rightarrow \infty$ spiral turns grow unlimitedly: $r \rightarrow \infty$).

The ratio of the derivative of the value of radius r with respect to angle φ to the radius is constant and equal to the coefficient in the exponent:

$$r'(\varphi)/r(\varphi) = b \quad (2)$$

therefore, the angle α between the spiral and the radii (half lines) originating from the spiral pole satisfies the equality: $b = \operatorname{ctg} \alpha$

So, this angle is constant: $\alpha = \operatorname{arctg} 1/b$, while coefficient b determines how "fast" and to which direction the spiral is turning.

For $b=0$ angle $\alpha=\pi/2$, the curve intersects the radii perpendicularly, therefore the spiral degenerates into a circle (with equation $r=a$). At b approach infinity, the spiral „straightens up”, in the limit approaching the half line $\varphi=0$.

The change in the sign of coefficient b is equivalent to the change in the sign of variable φ , so it corresponds to the axial symmetry relative to straight line $\varphi=0$, thus mapping the right-handed spiral into the left-handed one, and vice versa.

Coefficient a is the scale of the spiral (in function $\varphi \rightarrow r$ it occurs as a multiplier); so, it is responsible for the size of the curve. Change in its magnitude corresponds to the rotation of the spiral around the pole. Therefore, the parametric equations of the logarithmic spiral are as follows:

$$x(\varphi) = r \cos(\varphi) = ae^{b\varphi} \cos(\varphi) \quad (3)$$

$$y(\varphi) = r \sin(\varphi) = ae^{b\varphi} \sin(\varphi) \quad (4)$$

where $a, b \in R$.

The distance from the centre of successive spiral loops grows following a geometric progression. Starting from an arbitrary point on the spiral and continuing along it, one can go around the pole an arbitrary number of times, while not getting to it. However, the path that needs to be covered from that point to the pole is finite.

Figure 2 shows a logarithmic spiral in polar coordinates, which has been created in the Wolfram CDF Player program [4]. Each small black dot represents a spiral point $(x(\theta), y(\theta))$ for a different angle θ . In particular, 80 black dots are plotted for each completed round ($\Delta \theta = 2\pi$). The large black dot represents the "starting point" corresponding to $\theta=0$. The large blue dot highlights the point for a specific angle selected using a control.

Assuming that the radius R is equal to 35 mm, the formula for the curve will have the following form:

$$r = 35 + ae^{b\varphi} \quad (5)$$

Having already all the required data, one can proceed with assigning values to the variables needed for making the cam.

#101=35 (CAM DIAMETER),
#102=1.5 (AP),
#103=0 (START Z AXIS),

```

#104=50    (MILLING CUTTER DIAMETER),
#105=50    (SHAFT DIAMETER),
#120=0     (FI),
#107=1     (CONST "a"),
#108=1.4   (CONST "b"),
#109=0     (CAM INITIAL RADIUS),
#110=2     (DELTA FI),
#112=30    (DEPTH OF CUT),

```

This variable #110 (DELTA FI) is the angle value, upon which the computation of the polar coordinates follows. The first part of the curve lies in the angle range of $0^\circ - 90^\circ$, while the second part, in the range of $90^\circ - 180^\circ$.

Decreasing the value of variable causes an enhancement of the curve mapping. The angle 2° is sufficient for the mapping of the curve.

Having so defined variables, we move the tool to a safe distance by the roll.

```
G00 X-[#105/2+0.6*#104] Y#103 Z#103
```

The next step is to substitute the value of variable #102 for parameter #115 and the value of variable #120 for parameter #106. The successive commands of the program execute the lower cam profile.

```

#115=#102
N100
#106=#120
G01 X-[#105/2+0.6*#104]
G01 Z-#115
G01 X-[#101/2+#104/2]
G3 X#101/2+#104/2 Y0 I#101/2+#104/2 J0

```

After so made profile, the computation of the curve of the logarithmic spiral section follows. The first loop, in which the curve section will be executed depends chiefly on the parameter #106. Each loop passage increases it respectively by the value of #110. The loop continues operation until the value of variable #106 attains the angle of 90° . In the next step, variable #11 converts the value of parameter #106 from degrees to radians, and then the exponential function is computed in variable #114 and multiplied by the spiral scale. Finally, the computation of the radius length is made in variable #109. The tool shift needed for the determination of the curve in the X axis is computed by variable #130, while in the Y axis, by variable #131. The notation of the first loop is shown as follows:

```

N200
#106=#106+#110
#111=[#106*3.14159]/180
#113=#107*EXP[#108*#111]
#114=#101/2+#104/2.085
#109=#113+#114
#130=SIN[#106]*#109
#131=COS[#106]*#109 G01 X#131 Y#130
IF[#106 LT 90] GOTO 200

```

The second loop is very much similar. Only the method of computing the new angle and its conversion to radians change. Prior to performing computations, the value 90° is assigned to parameter #106. As the program continues, this parameter will be increased as before, that is by the value of variable #110.

```

#106=90
N300
#106=#106+#110
#111=[(180-#106)*3.1415]/180

```

```

#114=#101/2+#104/2.085
#113=#107*EXP[#108*#111]
#109=#113+#114
#132=SIN[#106]*#109
#133=COS[#106]*#109 G01 X#133 Y#132
IF[#106 LT 180] GOTO 300

```

The completion of the second loop means the execution of the full cam profile at the preliminary depth as defined by variable #102. To make the cam at the full depth, as defined by variable #110, another loop has been added, which encompasses the first and second loops and is execution as long, as the full depth value meets the condition.

```

#115=#115+#102
IF[-#115 GT -#112] GOTO 100

```

In the case when the remainder of the division of the total milling depth and the machining depth is not equal to zero, the value of variable #112 will be assigned to variable #115. This means that the mill has descended to the full depth. This descent will not damage the material during machining, because the layer being cut will not go beyond the layer written under parameter #102.

The whole of the programs looks as follows:

```

G90 G54 S450 M3 F100
#101=35
#102=1.5
#103=0
#104=50
#105=50
#120=0
#107=1
#108=1.4
#109=0
#110=2
#112=10
G00 X-[#105/2+0.6*#104] Y#103 Z#103
IF[#102 GE #112] GOTO 350
#115=#102
N100
#106=#120
G01 X-[#105/2+0.6*#104] G01 Z-#115
G01 X-[#101/2+#1A4/2]
G3 X#101/2+#104/2 Y0 I#101/2+#104/2 J0 N200
#106=#106+#110
#111=[#106*3.14159]/180
#113=#107*EXP[#108*#111]
#114=#101/2+#104/2.085
#109=#113+#114
#130=SIN[#106]*#109
#131=COS[#106]*#109 G01 X#131 Y#130 IF[#106 LT 90] GOTO 200
#106=90
N300
#106=#106+#110
#111=[[(180-#106)*3.1415]/180
#114=#101/2+#104/2.085
#113=#107*EXP[#108*#111]

```

```
#109=#113+#114
#132=SIN[#106]*#109
#133=COS[#106]*#109 G01 X#133 Y#132
IF[#106 LT 180] GOTO 300
#115=#115+#102
IF[-#115 GT -#112] GOTO 100 N350
#115=#112
N400
#106=#120
G01 X-[#105/2+0.6*#104] G01 Z-#115
G01 X-[#101/2+#104/2]
G03 X#101/2+#104/2 Y0 I#101/2+#104/2 J0 N500
#106=#106+#110
#111=[#106*3.1415]/180
#113=#107*EXP[#108*#111]
#114=#101/2+#104/2.08
#109=#113+#114
#130=SIN[#106]*#109
#131=COS[#106]*#109 G01 X#131 Y#130 IF[#106 LT 90] GOTO 500
#106=90
N600
#106=#106+#110
#111=[[180-#106]*3.1415]/180
#114=#101/2+#104/2.08
#113=#107*EXP[#108*#111]
#109=#113+#114
#132=SIN[#106]*#109
#133=COS[#106]*#109 G01 X#133 Y#132
IF[#106 LT 180] GOTO 600
G01 X-[#105/2+0.6*#104] Y#103 Z-#112 G0 Z#103
M30
```

To prevent the case in which the value of the total milling depth is smaller than the cutting depth, a loop has been used at the beginning of the program, which compares those values and in the case where the cutting depth is greater, a jump to line N350 is made. From this line, only one mill pass along the cam profile will occur at the full milling depth [5-9].

2.2 Simulation of the tool trajectory

The program and simulation are executed using CIMCO Edit. From the toolbar in the File Type tab, select the ISO Milling option and then click on the Nowy (New) icon.

After completing these activities, a new window will show up with a white field for entering the program code.

Passing successively to the Symulacja (Simulation) (1) toolbar tab and then the Inne (Others) item (2), select the relevant machine tool control system, under which the program is to be run.

Next, clicking on the simulation Window (3) will allow the preview of the previously written program code and the generation of the paths to be covered by the tool (Fig. 3).

To preview the executed part, the starting material needs to be generated by clicking on the solid Model, and then, depending on the machining to be performed, the dimensions needed during machining should be established. For the purposes of simulation, a roll of a

diameter of 50 mm and a height of 50 mm was used. The centre in the X and Y axes is equal to 0 mm, while in the Z axis, 25 mm.

Fig. 3. The generated tool paths.

By clicking on the OK button, a window will open, which will provide the capability to select the tool with which machining will be performed. Next, select the tool from the list and then enter its diameter in the respective field. This is a cylindrical mill of a diameter of 50 mm. Then, select the tool – mill.

Pressing the OK button will open a window with the already machined part. The program provided the capability to zoom in, rotate and move the part with the mouse. All of the basic views of the part can be set by selecting respective icons from the toolbar.

To run the program simulation, press the Start button (1). The speed of presentation is set by adjusting the slide bar (2). Button (3) allows simulation of the tool motion to be run block by block. Pressing button (4) causes the program to stop (Fig. 4). If the program and simulation run correctly, record the program.

Fig. 4. Program simulation.

2.3 Making a cam on the CBKO FYS 16NM milling machine

Before proceeding with the machining of the cam, the program should be entered to the machine tool. This can be done in two ways. The first method is to rewrite the program manually into the machine tool. This applies to short programs. The second method is to download the program via the RS-232C interface. Using a computer with the Hyperterminal software installed, send the cam program's code to the machine tool's memory. After sending, program simulation directly on the machine tool is required. Figure 5 shows a graphical animation of machining a part on the FYS milling machine.

Fig. 5. A view of the milling machine's monitor with a simulation of the mill trajectory.

In the case where the simulation has gone successfully, no errors are displayed, proceed with fixing the part and the tool in the machine tool's spindle.

The starting material is an aluminium roll with a diameter of 50 mm and a height of 35 mm. The roll was fixed in a machine vice equipped with prismatic jaws to enhance the rigidity of blank mounting. The tool for performing machining was a 50 mm-diameter cylindrical face cutter.

The next step involved facing of the roll face surface, descending of the tool along the Z axis to depth ap, sinking of the tool to the required diameter with linear interpolation G01 and making of the lower cam profile with circular interpolation G03.

The subsequent machining process was to generate the first curve, in which the condition was the increase of the angle up to a value of 90°. The interval, in which the angle increase occurred, spanned from 0° to 90°. The angle increase step was 2°; it could be changed in the program. With decreasing angle step, the machining accuracy increases. The tool is moved using linear interpolation G01.

The last part of the cam profile is done as before. However, the condition is to obtain the angle of 180°. The angle increment is contained between 90° and 180°. The tool is moved as before using linear interpolation G01.

After that, the tool retracts, goes down by depth ap, and then all the previous operations are repeated. The program continues operation until the programmed depth has been attained. The milling was done to a depth of 10 mm. The part made after the final pass is shown in Figure 6.

The following parameters were set during machining: feed rate, 100 mm/min; spindle rotational speed, 450 rpm, machining duration is 21 min and 23 sec. A mill with a diameter of 50 mm was used for the machining to eliminate vibration during milling.

The actual roughness value of the surface machined is however, dependent on many factors, chiefly on the plastic properties of the machined material, the vibrations of the machine tool-tool system, tool wear, etc. Their impact on the surface roughness is normally determined experimentally. The basic purpose of machining is to shape a new part surface so that it meets specified by the designer.

By selecting a high machining speed, a small feed and a small material layer, the additional finishing can be eliminated and a better surface quality can be achieved [10].

Fig. 6. The part made on the milling machine.

The following parameters were set during machining: feed rate, 100 mm/min; spindle rotational speed, 450 rpm; machining duration is 21 min and 23 sec. A mill with a diameter of 50 mm was used for the machining to eliminate vibration during milling.

3 Conclusions

Parametric programming has different notation capabilities, depending on the machine tool control system. This difference does not allow programming methods to be shown in a universal manner. For the creation of the machining code, the CIMCO Edit simulator was used. This program enables programs to be written for different machine tools with different control systems. To demonstrate the capability to cut complicated profiles during machining, a FYS milling machine was used, which had a Mitsubishi control system without in-built program cycles that would provide such capabilities. The program was created for a machine

tool with the Fanuc control system, which shows a broader possibility of using parametric programming. This difference is due to the lack of possibility of selecting the standard control system in the simulator. The Fanuc and Mitsubishi systems are very similar, which does not pose any problems during the creation of programs. The employed program enables graphical simulation and the detection of any errors contained in the machining program. The executed element is a cam, whose contour portion has the profile of a logarithmic curve. The program loaded using the RS-232C port was subjected to simulation directly on the machine tool to verify the consistence of tool motions. The object made represents only some capabilities of parametric programming.

When using the parametric programming of machine tools, the operator has the capability to machine many similar parts. Using variables, by changing the parameter value, an object of the same geometry but different dimensions can be obtained. In the case, where a control system does not have required machining cycles, the programmer has the possibility of creating a parametrized program and using it in the machining process. Parametric programming is complicated programming that requires the programmer to have a high level of expertise.

References

1. J. Honczarenko, *Obrabiarki sterowane numerycznie* (WNT Warszawa, 2008)
2. J. Kosmol, *Automatyzacja obrabiarek i obróbka skrawaniem* (WNT Warszawa, 2000)
3. W. Grzesik, P. Niesłony, M. Bartoszek, *Programowanie obrabiarek NC/CNC* (WNT Warszawa, 2010)
4. <http://demonstrations.wolfram.com>
5. Fanuc Series 16i-TB, 18i-TB, 160i-TB, 180i-TB, *Operation Manual*
6. Melder Series 60/60S, *Operation Manual*
7. Pronum, *Instrukcja programowania układu sterowania numerycznego CNC PRONUM 630 T*, (2011)
8. G. Nikiel, *Programming of CNC machine tools on the example of the Sinumerik 810D/840D control system* (ATH University of Bielska-Biala, 2004)
9. P. Rapp, *Engineers Black Book USA Edition* (Hi Impact Supplies, Los Angeles, 2005)
10. P. Paszta, MATEC Web of Conferences **94**, 02010 (2017)