

Advanced Programming Techniques for a CNC Milling Machine

Sotiris L. Omirou

Department of Mechanical Engineering, Frederick University,
Nicosia, Cyprus

Abstract:- This paper takes advantage of the powerful capabilities of modern CNC controllers such as sub-programming and parametrization to demonstrate the efficiency improvement of machining operations as a result of applying them in short, flexible and smart CNC programs. To show the benefits of using the above capabilities over conventional G/M programming language, three representative applications are presented through manufacturing drawings, flowcharts, program lists and simulation results. As it is deduced by all three cases, sub-programming and parametrization facilitates the full exploitation of the potential of a CNC machine and makes CNC machines easier to work with.

Key Words: CNC programming, Milling, Parameters, Canned cycles, Drilling, Cutter path generation.

1. INTRODUCTION

The operation of a CNC machine tool is controlled by a program written in the G/M-code programming language called part program. A part program contains an ordered sequence of statements specifying in detail the cutter motions to be executed and the auxiliary operations (e.g., spindle on/off, spindle speed and feedrate) to be realized by the CNC machine in order to machine a specified part. A statement is made up of words. Characters, and numerical digits are the elements that constitute a word. For example, the statement

```
N10 G01 X200 Y300 Z100 F200 M03 S800
```

consists of the words "N10", "G01", "X200", "Y300", "Z100", "F200", "M03", "S800", meaning that the statement with sequence number N10, requires movement of the tool from its current position to position (200, 300, 100) linearly (G01) with a feedrate of 200 mm/min and with the spindle rotating clockwise (M03) at 800 rpm.

Each of the codes other than G and M has a unique function. The G and M have many functions depending on the following two-digit number. These functions have been standardised and are commonly known as "G and M codes". Generally a G-code, named preparatory function code, defines one type of motion or one mode of operation while an M-code, named miscellaneous function code, turns various operations on/off (e.g., coolant flow, spindle, etc). It must be noted that not all the possible code numbers are allocated, which means that control systems manufacturers can use those remaining for whatever purpose they wish.

Parametric programming constitutes an innovative programming technique of G/M-code programming language. This innovation is not well known, which is unusual because of the fact that it simplifies the CNC procedures dramatically, lessens the user's dependence on CAD/CAM and contributes to bare minimum cycle time [8]. As production quantities grow, every second of program execution time becomes more critical. An important factor that minimizes the cycle time is how the program is formatted. While CAM systems have come a long way, it can be difficult, if not impossible, to cause some CAM systems to output CNC programs as efficiently as can be done with parametric programming technique.

Applying the technique of parametric programming, a part program can be written using variables, arithmetic and logic statements similar to usual programming languages such as Basic, Pascal or C. The specific is that in the case of CNC programming, all these parametric expressions are used to represent the machine axis position (x, y, z, a, etc.), feedrate, and spindle speed functions.

Like other computer programming languages, parametric programming comes in several versions. The most popular are those of Heidenhain and FANUC (Custom Macro B) controls [9]. Others include User Task (from Okuma), Q Routine (from Sodick), and Advanced Programming Language [APL] (from G& L). The present work uses the latest version of Heidenhain control [4], called iTNC 530.

The rest of the paper is organized as follows: Section 2 describes in detail the mechanism of defining and using parameters in the environment of Heidenhain control. Section 3 is an analytic presentation of three case studies. Each case represent an application category. Namely, case 1 illustrates the potential of sub-programming and parametrization to widen the motion generation capability of a CNC machine with interpolators other than the standard, case 2 points out the flexibility of a

parametric program to accommodate families of parts and case 3 shows how parametric programming can be used to invent new useful canned cycles. Finally, section 4 summarizes the contribution of the paper.

2. HEIDENHAIN PARAMETERS.

In the Heidenhain control [4], parameters are designated by the letter Q and a number between 0 and 99. Using Q parameters, an entire family of parts can be accommodated by a single part program. In such a case, Q parameters are entered in the part program instead of fixed numerical values. For example, several parts may require machining a pocket (figure 1) with different sizes. A single parametric program can be called up to machine the specific feature. Prior the execution of the parametric program, the numerical values for each pocket are transferred to the respective parameters. This approach eliminates the redundant codes in part program and substantially reduces the size of the program and programming time.

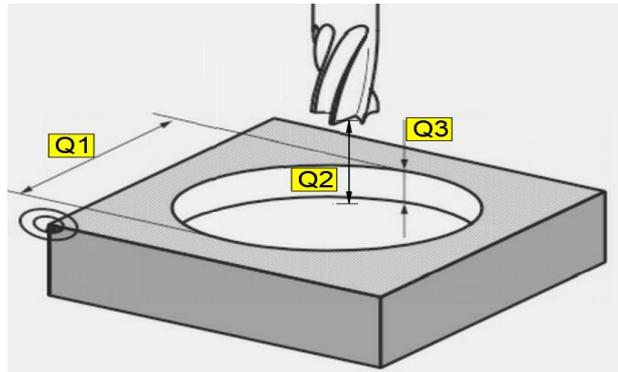


Fig. 1: A pocket dimensioned with parameters instead of fixed numerical values

Q parameters also enable the programming of contours that are defined with mathematical functions or can be used to make the execution of machining steps depend on logical conditions. According to the implementing function, Q parameters are classified in three groups (table 1.):

Table 1: Function Groups

Function group	Operation	Code
Basic arithmetic	Assign	D00
	Addition	D01
	Subtraction	D02
	Multiplication	D03
	Division	D04
	Square root	D05
Trigonometric functions	Sine	D06
	cosine	D07
	Root sum of squares	D08
	Angle	D13
If / then conditions, jumps	If equal jump	D09
	If not equal jump	D10
	If greater than jump	D11
	If less than jump	D12

Table 2: Description of the Operations

Operation	Example	Description of the example
D01: ADDITION	D01 Q1 P01 -Q2 P02 -5	Calculates and assigns the sum of two values to parameter Q1.
D02: SUBTRACTION	D02 Q4 P01 +10 P02 +5	Calculates and assigns the difference of two values to parameter Q4.
D03: MULTIPLICATION	D03 Q2 P01 +3 P02 +3	Calculates and assigns the product of two values to parameter Q2.
D04: DIVISION	D04 Q4 P01 +8 P02 +Q2	Calculates and assigns the quotient of two values to parameter Q4.
D05: SQUARE ROOT	D05 Q50 P01 4	Not permitted: division by 0
D06: SINE	D06 Q20 P01 -Q5	Calculates and assigns the square root of number 4 to parameter Q50.
D07: COSINE	D07 Q21 P01 -Q5	Calculate the sine of an angle in degrees (°) and assign it to parameter Q20
D08: ROOT SUM OF SQUARES	D08 Q10 P01 +5 P02 +4	Calculate the cosine of an angle in degrees (°) and assign it to parameter Q21.
D09: IF EQUAL, JUMP	D09 P01 +Q1 P02 +Q3 P03 5	Calculate and assign length from two values to parameter Q10.
D10: IF NOT EQUAL, JUMP	D10 P01 +10 P02 -Q5 P03 10	If the two values or parameters are equal, jump to the given label (5).
D11: IF GREATER THAN, JUMP	D11 P01 +Q1 P02 +10 P03 5	If the two values or parameters are not equal, jump to the given label (10).
D12: IF LESS THAN, JUMP	D12 P01 +Q5 P02 +0 P03 1	If the first parameter or value is greater than the second value or parameter, jump to the given label (5).
D13: ANGLE	D00 Q5 P01 +60	Assigns the numerical value +60 to parameter Q5.

3. CASE STUDIES

Case 1: Complex motions - Serpentine Curve

The case of a Serpentine curve (figure 2) was selected to illustrate the mechanism of generating complex motion using sub-programming and parametrization. Figure 3 shows the flow-chart at which the program of List 1 was based on. The same program can be executed to machine a serpentine of any size, at any starting angle, with any angle step and any milling depth. Simulation results are shown in figures 4 and 5.

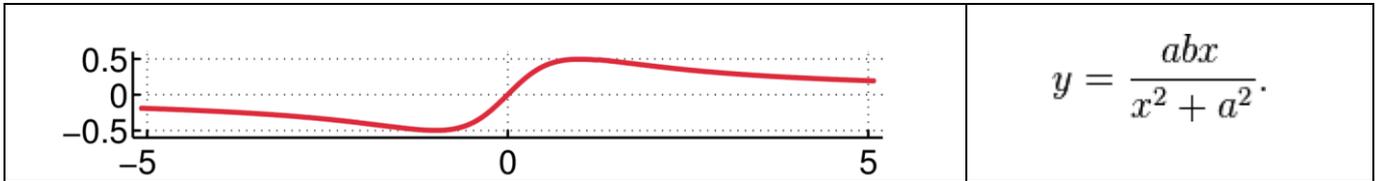


Fig.2: The serpentine curve for $a = b = 1$.

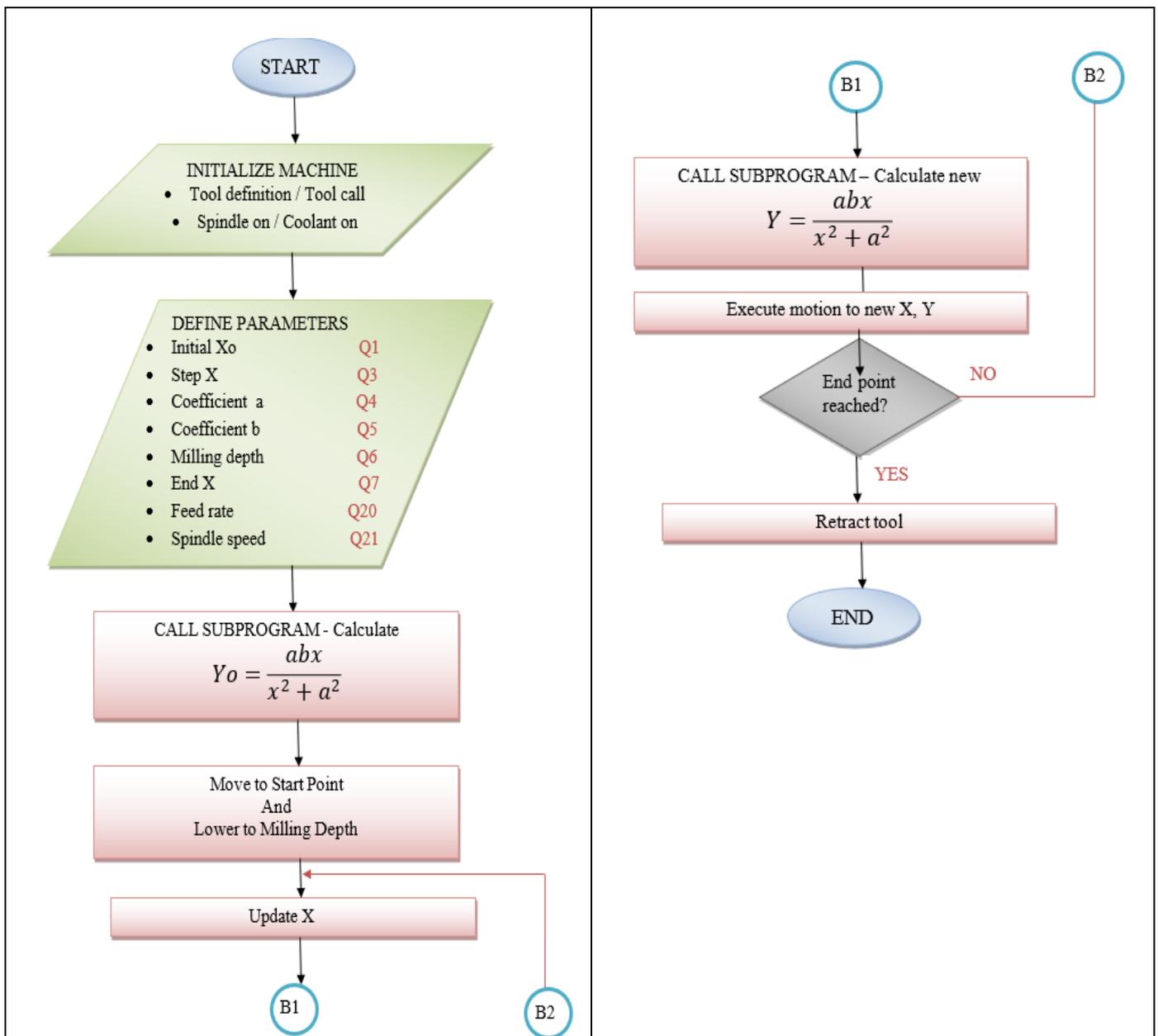


Fig.3: Flowchart for generating motion along a serpentine curve

```
%SERPENTINE G71 *
N10 G30 G17 X-60 Y-40 Z-20*
N20 G31 X+80 Y+40 Z+0*
N30 T1 G17*
N40 M3*
N50 D00 Q1 P01 +70*
N60 D00 Q3 P01 -1*
N70 D00 Q4 P01 +20*
N80 D00 Q5 P01 +15*
N90 D00 Q6 P01 -5*
N100 D00 Q7 P01 -40*
N110 D00 Q20 P01 +100*
N120 D00 Q21 P01 +1000*
N110 % CALCULATEY.I
N120 G90 G00 X+Q1 Y+Q2 Z+2*
N130 G01 Z+Q6 FQ20 SQ21*
N140 G98 L1*
N150 D01 Q1 P01 +Q1 P02 +Q3*
N160 % CALCULATEY.I
N170 G01 X+Q1 Y+Q2*
N180 D11 P01 +Q1 P02 +Q7 P03 1*
N190 G00 Z+100*
N200 M30*
N99999999 %SERPENTINE G71 *
```

-Tool definition/ Tool call
-Spindle on clock-wise,
coolant on

-Parameters definition
(Xo, Step X, a, b, milling
depth, end X, feedrate,
spindle speed)

-Call **Subprogram**
-Motion to starting point
-Lower to milling depth
-Subroutine1 - machining
operation
-Call **Subprogram**

-Motion to the next point
-Unfinished? Go to L1
-Retract in the tool axis
-End program

Subprogram

```
%CALCULATEY G71 *
N10 D03 Q10 P01 +Q4 P02 +Q5*
N20 D03 Q11 P01 +Q10 P02 +Q1*
N30 D03 Q12 P01 +Q1 P02 +Q1*
N40 D03 Q13 P01 +Q4 P02 +Q4*
N50 D01 Q14 P01 +Q12 P02 +Q13*
N60 D04 Q2 P01 +Q11 P02 +Q14*
N99999999 %CALCULATEY G71 *
```

-Calculation of Y

List 1: CNC program for motion generation along a serpentine curve

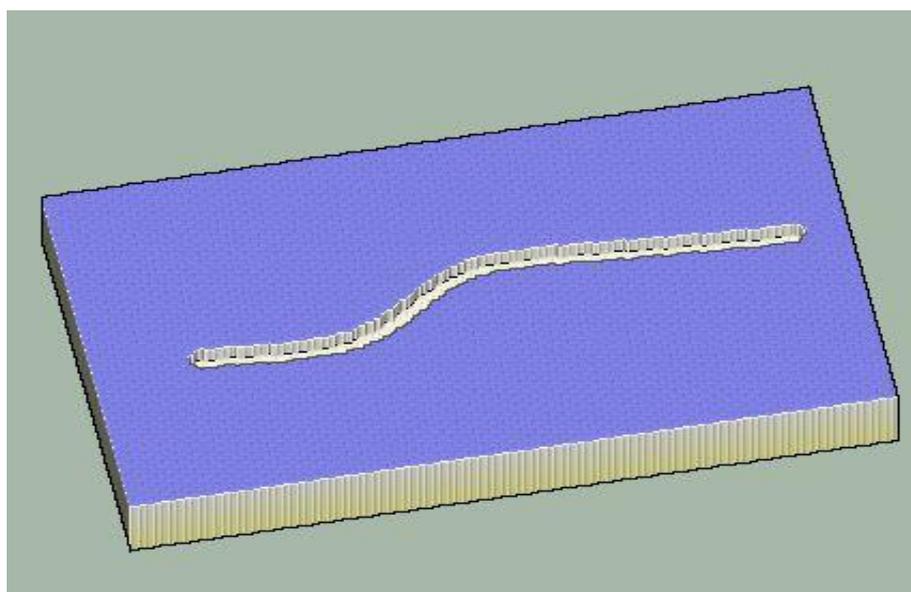


Fig. 4: Simulation test on HEIDENHAIN control

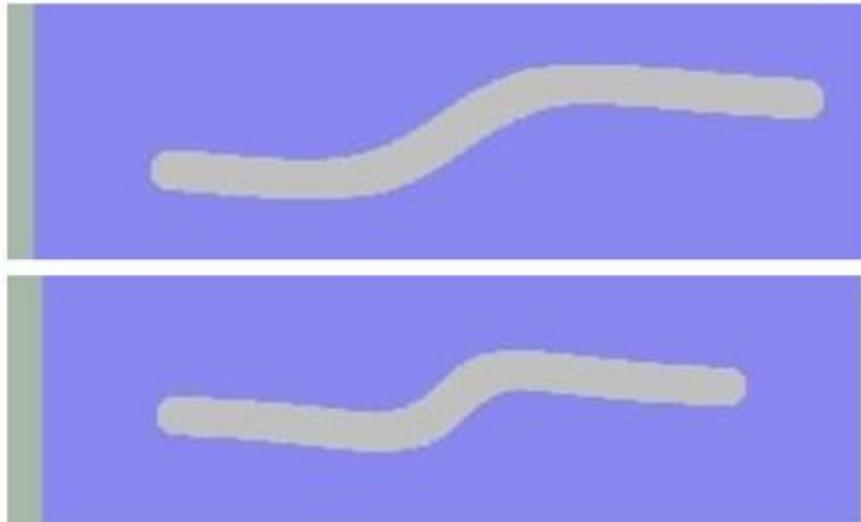


Fig. 5: Simulation tests for cutting a serpentine curve with different parameters.

Case 2 - Families of parts – A bolt circle

Almost all companies have at least some applications that fit into this category. Possibly they have prints dimensioned with variables right on the print. The programmer must reference a chart on the drawing to come up with values needed in the program. Or even if they don't have a perfect family of parts application, surely they have at least some workpieces that require similar machining operations. With parametrization, it is possible the development of general-purpose routines for similar machining operations.

The present application demonstrates the effectiveness of parametric programming for machining holes located on a circle at variable angle (Fig. 6). Figure 7 shows the flow-chart on which the program of List 2 was based on. Simulation result is shown in figure 8

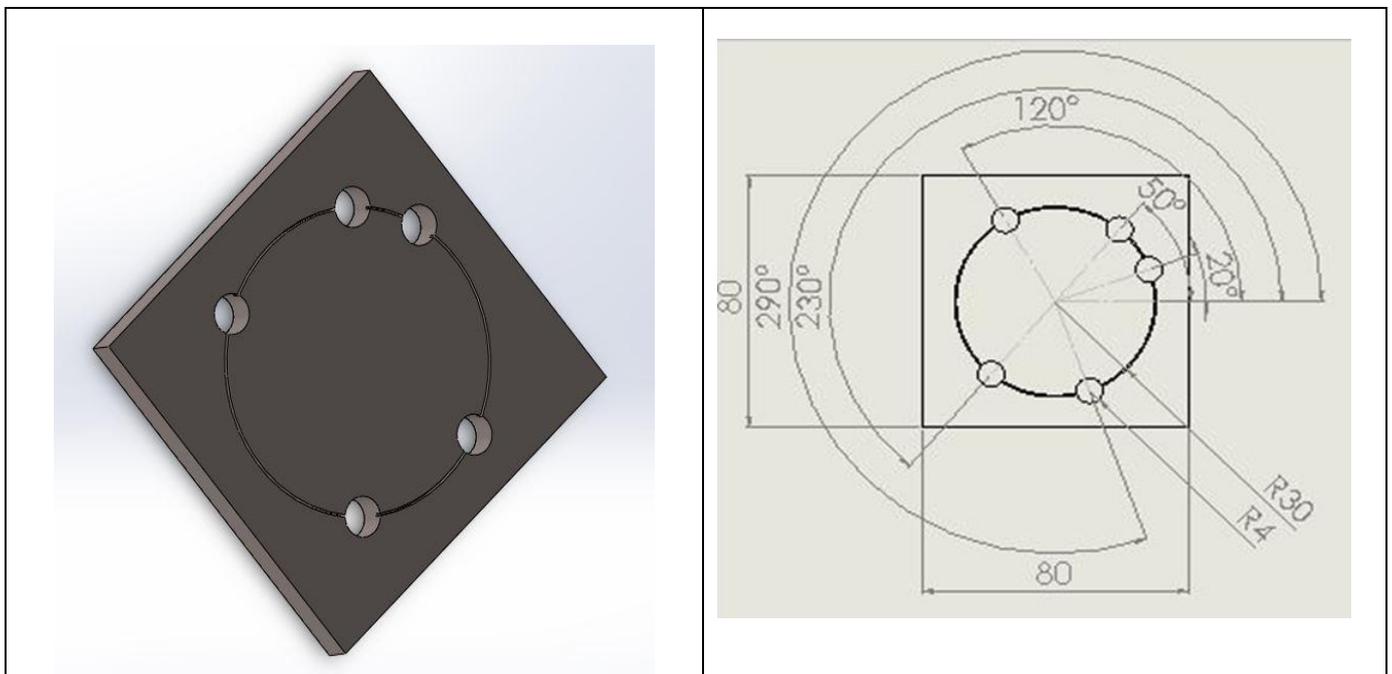


Fig. 6: Holes on a circle at variable angle

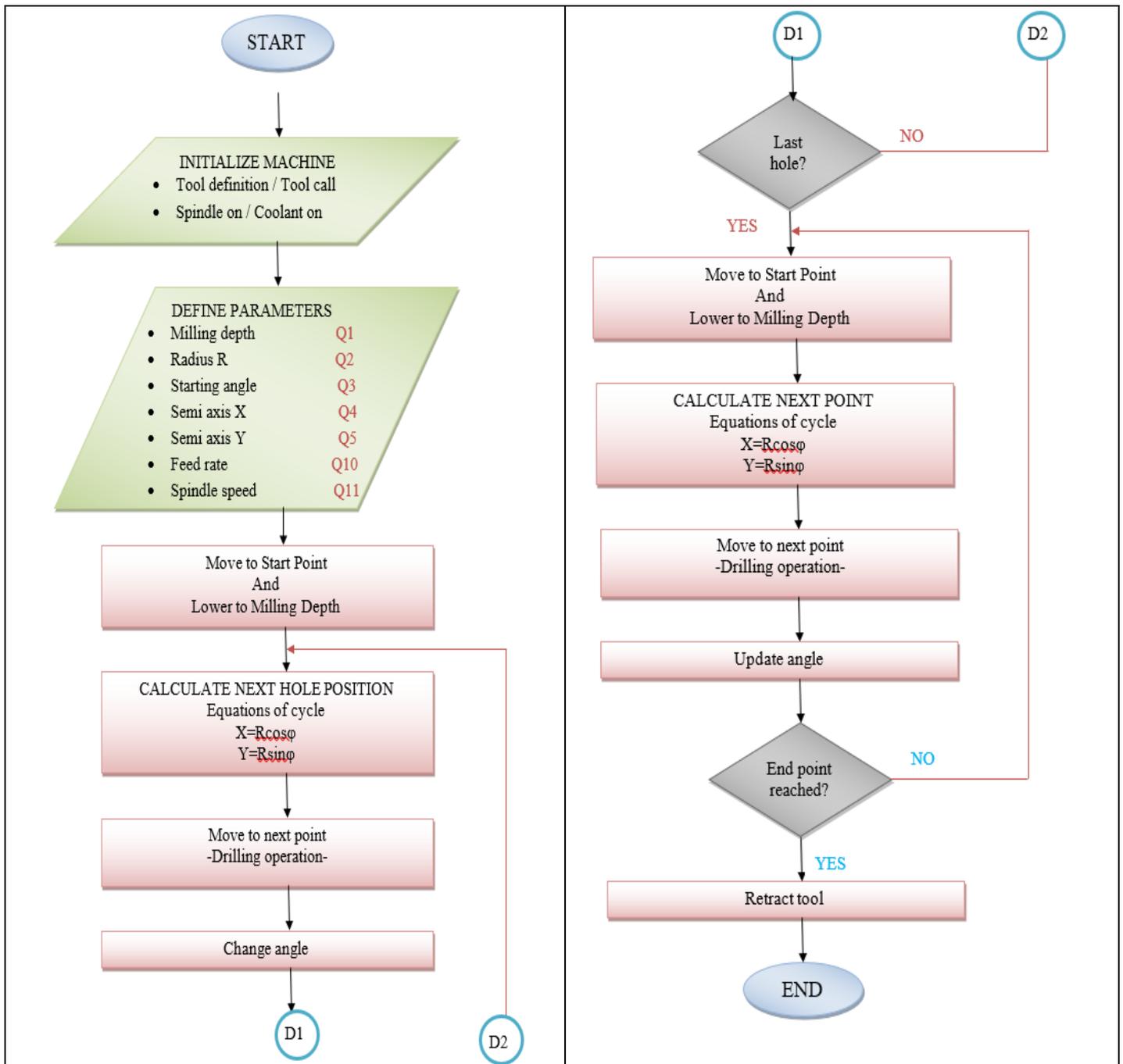


Fig. 7: Flow chart for drilling holes on a circle with variable number and angle

<pre> %4H G71 * N10 G30 G17 X-40 Y-40 Z-20* N20 G31 X+40 Y+40 Z+0* N30 T4 G17* N50 D00 Q1 P01 -25* N60 D00 Q2 P01 +30* N70 D00 Q3 P01 +20* N80 D00 Q4 P01 +0* N90 D00 Q5 P01 +0* N100 D00 Q10 P01 +100* N110 D00 Q11 P01 +1000* N80 M13* N80 G90 G00 X+Q4 Y+Q5 Z+2* N100 G98 L1* N110 D07 Q6 P01 +Q3* N120 D03 Q4 P01 +Q2 P02 +Q6* N130 D06 Q8 P01 +Q3* N140 D03 Q5 P01 +Q2 P02 +Q8* N150 G00 X+Q4 Y+Q5* N160 G01 Z+Q1 FQ10 SQ11* N170 G00 Z+2* N180 G98 L0* N190 Q3 = 50 N200 L1* N210 Q3 = 120 N220 L1* N230 Q3 = 230 N240 L1* N250 Q3 = 290 N260 L1* N250 G00 Z+100* N260 T3 G17 S1000* N270 M13* N260 G90 G00 X+Q2 Y+0 Z+2* N270 G01 Z-5 FQ10 S1000* N280 G98 L2* N290 D07 Q20 P01 +Q4* N300 D03 Q21 P01 +Q2 P02 +Q20* N310 D06 Q22 P01 +Q4* N320 D03 Q23 P01 +Q2 P02 +Q22* N330 G01 X+Q21 Y+Q23 F200* N340 D01 Q4 P01 +Q4 P02 +2* N350 D12 P01 +Q4 P02 +360.01 P03 Z N360 G01 Z+100* N370 M30* N99999999 %4H G71 * </pre>	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p>-Tool definition/ Tool call -Spindle on clock-wise, coolant on</p> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p>-Parameters definition (Milling depth, radius R, starting angle, axis X, axis Y feedrate, spindle speed)</p> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p>-Motion to starting point -Lower to milling depth -Subroutine1 - machining operation -Calculation of cosine, sine, X, Y for Holes</p> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p>-Change of angle -Resumption of Subroutine1</p> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p>-Retract in the tool axis -Change Tool</p> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p>-Spindle on clock-wise, coolant on -Motion to starting point -Lower to milling depth - Subroutine2 - machining operation -Calculation of cosine, sine, X, Y for Circle</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p>-Motion to the next point -Unfinished? Go to L2 -Retract in the tool axis -End program</p> </div>
---	---

List 2: CNC program for drilling any number of holes on a circle at desired angles

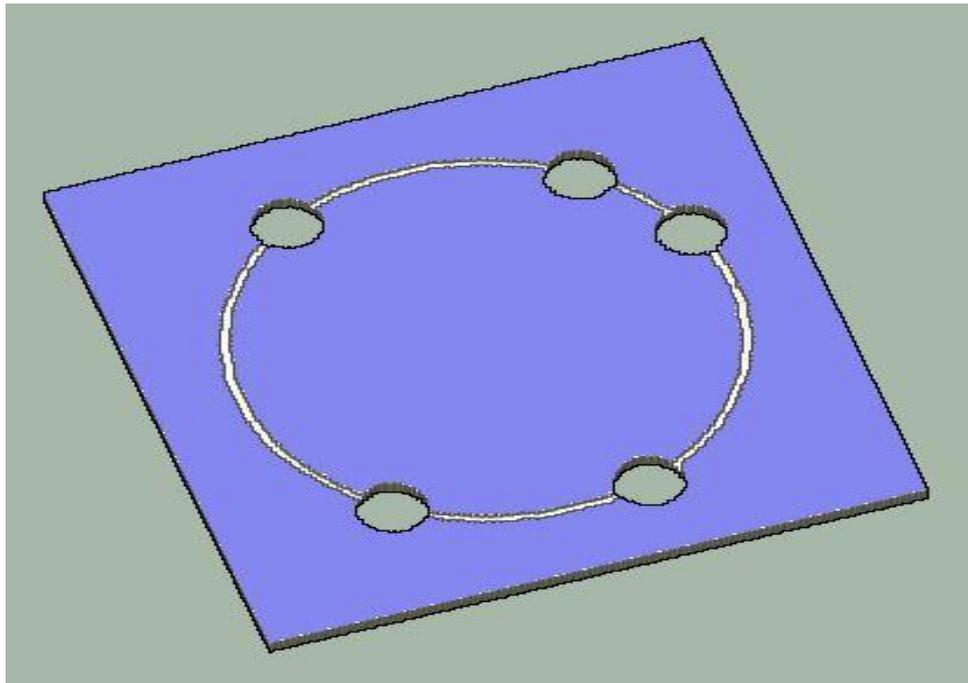


Fig. 8: Simulation test on Heidenhain – Five holes drilled on a circle with desired angles

Case 3 - Inventing canned cycles – A spherical pocket

Canned cycles provide a programming method of a CNC machine to accomplish repetitive machining operations, using the G/M code language. Their use, eliminates the need for many lines of programming, reduces the programming time and simplifies the whole programming process [7, 10]. All CNC machining controls come with a set of helpful machining canned cycles. However, the standard canned cycles are limited in number and capability, being unable to accommodate the needs of applications requiring something unusual—something that the standard canned cycles can't handle. Parametrization and sub-programming comes to remedy this situation, by offering to the programmer the ability to invent and develop his own canned cycles. The following work demonstrates the application of parametrization and sub-programming in developing a canned cycle for machining a spherical pocket.

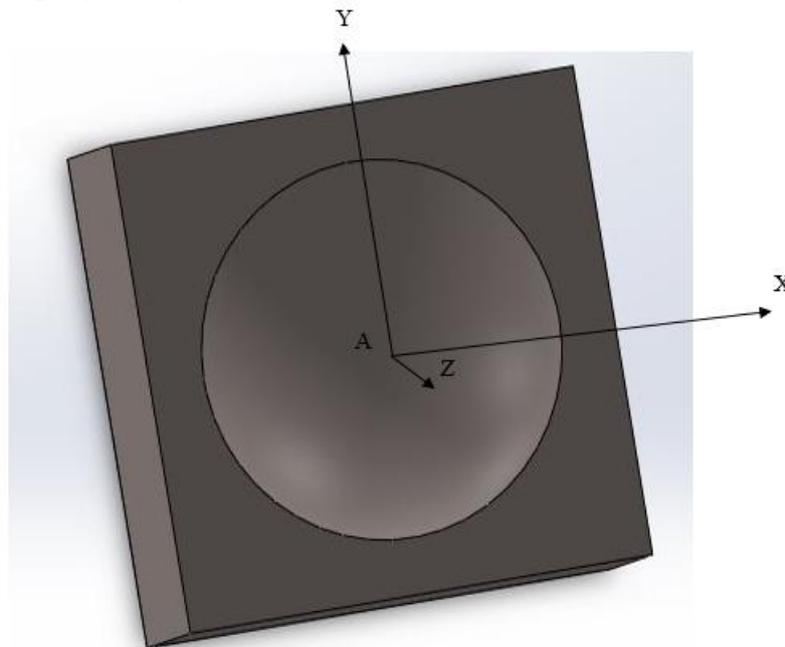


Fig. 9: A spherical pocket

The machining of the spherical pocket is achieved by repeating 2 successive motions which are programmed in a sub-program. These motions are:

- a. Full circular polar motion on XY plane
- b. Linear motion with selected step along the generatrix on XZ plane

It should be noted that the accuracy of the machining depends on the step selection along the generatrix. Figure 10 shows the flow-chart on which the program of List 3 was based on. Simulation result is shown in figure 11

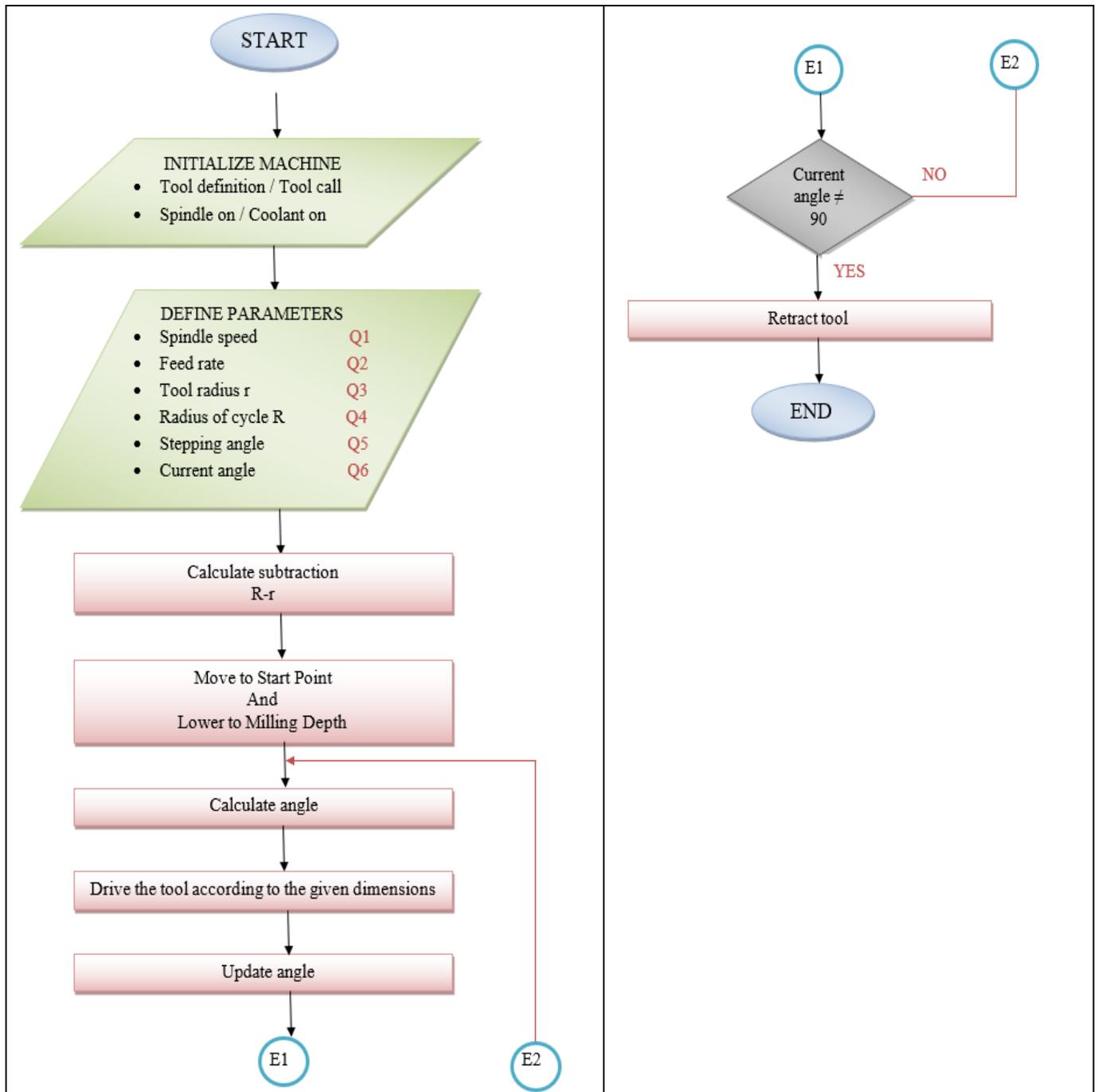


Fig. 10: Flow chart for machining of spherical pocket

```
%CANNEDCYCLES G71 *
N10 G30 G17 X-65 Y-65 Z-65*
N20 G31 X+65 Y+65 Z+0*
N30 T2 G17*
N40 M13*
N50 D00 Q1 P01 +1500*
N60 D00 Q2 P01 +200*
N70 D00 Q3 P01 +2.5*
N80 D00 Q4 P01 +50*
N90 D00 Q5 P01 +2*
N100 D00 Q6 P01 +0*
N110 D02 Q7 P01 +Q4 P02 +Q3*
N120 G90 G00 X+Q7 Y+0 Z+2*
N130 Z-Q3 FQ2 SQ1*
N140 G98 L1*
N150 G13 G91 H+Q5 G90 I+0 K+0*
N160 G02 G91 X+0 Y+0 G90 I+0 J+0
N170 D01 Q6 P01 +Q6 P02 +Q5*
N180 D10 P01 +Q6 P02 +90 P03 1*
N190 G90 G00 Z+100*
N200 M30*
N99999999 %CANNEDCYCLES G71 *
```

-Tool definition/ Tool call
-Spindle on clock-wise,
coolant on

-Parameters definition
(Spindle speed, feedrate, r-
tool radius, R-cycle radius,
stepping angle, current angle)
-Subtraction (R-r)

-Motion to starting point
-Lower to milling depth
-Subroutine1 - machining
operation
-Form a cycle
-Motion to the next point
-Unfinished? Go to L1

-Retract in the tool axis
-End program

List 3: CNC program for machining of a spherical pocket

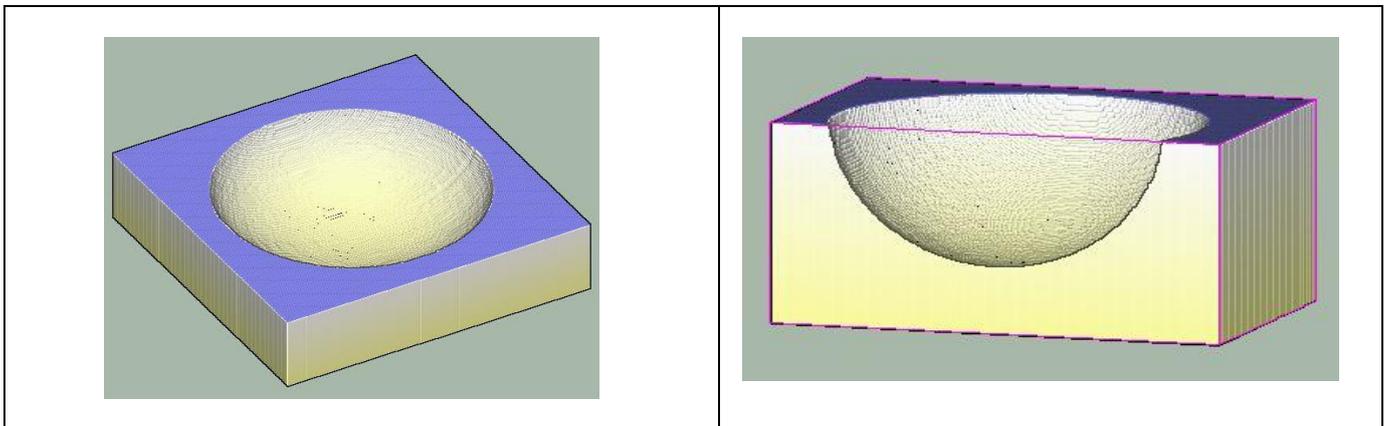


Fig. 11: Simulation test on Heidenhain – Machining of a spherical pocket

4. CONCLUSIONS

Programming method is of utmost importance in achieving effective CNC programs regarding flexibility, programming time and most critical to cycle time reduction. As it is demonstrated by the three representative case studies, parametrization and sub-programming techniques constitute a powerful programming tool in this direction. The generation of various motions with the desired each time accuracy, the development of programs capable of machining families of parts and the invention of new canned cycles enhancing the feature generating capabilities of CNC systems were all achieved by short flexible programs incorporating these advanced programming techniques.

BIOGRAPHY



Dr. Sotiris Omirou is a professor at the Department of Mechanical Engineering of Frederick University in Cyprus. He has over 30 years teaching experience in tertiary education. His research interests are focused on CAD/CAM and CNC manufacturing methods. He has published a significant number of scientific papers in international peer-reviewed journals and he is the author of two technical books.

REFERENCES

- [1] Abbas A.T. and Megahed S.M., "A General Algorithm for Drilling Holes Lying in a Matrix", Vol. 21, pp. 235-239, 2005.
- [2] Djassemi Manocher, "An efficient CNC programming approach based on group technology", Journal of Manufacturing Systems", 2000.
- [3] Djassemi Manocher, "A Parametric Programming Technique for Efficient CNC Machining Operations" Computers and Industrial Engineering, Vol. 35, pp 33-36, 1998.
- [4] Heidenhain iTNC 530 User's Manual.
- [5] Voelcker H. and Reuicha A., "Geometric Modelling of Mechanical Parts and Processes", Computer, Vol. 10, No.12, pp. 48-57, 1977.
- [6] Sava M. and Pusztai J., "Computer Numerical Control Programming", Prentice Hall, 1990.
- [7] Stenerson J. and Curran K., "Computer Numerical Control – Operation and Programming", Prentice Hall, 2005.
- [8] <http://www.mmsonline.com/articles/0402cnc.html>, "CNC Tech Talk" by Mike Lynch.
- [9] S.K Sinha, "CNC Programming using Fanuc Custom Macro B" 1st Edition, Kindle Edition , 2018
- [10] S. Omirou "Technology and Programming of CNC Machines" Kleidarithmos press, 2018