Automatic Generation of CNC Codes Based on Machining Features

Saleh M. Amaitik
Department of Industrial and Manufacturing Engineering, University of Misurata
Misurata, Libya
saleh.amaitik@eng.misuratau.edu.ly

***Abstract* – This paper presents a methodology for automatic generation of Computer Numerical Controlled (CNC) codes of milling operations. Machined features are used as the basis to input the required data for machining. The most commonly used machined features are studied in this paper. These features are: slot feature, pocket feature and hole feature. Cutting tool paths are determined based on the coordinates of the tool stops required to cut each machined features. These tool paths are used to generate the CNC part programs. Several examples were presented in the project to test the affectivity of the developed steps or algorithms. The results obtained are good enough to continue developing these algorithms for more complicated machined parts.**

***Index Terms*—machining feature, CNC, milling operation, tool paths.**

I. INTRODUCTION

Manual part programming can be time consuming, tedious, and subject to errors for parts possessing complex geometries or requiring many machining operations. In these cases, and even for simpler jobs, it is advantageous to use computer-automated part programming. A number of CNC part programming language systems have been developed to accomplish many of the calculations that the programmer would otherwise have to do. This saves time and results in a more-accurate an efficient part program [1].

In computer automated part programming approach to CNC programming, several aspects of the procedure are automated. In the future, it should be possible to automate the complete CNC part programming procedure. Given the geometric or feature model of a part that has been defined during product design, the computer automated system would possess sufficient logic and decision making capability to accomplish CNC part programming for the entire part without human assistance [1].

This can most readily be done for certain CNC processes that involve well-defined, relatively simple part geometries. Examples are point-to-point operations such as CNC drilling and milling machines. In these processes, the program consists of a series of locations in an x-y coordinate system where work is to be performed. These locations are determined by data that are generated during product design. Special algorithms can be developed to process the design data and generate the CNC program for the particular system. Automatic programming of this type is closely related to Computer Automated Process Planning (CAPP) [1].

The aim of this paper is to develop algorithms contain sufficient logic and decision making capability to automatically generate CNC part programs for certain machining features on prismatic parts.

II. CNC CODE GENERATION PROCEDURE

The developed procedure of the proposed algorithms for each machining feature is described in four main steps as follows:

1. Defining the engineering meaning of the machining feature considered.
2. Defining the parameters required to generate the CNC code for each feature.
3. Defining the tool path planning for each machining feature.
4. Building the rule-base structure for each machining feature.

In this paper, the approach to the automatic generation of CNC codes is based on machining features available on prismatic parts. Geometrical and technological information regarding the original blank and the finished workpiece that are required for making decisions in the generation process are available from the feature-modeler based on feature data model. The geometric models of the parts are developed using a set of machining features. The technological information attributes include material data, surface roughness data and accuracy data. Based on this knowledge, the proposed system generates CNC part programs required for manufacturing.

Figure 1 shows the framework of the proposed algorithms for CNC code generation. With this framework, the following main steps have been developed and are detailed as following. In the first step, the system receives part information which includes nominal shape information, machining form feature information, surface information, dimensions and tolerances information, and material information. The second step the system will determine the machining region from the available data which is useful for the moving tool economically. In the third step, the system selects the depth of cut. The selection is done by distributing the depth into several cuts according to the total depth and tool diameter. After that, the fourth step determines the number of passes required to achieve the dimensional and surface requirements. In the fifth step the system calculates the required coordinates for tool path planning which is followed by specifying tool path for each cut. In the final step, the system will automatically generate CNC codes [2].

The following subsections present a detailed description of the development procedure steps of the proposed algorithms.

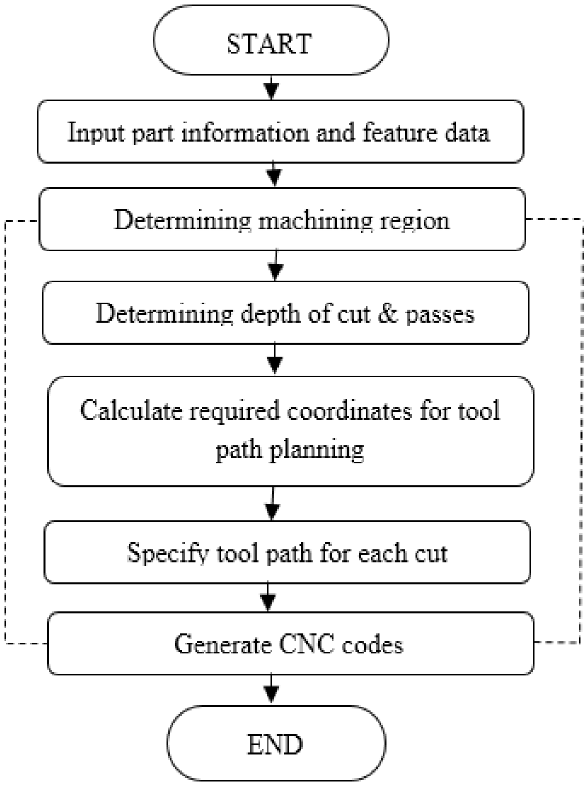

Figure 1. Proposed framework for CNC code generation

A. *Part and Features Data*

A block base shape is considered as the base shape of the manufactured part. Length, width and height are the dimensional parameters needed to define the size of the block shape. The block shape shall be positioned with the z-axis parallel to the height, the y-axis parallel to the width, and the x-axis to the length of the shape. The axes shall be positioned in the left corner on the top face of the part.

Features considered in the proposed algorithm are restricted to a set of 3-axis milling operation features. Features data contains the information necessary to identify shapes which represent volumes of materials that shall be removed from a part by machining or shall result from machining [3]. For demonstration purpose, this paper will limit its analysis to rectangular closed pocket. It is a type of pocket milling feature that is an enclosed area bounded by four sides with opposite sides equal in length and corners at 90 degrees. The orientation is at the center of the rectangle, the X-axis is parallel to the length of the rectangle and the Y-axis is parallel to the width. Figure 2 shows rectangular closed pocket [4].

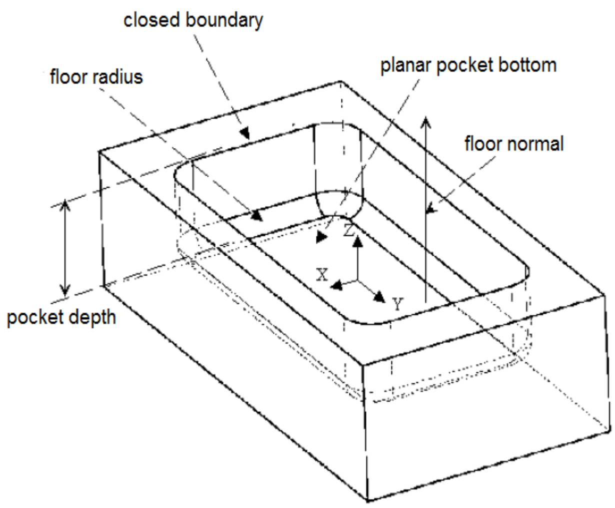

Figure 2. Rectangular closed pocket

B. *Machining Regions*

In this step, a two-dimensional machining feature region is created from the feature basic dimensions and its insertion point. The created machining region is used to facilitate the generation of the CNC code required to remove the feature volume. Figure 3 shows the created 2D machining region of rectangular closed pocket with round corners.

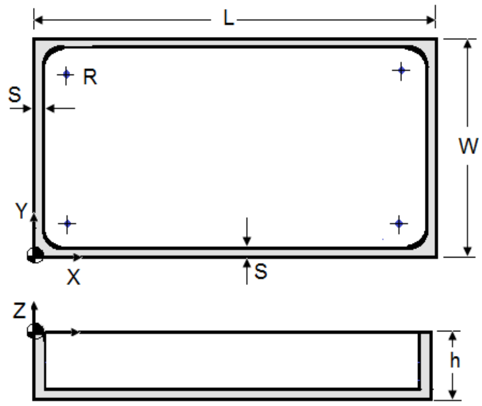

Figure 3. Machining region for rectangular pocket

C. *Depth of Cut and Number of Passes*

The toolpath generation for pocket feature is dependent on the number of roughing and finishing operations involved. Several rough cutting toolpaths will be generated which usually depends on pocket width and depth. Programmers often leave a small amount of material on the walls and bottom of the pocket for a final finishing pass. This amount depending on the size of the end mill. A finishing pass cleans up the pocket and leaves a good surface finish.

The depth of cut and number of cuts can be calculated using the following formula:

$$d = \frac{h}{m}. \quad (1)$$

Subjected to the following condition:

$$\frac{(m-1)R}{2} \leq h \leq \frac{mR}{2}. \quad (2)$$

Where d is depth of cut, m is integer representing the number of cuts (m=1, 2, 3, etc), h is pocket depth, and R is cutting tool radius.

To determine the number of passes per cut, the tool step-over can be computed using the following formula:

$$Q = \frac{W-2R-2S}{N}. \quad (3)$$

Subjected to the condition:

$$1.6R \leq Q \leq 1.8R. \quad (4)$$

Where Q is tool step-over, W is pocket width, S is finishing allowances, and N is integer representing number of passes per cut (N=1, 2, 3, etc).

D. Cutting Tool coordinates

Milling programs for machining features require calculations to determine the toolpaths used to machine the feature. Each toolpath consists of a series of coordinate locations that determine the tool's distance from programs zero. The program zero is the origin located on the machining features that provides the reference point for distances between machining features and the center of the tool. For rectangular pocket feature, program zero can be located on a feature left corner with the top face [5].

As shown in figure 4, pocket milling is often done with an end mill tool using zigzag fashion, i.e. going from one side to the other and back with a tool step-over calculated from (3).

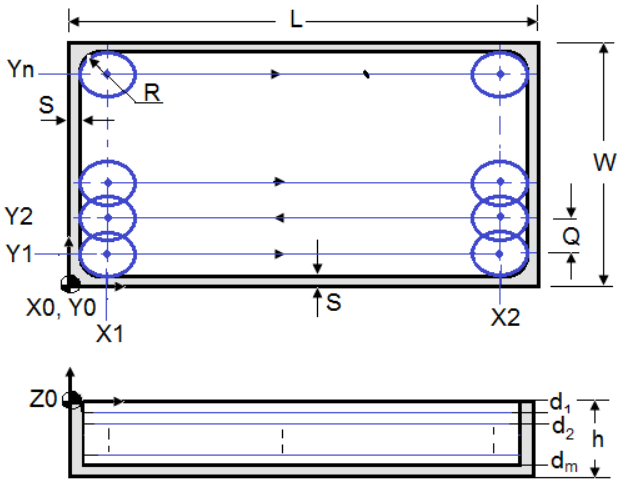

Figure 4. Rough machining of rectangular pocket feature

In pocket calculations, what needs is a series of simple X and Y coordinates. Each tool movement requires only the coordinates for the new location. The long series of X and Y codes moves the tool along straight lines that gradually enlarge the pocket. These codes are calculated based on feature data using the following formula:

$$X_{i,j}^{k} = \begin{cases} -(L-2(R+S)) & \text{if } i \bmod 2 = 0 \\ L-2(R+S) & \text{if } i \bmod 2 \neq 0 \end{cases} \quad (5)$$

$$Y_{i,j}^{k} = \begin{cases} Q & \text{if } i > 1 \\ 0 & \text{if } i = 1 \end{cases} \quad (6)$$

Where i is pass number per cut ($i=1,2,3,\ldots,N$), j is cut number ($j=1,2,3,\ldots,m$), and k is feature number.

E. Toolpath Planning

To machine the pocket feature and determne the toolpath, there is no outside location for the tool entering into the material. The tool has to plunge into the material along Z-axis at starting point, which usually in the bottom left corner of the pocket feature. Multi linear motion cuts will be carried out to the required depth using zigzag routine. The pocket will be roughed out by linear interpolation along pocket length and a tool step-over is used. The cutting continues back along pocket length and repeating to number of passes which depending on the pocket width. At the end, the tool will be retracted above the material and positioned to the center of the pocket for finishing.

F. CNC code generation

Since the relationship between the machining data covered in the machining feature and the CNC codes has been analyzed, it is possible to translate the pocket machining process into CNC codes using a code generation algorithm. This algorithm is used to scan through the data and produce machining actions into CNC codes. The code generation algorithm for pocket machining feature is shown in the flow chart in figure 5.

To demonstrate the application of the developed algorithm, a sample pocket feature is shown in figure 6. The pocket is 50 mm length, 35 mm width, and 5 mm depth. The pocket also has 5 mm radius in each corner. The cutting tool used is 10 mm diameter. The finishing allowances used is 0.4 mm and the pocket program zero is located at X_0=12.5 mm and Y_0=12.5 mm. The depth of cut is calculated from equations (1) and (2) as follows d=2.5 mm and number of cuts m=2. The number of passes and tool step-over are calculated from equations (3) and (4) as follows N=4 and Q=8.067 mm.

A CNC machine can only perform what it is programmed to do. Before a machining feature is made on a CNC machine, a programmer takes the numerical dimensions of the blueprint and turns them into step-by-step instructions. The CNC machine then uses the coordinates to perform these instructions, one after another, to make the machining feature. The coordinates required to create the pocket feature are calculated from equations (5) and (6). The results obtained are listed in Table 1.

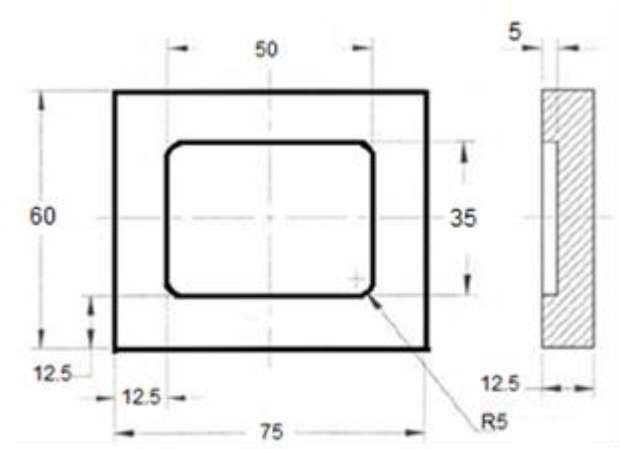

Figure 6. Sample drawing of rectangular pocket feature

TABLE 1. TOOL MOVEMENT COORDINATES

Feature	Cut	Pass	X	Y
1	1	1	38.4	0
1	1	2	-38.4	8.067
1	1	3	38.4	8.067
1	1	4	-38.4	8.067
1	2	1	38.4	0
1	2	2	-38.4	8.067
1	2	3	38.4	8.067
1	2	4	-38.4	8.067

A cutting path usually consists of a tool movement with rotating spindle and feed rate (when a cutter is used to machine a feature) or without a rotating spindle (when a cutter is moved to a new position for a new cut) [6]. For the pocket feature sample shown in figure 6, the automatically generated CNC code are:

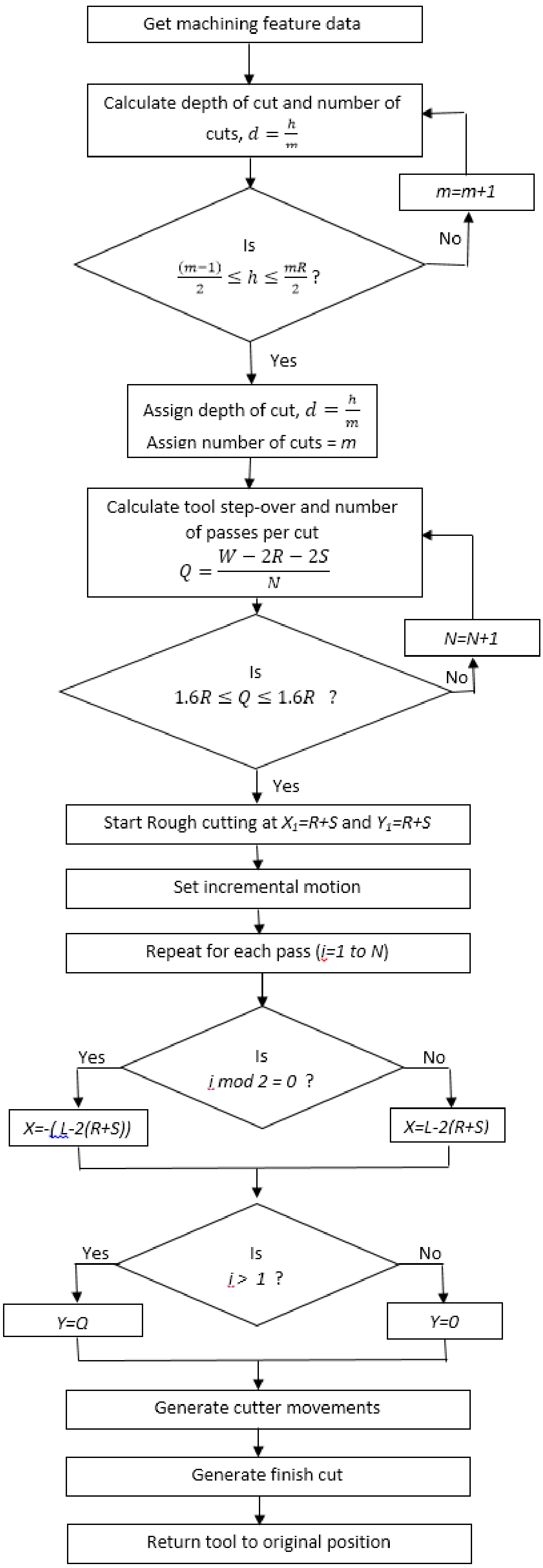

Figure 5. Flow chart of CNC code generation

```
N10 G21 G17 G40
N20 T01 M06                 (Roughing tool)
N30 S1200 M03
N40 G90 G00 X12.5 Y12.5     (Rapid to start point)
N50 G43 Z3 H01 M08
N60 G01 Z-2.5 F90           (First depth of cut)
(Roughing start)
N70 G91 X38.4               (Pass 1)
N80 Y8.067                  (Tool step-over 1)
N90 X-38.4                  (Pass 2)
N100 Y8.067                 (Tool step-over 2)
N110 X38.4                  (Pass 3)
N120 Y8.067                 (Tool step-over 3)
N130 X-38.4                 (Pass 4)
N140 G90 G00 Z3
N150 X12.5 Y12.5
N160 G01 Z-5                (Second depth of cut)
N170 G91 X38.4              (Pass 1)
N180 Y8.067                 (Tool step-over 1)
N190 X-38.4                 (Pass 2)
N200 Y8.067                 (Tool step-over 2)
N210 X38.4                  (Pass 3)
N220 Y8.067                 (Tool step-over 3)
N230 X-38.4                 (Pass 4)
N240 G90 G00 Z3
N250 M01                    (Optional stop)
N260 T01 M06                (Finishing tool)
N270 S1500 M03
N280 G00 X37.5 Y30          (Rapid to pocket center)
N290 G43 Z3 H02 M08
N300 G01 Z-5 F70            (Finish depth of cut)
(Finishing start)
N310 Y12.5
N320 X57.5
N330 Y42.5
N340 X17.5
N350 Y17.5
N360 X37.5
N370 G00 Z3
N380 X0 Y0 M09
N390 M05                    (Spindle off)
N400 M30                    (End program)
```

The CNC code generated for pocket machining starts with certain G&M codes that prepare the machine before the actual cutting takes place. These initial G-codes compensate for the part and tool offsets (G40), set default cutting plane (G17), type of units (G21), and coordinates system. M-codes are also used to perform miscellanous tasks. In the generated CNC code, M06 changes cutting tool, M03 turns the spindle on clockwise, and M08 turns coolant on.

The first pocket milling tool path in the program starts with rapid tool move (G00) to the starting point at X=12.5 and Y=12.5. Then, the tool feeds in the Z-axis to begin cutting. The first cutting block G01 Z-2.5 F90 plunges the tool into the material along Z-axis to the depth of 2.5 mm from the top of the part surface with feed rate 90 mm/min. What follows is a rough cutting with series of simple X and Y coordinates. Because G01 and the F codes are modes, they stay in effect until another code replaces them. Consequently, each following tool movement requires only coordinates for the new locations (N70-N230). The series of X and Y coordinates move the tool along straight lines that gradually enlarge the pocket. An optional stop (M01) to pause the machine after the sequence and waits for the operator to start the next cycle.

Once the pocket is roughed out, another tool (or even the same tool in some cases) can be used to finish the pocket to its final size (N260-N270). Typical strating tool position is at the pocket center, N280 G00 X37.5, Y37.5, as shown in figure 7 [7].

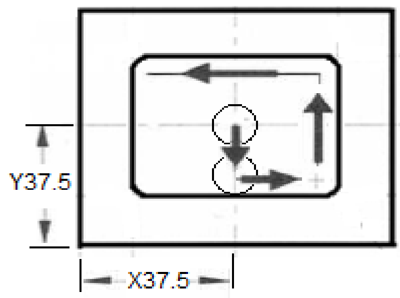

Figure 7. Typical finishing tool path for a rectangular pocket feature

III. CONCLUSIONS

With the emerging trends of digital manufacturing and factory automation, it becomes essential to automate the CNC code generation. The paper presents a methodology for automatic generation of CNC codes based on machining features. It has highlighted the need for automatic generation of CNC code to enhance the link between process planning and machining, and to minimize lead times in the generation of CNC codes of machined parts. The work presented in this paper can be further expanded to include more machining features with cutter path simulation.

REFERENCES

[1] M. Hou and T. Nfaddis, "Automatic tool path generation of a feature-based CAD/CAPP/CAM integrated system", *Int. J. of CIM*, Vol. 19, No. 4, pp. 350-358, 2006.

[2] A.M. Pinar PINAR, A. Gullu, "Time Minimization of CNC Part Programs in a Vertical Machining Center in Terms of Tool Path and Cutting Parameter Criteria", *Turkish J. Eng. Env. Sci.*, Vol. 29, pp. 75-88, 2005.

[3] S.M.Amaitik and S.E. Kilic, "STEP-based feature modeler for computer-aided process planning", *Int J Prod Res,* Vol. 43, pp. 3087–3101, 2005.

[4] International Organization for Standardization, ISO 10303-224 Industrial Automation Systems and Integration-Product Data Representation and Exchange - Application Protocol: Mechanical product definition for process planning using machining features, 2000.

[5] R.M. D'Souza, "On setup level tool sequence selection for 2.5-D pocket machining", *Robotics and Computer-Integrated Manufacturing*, Vol. 22, pp. 256–266, 2006

[6] W. Xiang, C. W. Chuen, C. M. Wong and L. H. Yam, "A Generative Feature-Based CAPP/CNC System for Hydraulic Manifold Blocks", *Int. J. Adv. Manuf. Technol.*, Vol. 19, pp. 805-811, 2002.
[7] Peter Smid, *CNC Programming Handbook*, 2nd ed., New York, Industrial press, Inc., 2003.

BIOGRAPHIES

Saleh M. Amaitik is an associate professor in the Department of Industrial and Manufacturing Engineering at the University of Misurata. He received His MSc degree in Manufacturing Engineering from the University of Garyounis in 1994 and PhD degree in Mechanical Engineering from Middle East Technical University in 2005. His main research interests are Computer Aided Process Planning - CAPP, Feature-based Modeling, CAD/CAM Integration, Product Data Modeling using STEP and Intelligent Manufacturing Systems