International Journal of Engineering & Technology Sciences (IJETS) 1(4): 218-225, 2013
ISSN 2289-4152

Research Article

Parametric Programming in Feature-Based Machining

M.A. Razak [a,*], M.R. Ibrahim [a], S. Sulaiman [a], A. Jusoh [b], A. Zakaria [b]

[a] Manufacturing Section, Universiti Kuala Lumpur Malaysian Spanish Institute, Kedah, Malaysia
[b] Universiti Kuala Lumpur Institute of Product Design and Manufacturing, Kuala Lumpur, Malaysia

* Corresponding author. Tel.: 604 403 5199; fax: 604 403 5201
E-mail address: alhapis@msi.unikl.edu.my

A b s t r a c t

Keywords:
Parametric programming
Feature-based machining
CNC
Macro

This paper evaluates the feasibility of using parametric programming in the implementation of feature-based machining. Custom Macro B was used in this study. Macro programming technique comprises of a main program and subprograms. NC program for a basic type of machining feature namely pocket was first generated using macro and CAM system. Macro program utilizes very fewer blocks than CAM generated program. The different between both programming methods are discussed. Finally, an integrated feature based machining system is proposed.

Accepted:17 July 2013

1. Introduction

Machining feature is the remnant volume after subtracting the designed feature from the raw material [1, 2, 3, 4]. Rough machining feature is the volume after roughing is subtracted from the raw material. The machining feature should include the topological and geometric information of the machining region to offer preliminary data for generating the machining strategy. Examples of machining features include faces, pockets, holes, and slots.

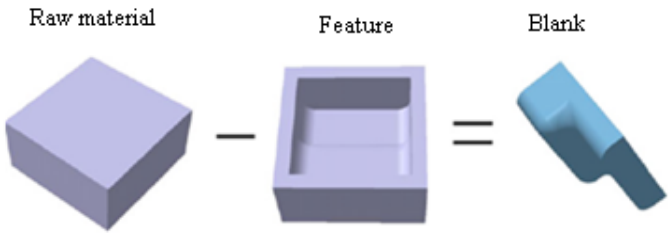

Fig. 1: Example of machining feature

The machining features used to make a specific part may be instances of a fixed library of parametrically defined features, or they may be defined without a library by making boundary or constructive solid geometry representations [5]. There are many alternative definitions of machining features. To machine a part, both the machining features and the machining operations must be defined, and the operations must be sequenced. A feature-based machining system strengthening the function of material removal machine to work in more intelligent and helps the user to shorten the pre-machining setup time [6]. While a generative process planner enables a computer numerical control (CNC) machine to automatically define the operations for cutting the designed features [5]. Without machining feature, program is input manually and the cutting path will based on the coordinate given. Programming will take longer time and the machining efficiency will not achieve as good as feature-based.

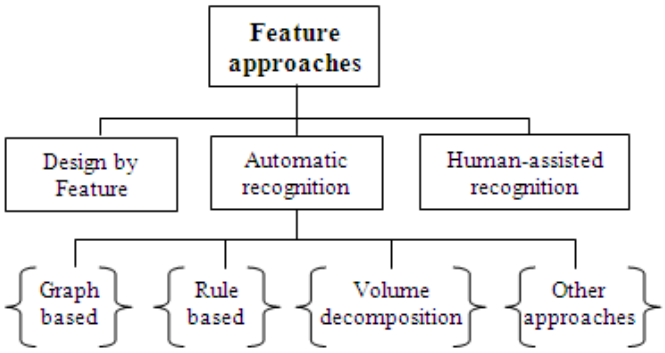

Fig. 2: Feature approaches classification [7]

Feature-based machining can be implemented in three ways as shown in Fig. 2. Design by feature [8] is a technique by which the designer can document and communicate the design intent. For instance to define a hole, designer can specify radius, depth and location parameters. The second approach is by means of pattern recognition. This method is largely used in computer-aided manufacturing (CAM) software. However generating NC program by commercial CAM software will result in longer part program besides utilizing a large memory space in CNC machine controller [9]. And the last approach is by human assisted recognition. Based on human-assisted recognition, this paper aimed in evaluates the feasibility of using parametric programming in the implementation of feature-based machining.

2. Parametric Programming by Macro

Parametric programming [10] is applied to CNC operations in generating a single CNC program for parts with similar design, inventing macros for machining custom design features, and developing subprograms for a group of parts that are not similar in design but require similar machining operations. Parametric programming can significantly reduce the part programming time and these applications particularly fit group technology manufacturing in which similar parts are grouped into part families and then processed by a number of machine tools within a cell or by a single multi-tasking machining center. Different controller manufacturer provide different version of parametric programming such as User Task (from Okuma), Q Routine (from Sodick), and Advanced Programming Language (APL) (from G&L). In Fanuc or Fanuc compliant CNC controller, parametric programming can be implemented by Custom Macro B.

Macro is very similar to subroutine. The different is that macro enables user to specify arguments and control the variables [11]. With macros, repetitive cycle can be defined. It may be considered as the highest level of NC programming [12]. This technique is more powerful and flexible. In the conventional CNC programming, there is limitation in terms of function of each G-Code. Designed in separated programs, macros can be called by the main program or other macros using macro number.

Fig. 3 shows the path taken by G&M code interpreter modules [13] which are part of a CNC system to execute the part program. The conventional G&M codes have to pass through a parser, an executor and a path generator. Macro program will not go through the normal executor. Macro executor interprets and executes macro commands included in an NC part program.

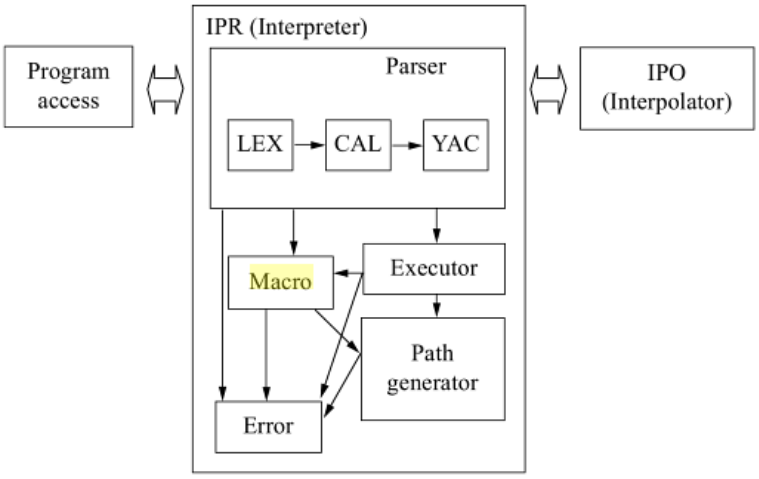

Fig. 3: Code interpreter modules [13]

Another advantage of macro is since it is similar to the BASIC language; user can make specific functions that are not provided by the CNC maker by using macro language.

3. Methodology

Fig. 4 and 5 show the experimental workflow and graphic simulation in CAM. The most commonly machining feature in tool and die making; pocket was used in this study. Feature was first created in CAD and its NC program generated by a commercial CAM. NC programs are also generated by manual programming using macros. Machining simulation was then performed on a CNC milling machine controller to prove its functionality. The experiment was carried out on Fanuc Robodrill α-T14íFse machine with Fanuc Series 31í-Model-A controller. Different sizes of pockets (width, length and depth) were studied. Machining parameters such as spindle speeds and feedrates were kept constant. High speed steel (HSS) straight end mill of 10mm diameter was used for this purpose.

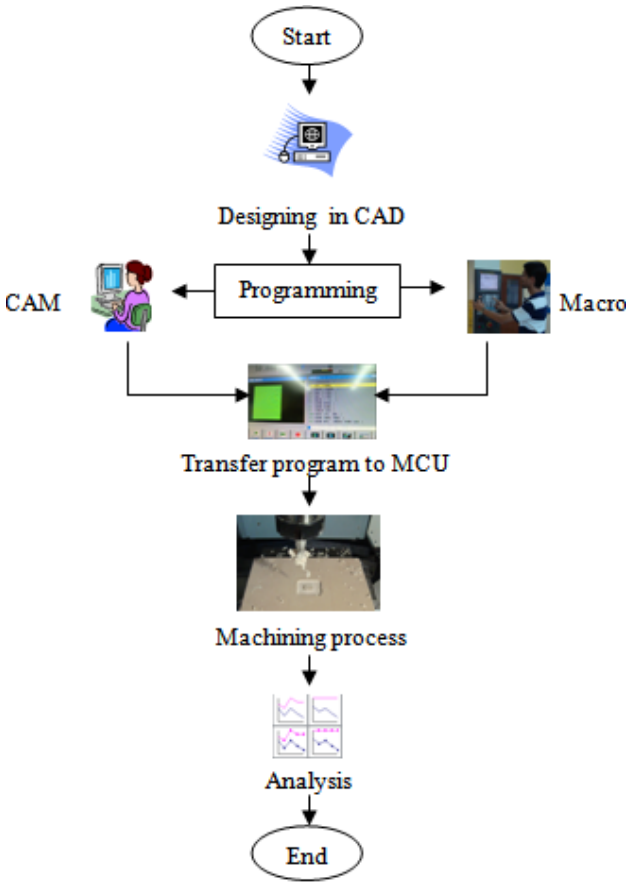

Fig. 4: Experimental workflow

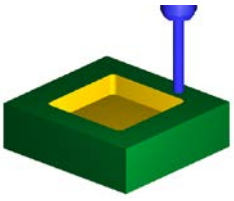

Fig. 5: Graphic simulation in CAM

In macro program, variable numbers represent specific information declared by user or referred to earlier part programs. For instance, #1 is representing the pocket length and #2 refers to pocket width. To generate a new program for different design, the user needs only to redefine the variables in the macro. As there is no new program added, the memory used in the controller remains more or less constant.

There are main program and macro program in macro approach. The main program prompts user to select the type of feature required. Once selected, user is requested to key in details of the feature. These include its center location, pocket length, depth, corner radius, depth and width of each cut, tool number, cutting feedrate and spindle speed. The different between macro and CAM generated NC programs are discussed in the next section.

3. Results and Discussion

```
N126 G1 Z69.025
N128 X-5.
N130 Y-120.
N132 X5.
N134 Y-110.
N136 Z79.025
N138 G0 Z80.
N140 X2.5 Y-112.5
N142 Z79.025
N144 G1 Z68.05
N146 X-2.5
N148 Y-117.5
N150 X2.5
N152 Y-112.5
N154 Z78.05
N156 G0 Z80.
N158 X5. Y-110.
N160 Z79.025
N162 G1 Z68.05
N164 X-5.
N166 Y-120.
N168 X5.
N170 Y-110.
N172 Z78.05
```

a) CAM generated NC program

```
WHILE[#118LT[#111-#117-#23]]DO1
#118=[#118+#10]
IF[#118GE[#111-#117-#23]]THEN
#118=[#111-#117-#23]
Y[#102-[#112-#117-#23]]
X[#101-[#118-#23]]
Y[#102+[#112-#117-#23]]
X[#101+[#118-#23]]
Y[#102-[#112-#117-#23]]
X#101
END1
```

b) Macro program

Fig. 6: Different between CAM generated and macro program

In the conventional CNC part program, G-Codes are used for specific functions. For example, modal G codes such as G1, G2 and G3 are for linear interpolation, clockwise interpolation and counter-clockwise interpolation respectively while the canned cycle like G81 is for drilling a hole. The same codes may be used repeatedly in order to create or to cut a given machining feature. This technique is applied in CAM to generate NC program as shown in Fig. 6 (a). Therefore, the size of NC programs generated by commercial CAM system is usually very large. When using parametric programming,

the routines can be written as simple as shown in Fig. 6 (b) and makes it much shorter compared to CAM.

With CAM, the part program can be thousands of blocks in size. It happened because of CAM generates the program based on cutting path coordinate. The larger the machining part, the longer the part program. Also the deeper the cutting depth, the larger the program sizes. On the contrary, the macro program size for the same feature can be constant for different sizes. This is due to the fact that the size changes can be done by simply redefining the feature variables. These variables are defined once only but their values can be changed according to the feature. If redefined in main program, the new arguments will be used by the called macros. Fig. 7 (a) shows macro call program for a pocket.

```
……
G65P0147X#24Y#25A#1B#2C#3D#7U#21F#109W#23T#20
……
```

a) Macro call for rectangular pocket

```
O0147 (RECTANGULAR POCKET)
N100G0G90X#101Y#102Z5.(POCKET CENTER LOCATION)
G1Z[#510-#12]F#9
N200WHILE[#118LT[#111-#117-#23]]DO1(COND_DO
POCKET)
#118=[#118+#10]
IF[#118GE[#111-#117-#23]]THEN#118=[#111-#117-
#23]
Y[#102-[#112-#117-#23]]
X[#101-[#118-#23]]
Y[#102+[#112-#117-#23]]
X[#101+[#118-#23]]
Y[#102-[#112-#117-#23]]
X#101
END1
N250IF[#12GE#3]GOTO300(COMPLETE Z ROUGH)
#12=[#12+#21]
IF[#12GE#3]THEN#12=#3(COMPLETE Z ROUGH)
#118=0(START NEW POCKET CYCLE WITH NEW DEPTH)
GOTO100
N300IF[#23EQ0]GOTO400
G1X[#101-[#111-#117]]Y[#102-[#112-
#117]](FINISHING)
Y[#102+[#112-#117]]
X[#101+[#111-#117]]
Y[#102-[#112-#117]]
X[#101-[#111-#117]]
M99
```

b) Pocket macro (in machine memory)

Fig. 7: Macro program for pocket

The block “G65P0147X#24Y#25A#1B#2C#3D#7U#21 F#109W#23T#20” can be translated as call a pocket macro from memory O0147 with the parameters X, Y, A, B, D, U, F, W, and T. Therefore for a new pocket feature, user needs only to change these parameters. The pocket macro in Fig. 7 (b) is not visible to the user. It stored in machine memory.

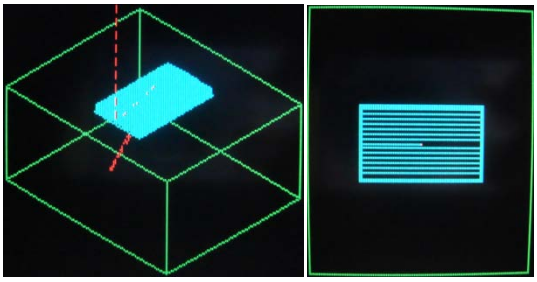

Fig. 8: Simulation on CNC controller

Unlike macro, CAM software will generate a new program for the same feature although there are minute changes in its parameters. This is where the user can apply macros as alternative. By controlling the variables, no more programs to generate for the same feature. Unfortunately many CNC machines come with a limited working memory. Consequently for a machine without a DNC facility, transferring a new part program is difficult without first deleting some old programs.

Other advantage of macro is that, users are allowed to create their own canned cycles, automatic determination of feedrate and spindle speed, and creation of new alarms other than that provided by the controller manufacturer. These all are can easily be done by macros. However, few CNC users are aware about the existence of macro and know how to create macro program. In fact, people in industries are continuing the same daily routine with using CAM to generate NC program. This is mainly because of creating macro program requires individual programming skills and hence users are often eschew this task. The good news is currently there are softwares available in the market to simulate limited capability of macro program.

4. Conclusions

From the result, it is confirmed that parametric programming can be used in the implementation of feature-based machining. For a given feature, commercial CAM system generates large program blocks hence there is an increase in program memory. With the macro approach however, one can expect comparatively much smaller program size. Consequently, in practice the old files need no longer to be deleted very frequently before transferring new programs. Although this could be the main advantage of macro program over conventional CAM system, further analysis is required particularly in terms of surface finish, machining time or even ease of programming. Finally, it is envisaged that an integrated feature based machining system can be realized by having macro programs integrated GUI (graphical user interface). Hence a post processor is no longer required for NC program generation.

References

[1] E.Y. Heo, D.W Kim, J.Y Lee, C.S. Lee, F.F. Chen. High Speed Pocket Milling Planning by Feature-Based Machining Area Partitioning. *Robotics and Computer-Integrated Manufacturing*, 2011; 27:706–713.

[2] A. Cardone, S.K. Gupta, A. Deshmukh, M. Karnik. Machining Feature-based Similarity Assessment Algorithms for Prismatic Machined Parts. *Computer-Aided Design* 2006; 38:954–972.

[3] X. Xu, S. Hinduja. Recognition of Rough Machining Features in 2.5D Components. *Computer-Aided Design*, 1998; 30:503–516.

[4] Y.J. Tseng. Fixturing Design Analysis for Successive Feature-based Machining. *Computers in Industry*, 1999; 38:249–262.

[5] Kramer, Thomas R. Submitted to National Institute of Standards and Technology, United State of America 1987.

[6] M. Razak, A. Zakaria. Review on the Evolutions of CNC Programming Methods. *2nd Colloquium on Manufacturing Technology*, Kulim 2010.

[7] A. Mokhtar, A.T. Bina, M. Houshmand. Approaches and Challenges in Machining Feature-Based Process Planning. *4th International Conference on Digital Enterprise Technology* 2007, London.

[8] S. Venkataraman. Integration of Design by Features and Feature Recognition. Arizona State University, Arizona 2000.

[9] M. Razak, A. Zakaria. A Framework for a Feature Based Machining using Macro. *Applied Mechanics and Materials Journal*, 2012; 110:1711–1715.

[10] M. Djassemi. A Parametric Programming Technique for Efficient CNC Machining Operations. *Computer and Industrial Engineering*, 1998; 35:33–36.

[11] User's Manual. Common to Lathe System/Machining Center System. *Fanuc series 30i/300i/300is-Model A*, Vol. 1-3.

[12] M. Razak, A. Jusoh, A. Zakaria. Feature-Based Machining Using Macro. *International Conference on Machine Design and Manufacturing Engineering*, Paris 2012.

[13] S.H. Suh, S.K. Kang, D.H. Chung, Ian S. Theory and Design of CNC Systems. *Springer Series in Advanced Manufacturing*, 2008; 82–93.