Journal of Basic and Applied Engineering Research
p-ISSN: 2350-0077; e-ISSN: 2350-0255; Volume 2, Number 18; July-September, 2015, pp. 1566-1569

http://www.krishisanskriti.org/Publication.html

Computer Numerical Control Machines: An Introduction to Parametric Programming using Custom-Macro B

Mohd Asif Hasan

University Polytechnic, Faculty of Engineering and Technology,
Aligarh Muslim University (AMU), Aligarh (India) - 202002
E-mail: hasan_in@hotmail.com

Abstract—*This paper discusses a lesser known but highly potential and useful method of programming the Computer Numerical Control (CNC) machines known as Parametric programming. Parametric programming can be compared to any computer programming language and possesses both computer related features as well as CNC related features. This Parametric programming language is provided in the CNC controller and can be used along with the manual part programming techniques i.e. at the level of G/M code. Parametric programming supports many computer related features like variables, arithmetic, logic statements, looping, etc., and also possesses many CNC related features like tool offsets, axis position, alarms, handling of devices like probes and edge finder, etc. This paper also exclusively considered the most popular version of parametric programming i.e. Fanuc's Custom Macro B. Various application fields of Custom Macro B have also been examined. At the end of the paper a sample program has been illustrated for a quick but limited glance of the Parametric programming format. The paper comes to an end with a discussion and conclusion.*

Keywords*: Computer Numerical Control Machines; CNC Machines; Part Programming; Parametric Programming; Macro Programming; Custom Macro B.*

1. INTRODUCTION

Parametric programming is programming the CNC machines making use of the capabilities of the controller in making arithmetic calculations, feeding formulae, storing and assigning values to the system variables as well as Local and Common variables, etc. These CNC controller variables are just like algebra variables that can be used along with G-code. They can be assigned values, and when referred in the part program, give back the last value they were assigned.

Parametric programming is also known as Macro Programming. Macros are simple part-programs which reside in the memory of the controller and are called using a specific code for that macro. All sort of canned cycles are basically Macros [1-3].

Parametric programming can be compared to any computer programming language like BASIC, C Language, and PASCAL. However, this programming language resides right in the CNC controller and can be accessed at G code level. Thus, parametric programming techniques can be used along with the manual part programming techniques [4].

Parametric programming possesses both Computer-related features as well as CNC specific features. Computer-related features like variables, arithmetic, logic statements, and looping are available in Parametric-programming. Like other computer programming languages, parametric programming is also available in several versions. The most popular is Custom Macro B (used by Fanuc and Fanuc-compatible controls). Others include User Task (from Okuma), Q Routine (from Sodick), and Advanced Programming Language [APL] (from G& L).

In addition to having many computer-related features, most versions of parametric programming have extensive CNC-related features. Custom macro, for example, allows the CNC user to access many things about the CNC control (tool offsets, axis position, alarms, generate G codes, and program protection) right from within a CNC program. These things are impossible with only normal G code programming techniques [1, 5, 7].

2. PARAMETRIC PROGRAMMING USING CUSTOM MACRO B

The most popular version of Parametric programming is Fanuc's Custom Macro B. Many control manufacturers (including Haas, Mistubishi, Mazak, Yasnac, and Seikos) use Custom Macro B as their version of Parametric programming.

Although for most CNC controls, Custom Macro B is an option for parametric programming but it doesn't come standard with the controller. However, many machine tool builders include Custom Macro B in the standard package of

options they include with the machines they sell, especially if the machine has some special accessory like a probing system.

Most CNC related features of parametric programming are handled with system variables. With Custom Macro B, system variables range in the #1000 through #6000 series. Like any other variable in Parametric programming, most system variables contain numeric values. However, each system variable has a fixed function, and can only be used to reference its related function.

Variables are expressed in the program by a numerical value preceded by the pound sign for example #1000, #2000. Values can be input manually into a variable register. Values can also be assigned via the part program, for example #2000=1.5, #4000= #2000+1. Variables can only be Mathematical Quantities (numbers).

The actual names of system variables vary from one version of parametric programming language to the next, as does the actual usage for given system variables. The names and usage for certain system variables vary even among control models within a given control manufacturer's product line. For example, With Fanuc's custom macro B the control model and even how the control is equipped with certain control options determine the actual names of system variables. For this reason, one must refer one's own control manufacturer's programming manual to determine which system variables are available with the particular CNC control/s.

So one other important point is how to determine if a given control has Custom Macro B? One universal way is to perform a simple test in the Manual Data Input (MDI) mode. In the MDI mode, simply enter and execute the command:

#101=1

One may expect this command to be invalid as most Fanuc controls require terminating MDI commands with a semicolon (;). But, if the control has Custom Macro B, it will execute this command without generating an alarm. If it does not, an alarm will be sounded something like "syntax error" or "unrecognizable address".

Additionally, if the machine has Custom Macro B, one should be able to find a display screen that shows the Custom Macro variables. With Fanuc controls, it will either be in the Offset or Setting display screen [2-3]. Common Interpretation of Fanuc controllers of variables ranges is as follows:

Variable No.	Type of Variable	Function (Specific Machine Manual must be referred before any use. Don't assign anything to these unless one is sure what they do!)
#0	Null	Its value is always "null", which means "no value" e.g., #100 = #0 (Make #100's value "null")
#1 - #33	Local Variables	Local variables are used to pass arguments to macros and as temporary scratch storage. The controller doesn't remember the value of any local variables when the power is turned off. Local variables could be used for nesting and with subprograms.
#100 - #199 #500 - #999	Common Variables	Common Variables are shared by all your macro programs. When the power is turned off on the controller, variables from #100 - #199 are cleared to null. However, variables from #500 - #999 remember their values for the next time power is turned on.
#1000 and up	System Variables	System variables can be used to tell and know things about what the controller is doing, such as the input/output signals, tool/fixture offsets, alarm generation, current/absolute position, feed, federate override, etc.

The syntax for a variable is the pound sign followed by a number which may be of as many digits as the machine's controller supports to identify the variable. For example, it may be written "#5 = 50" to assign the value "50" to the variable "#5". These variables are of great use when you might want to change a value of these variables in different situations resulting in all other values based on these variables will be updated automatically.

Formulas used in parametric programming are called *Expressions*. They work pretty much as one would expect algebraic formulas to work with a few things to keep in mind:

- Uses square brackets ("[" and "]") instead of parenthesis. For Example: #5 = [#2 * #3/#1] + #4

- The standard operators are add ("+"), subtract ("-"), multiply ("*" asterisk and not "x"!), and divide ("/").

In parametric programming the flow of execution of part program can be done in different sequence by conditional instructions e.g. *IF..*, *THEN..*, GOTO.., etc. In addition, a multiple repetition of the part program segment is possible using *REPEAT* loop, *WHILE* loop etc. [8]. A list of Fanuc's Custom Macro B Variables including types of variables, arguments, arithmetic commands, control command, macro call etc. can be accessed through the CNC information webpage [9]. However, there is a list of other operators and functions that may be somewhat control-dependant. Make sure to check machine manual to confirm if your control supports them and if so just how they work.

3. APPLICATIONS OF CUSTOM MACRO B

Applications for Custom Macro B [2-3, 6, 10] in particular and Parametric programming in general may be categorized into five basic categories:

3.1 Family of Parts

Almost all companies have at least some applications for custom macro that fit into this category. A single drawing for the similar components or family of parts to be manufactured is provided generally in a format in which the drawing is dimensioned with variables right on the print and the part-programmer must make calculations or reference a chart on the drawing to come up with values needed in the program for the various similar components to be manufactured.

With Parametric programming, a part program is developed which is similar to an application form which you fill as enquired in the form. In contrast to manual part programming, Parametric-programming makes use of variables to store the various key numerical values or dimensions of a component to be manufactured which could be easily altered as required. Many other dimensions of a component are a result of some calculations on these key numerical values or dimensions. In parametric programming, these calculations are made to be performed by the controller on the variables storing these key numerical values and not directly on these numerical values. Since the calculations in the parametric programming are made on the variables, any desired change in the numerical value of the key dimension will update all other dimensions. As in a family of parts the sequence of operations as well as ratio of parts size is same, this feature of Parametric Programming is utilized in developing programs for family of parts. A single parametric program could serve for the entire family of parts by merely making some desired changes in the values of the stored key variables.

3.2 User-created Canned Cycles

Even if an industry doesn't have a perfect family of parts application for custom macro, surely it will have at least some work pieces that require similar machining operations.

Canned cycles provide a programming method of a CNC machine to accomplish repetitive machining operations using the G/M code language. Essentially, canned cycles are a set of pre-programmed instructions permanently stored in the machine controller that automate many of the required repetitive tasks. These canned cycles are executed or called upon by entering a certain code together with any required variable information. Their use eliminates the need for many lines of programming, reduces the programming time and simplifies the whole programming process.

As the canned cycles provided as standard with the CNC machine are for some standard motions and geometry, and thus necessitating the development of canned cycles for other motions and complex geometries. Through Parametric Programming, one can develop a canned cycle for the desired motion and geometry. With custom macro, one can develop general purpose routines for operations like thread milling, grooving, and pocket milling. In summary, one can develop one's own canned cycles.

3.3 Complex Motions

Through Parametric programming, a controller may be equipped with a capability of a different type of interpolation required for a complex motion for which the controller may be incapable as such. For example, to perform accurate taper thread milling (taper threads), the CNC controller must have the ability to form a spiral motion in XY while forming a linear motion in Z (helical motion will not suffice in this case). Unfortunately, most CNC controls do not have spiral interpolation. But, with custom macro this desired motion could be generated. Thus, custom macro also allows creating desired forms of interpolation. Moreover, variables could be used to adjust dimensions that offsets alone cannot perform.

3.4. Driving Accessory Devices

Custom macro is the most popular parametric programming language used to drive the devices such as Probes, in-process and post process gauging systems, and many other sophisticated devices. All these devices for the performance of their respective functions require a higher level of programming than that can be found in standard G code level programming. In fact, it is said that if a machine has a probe then it is very likely that machine has a controller equipped with custom macro.

3.5 Utilities

Utilities are those equipment or processes or works that are not part of the standard machine or process or work but supportive to it. Custom Macro B is used to communicate with these external devices such as printers, metrology devices, etc. Custom macro can help reduce setup time, cycle time, program transfer time, and in general, facilitate the use of these equipment. A few example applications that fit into this category include part counters, tool life managers, jaw boring for turning centres, using standard edge finders as probing devices, and facilitating the assignment of program zero. Custom Macro B could also be used to check if machine is in proper position by using "Safe Start" Macro.

4. SAMPLE PROGRAM

To appreciate what can be done with parametric programming [4], a simple illustrative program is written in Custom Macro B. This program will taper turn a shaft for a length equal to three times of its diameter with a taper of one-tenth of its diameter:

```
O0001
#101=30.0;
#102=0.0;
#103= #101/10;
#104= 100;
#105= 0.15;
G21 G90 M04 S#104;
```

```
G00 X#101 Z#102;
G91 G01 F#105 X #103 Z –[3*#101];
M05;
M30;
```

(It is advised to refer the Manual of the specific CNC machine as the program in this paper is for broad understanding and general illustration purpose only).

5. DISCUSSION AND CONCLUSION

Parametric programming of the CNC machines presents a tool in the hands of the part programmer with numerous applications and possibilities. This paper presents only a tip of the ice-berg of the realm of Parametric programming. Among several advantages, it also presents an easy, efficient and flexible method for low cost automation of the process of part programming. However, extreme care must be observed while assigning values to the variables especially the system variables. An in-depth study of the specific version of Parametric programming language along with the type and behaviour of the various associated variables is a pre-requisite for Parametric programming. There is a need for popularising its uses and applications through research works in the field of generation of more sophisticated canned cycles, generation of part programs especially for family of parts, generation of complex motions of the tool and drives of the CNC machines, driving accessories like edge finder and probes, and for the effective use of utilities.

REFERENCES

[1] Adithan, M. and Pabla, B.S., "*CNC Machines*", New Age International (P) Limited, Publishers, New Delhi, 2007.

[2] Lynch, M., "*Modern Machine Shop Magazine*", www.mmsonline.com

[3] CNC Concepts, Inc., *www.cncci.com*

[4] Djassemi, M., "A Parametric Programming Technique For Efficient CNC Machining Operations" *Computers and Industrial Engineering*, 35, 1-2, 1998, pp. 33-36.

[5] http://faculty.etsu.edu/hemphill/entc3710

[6] www.cnccookbook.com

[7] www.CNCSimulator.com

[8] Nikiel, G., "Computer-Aided CNC Programming for the Machining of Non-Typical Parts", *Advances in Manufacturing Science and Technology*, 31, 4, 2007, pp. 21-36.

[9] http://www.machinetoolhelp.com/Applications/macro/macro_variables.html

[10] www.cnc.info.pl/download.htm?id=26552