CHAPTER 6
CNC PART PROGRAM MODULE

6.1 PART PROGRAMMING USING CAD/CAM

A CAD/CAM system is a computer interactive graphics system equipped with software to accomplish certain tasks in design and manufacturing functions. One of the important tasks performed on a CAD/CAM system is NC part programming. In this method of part programming, portions of the procedure usually done by the part programmer are instead done by the computer. The two main tasks of a part programmer in a computer assisted programming are (a) defining the part geometry and (b) specifying the tool path. The proposed methodology is used to automate both of these tasks.

**Part geometry definition:** The fundamental objective of CAD/CAM system is to integrate the design engineering and manufacturing engineering functions. Certainly one of the important design functions is to design the individual components of the product. If a CAD/CAM system is used, a computer graphics model of each part is developed by the designer and stored in the CAD/CAM database. That model contains all of the geometric, dimensional and material specifications for the part. When the same CAD/CAM system, or a CAM system that has access to the same CAD database in which the part model resides, is used to perform NC part programming it makes little sense to recreate the
geometry of the part during the programming procedure. Instead, the programmer has the capability to retrieve the part geometry model from the storage and to use that model to construct the appropriate cutter path. The significant advantage of using CAD/CAM in this way is that it eliminates one of the time-consuming steps in computer-assisted part programming geometry definition. After the part geometry has been retrieved, the usual procedure is to label the geometric elements that will be used during part programming. These labels are the variable names (symbols) given to the lines circles and surfaces that comprise the part. Most systems have the capacity to automatically label the geometry elements of the part and to display the labels on the dias.

If the NC programmer does not have access to the data base, then the NC Programming must be defined. This is done by using similar interactive graphics techniques that the product designer would use to design the part. Points are defined in a coordinate system using the computer graphics system, lines and circles are defined from the points, surfaces are defined, and so forth, to construct a geometric model of the part. The advantage of using the interactive graphics system over conventional computer-assisted part programming is that the programmer receives immediate visual verification of the definitions being created. This tends to improve the speed and accuracy of the geometry definition process.
**Tool path generation using CAD/CAM:** The second task of the NC programmer in computer-assisted part programming is tool path specification. The first step in specifying the tool path is to select the cutting tool for the operation. Most CAD/CAM systems have tool libraries that can be called by the programmer to identify what tools are available in the tool crib. The programmer must decide which of the available tools is most appropriate for the operation under consideration and specify it for the tool path. This permits the tool diameter and other dimensions to be entered automatically for tool offset calculations. If the desired cutting tool is not available in the library, an appropriate tool can be specified by the programmer. It then becomes part of the library for future use.

The next step is tool path definition. There are differences in capabilities of the various CAD/CAM systems, which result in different approaches for generating the tool path. The most basic approach involves the use of the interactive graphics system to enter the motion commands one-by-one, similar to computer-assisted part programming. Individual statements in APT or other part programming language are entered and the CAD/CAM system provides an immediate graphic display of the action resulting from the command, thereby validating the statement.

A more-advanced approach for generating tool path commands is to use one of the automatic software modules available on the CAD/CAM system. These modules have been developed to accomplish a number of
common machining cycles for milling, drilling and turning. They are subroutines in the NC programming package that can be called and the required parameters given to execute the machining cycle.

**Computer-Automated part programming:**

In the CAD/CAM approach to NC part programming, several aspects of the procedure are automated. In the future, it should be possible to automate the complete NC part programming procedure. The proposed system is an automated system where the input is a geometric model of a part that has been defined during product design and the output is a NC part program. The system possesses sufficient logic and decision-making capability to accomplish NC part programming for the entire part without human assistance.

This can most readily be done for certain NC processes that involve well-defined, relatively medium complex part geometries. Special algorithms have been developed to process the design data and generate the NC program.

**6.2 INTRODUCTION**

The proposed CNC module is based on the process sequence made in the previous module and on the parameters of the features as derived from the feature recognition module. The CNC code contains the geometry of the tool path and the motion of the tool specified in a definite syntax as demanded by the type of controller. The overall tool path generated for the part is a union of the individual tool paths for each feature/operation.
and combined in a logical sequence. To avoid any collisions, a suitable tool retraction plane is used.

Each type of operation or feature is associated with a particular type of path and motion patterns and these are stored in parametric form and are instanced whenever necessary. Once the tool path is defined it is only a matter of issuing the motion statements to describe the motion. These are also standard per feature/operation type and are stored with the parametric tool path data. The user is also provided with a choice of selecting process parameters.

Tool changing statements are issued automatically but the user is advised to confirm that the suitable tool is placed in the Automatic Tool Changer (ATC) or turret. And other auxiliary statements (coolant on/off, etc) complete the program. A program in C is used to generate the NC part program. And the output of this module is stored in separate text file. This output file can be opened as an input file for the CNC machine. The simulation run is made to check for any tool hits or other difficulties. If required the user may make necessary changes in the program so as to suite his requirements. And finally the component is produced.

The NC code generated in this module is suitable for the FANUC controllers. Simulation runs are carried out on the CNC machine, for the sequence of operations, for collision free tool paths and few components are also produced, to validate the proposed system. The results were found to be satisfactory.
The CNC machine tool receives the directions for operation through a CNC program. The program is generated as per the program manuscript written for the job/operations to be carried out on the CNC machine. The program is prepared by listing the coordinate values (x, y and z) of the entire tool paths as suited to machine the complete component. The coordinate values are prefixed with preparatory codes to indicate the type of movement required like point-to-point, straight, and circular, along with coordinate values for tool path generation. Also, the co-ordinates are suffixed with miscellaneous codes for initiating spontaneous machine tool functions like start/stop, spindle/coolant, program/optional stops. In addition to these coded functions the coordinate values are supplemented with feed rate figures, spindle speed codes and tool codes, for proper selection of speeds, feeds and required cutting tools, during a particular operation. All these elements in a line of information form one meaningful command for the system/machine to execute, and is called a block of information. A number of such blocks sequentially written form a program for the particular component.

**Part Program:**

A part program contains all the information for machining a component which is given as a input to the control unit. The control unit provides the control signals at the correct time and in the correct sequence to the various drive units of the machine.
The input information required is a series of blocks; one operation requires one block. Within each block there may be different types of data.

**6.4 PREPARATION OF PROGRAM**

During the preparation of program for any particular component, some amount of data processing is required in making the mathematical computations for achieving the dimensional accuracy of the component produced and in making the logical decisions like coolant ‘on’ or ‘off’, spindle ‘on’ CCW or CW etc.

While making the part program for a component, the programmer first studies the drawing and decide upon the sequence of operations, the cutting tools, the path of cutter/tool, speeds and feeds at various points, other necessary information like starting and stopping of machine etc. The information is entered in a program sheet in a particular format acceptable by the machine tool-control unit combination. In the proposed system the programming is done with the help of a computer which replaces a programmer and performs the above tasks based on some logics and instructions given to it in the form of a software program.

**6.5 CODE GENERATION OF THE PART**

After the recognition of manufacturing features/operations and their sequence from the previous module i.e. the feature recognition module, the output file of the previous module is taken as an input file for this module. After mapping the features, the algorithms in the proposed
program initiate the corresponding CNC code. Each manufacturing feature has a pre-structured CNC code. The proposed system reads the strings in the feature recognition module and their corresponding coordinate values to select an equivalent code. It is to be noted that the initial and final commands of CNC code (called the Head and tail of the program) remains same irrespective of the component to be produced (commands like G21, G40, G28, M05, M30). Hence, the proposed system generates the CNC code by taking this fact in to consideration. Thus, the body of the code is part specific and changes as per the component. The important feature of this module is that, the user can check the generated CNC code and can make any alterations if required. The edited output file can be given as an input to the CNC machine tool, the simulations runs are to be made to assure an error free program. If the simulations runs are satisfactory the component may be produced.

The main challenge in the process of CNC code generation is the process of converting the coordinate values of the component (expressed in terms of X and Y) to their equivalent terms in the CNC code. For instance, in the feature recognition module the coordinate values of the features are given in terms of X and Y. And the difference of any two consecutive X ordinate values gives the value of length and that of Y ordinates gives the value of diameter, where as in the CNC programming the X ordinate values are to be expressed in terms of Z (length) and Y ordinate values are to be expressed in terms of X (diameter). Therefore,
the system has to initially transfer all the coordinate values in to their respective lengths and diameters, suitable for the generation of CNC code. Thus each of the manufacturing feature details is to be transferred to their respective points, lengths and diameters.

The coordinate values are rearranged as per the requirements of the code. For instance, in the feature recognition module the coordinate values of the features are given in terms of X and Y where as in the CNC programming the X ordinate values are to be

Another important factor being considered in the generation of the CNC code is the concept of ‘Reference point’. The reference point for manufacturing feature module represents the right most point on the outer profile of the component on the axis line or that could be present on it in case if the component has internal features (for the TEST1 component it is (154.000000, 5.000000). In connection with CNC code this reference point is to be taken as initial point (0.000000, 0.000000) and the length and diameters are to be expressed in relation to this reference point. Hence all the coordinate values from the feature recognition module need to be expressed in relation to this reference point. And also, the value of Z (length) is to be converted into negative values as the tool moves away from this reference point.

And as per the sequence of the processes is considered the individual codes are placed in an order, to form the final code with a logical connectivity among them. In doing so the system takes in to
consideration, the initial position to which the tool is to be brought with fast traverse, for each of the processes. The other basic variables in the code are speed, feed and depth of cut which may be provided by the user or they may be selected from the data provided in the system.

6.6 CNC SAMPLE CODES:

Given below are some of the CNC codes that are developed as part of the proposed work. These are given to explain the methodology by which the codes are generated.

6.6.1 CODE FOR STEP TURNING:

The essential data required for the code generation of step turning operation, is the length of the step (C) and the reduction in the diameter of the component (D) for that step whose values are obtained from the manufacturing feature details provided in the output file of feature recognition module (Table 5.4). The value of the depth of the cut (d.o.c) may be given by the user and is used to calculate the number of times through which this operation is to be performed to complete the step. The machine capabilities are to be taken into consideration while giving the value of the depth of cut. The proposed system reads the concerned data from the output of the previous module and correlates it with process parameters before it generate the CNC code for that operation.

Let us consider the operation number 4, of the test component, as shown in Figure 5.6, which represents a step turning operation (CD).
From the manufacturing features table (Table 5.4), we find the ordinates and reference values to be as given below. The proposed system identifies these parameters as variables that will be incorporated into the CNC code and are represented as given in the brackets.

ODR1 CD

C 154.000000 29.000000 100.000000 29.000000 ((x₁, y₁) (x₂, y₂))

D 100.000000 29.000000 100.000000 35.000000 ((x₂, y₂) (x₃, y₃))

Reference value 154.000000, 5.000000 (refₓ, refᵧ)

Let the feed (F), speed (S) and depth of cut (doc) are given by the user.

For this operation ‘CD’ the required calculations are made as following.

1) Number of cuts n = (y₃ - y₁)/doc

   If doc is 2 then, in this case n= (35.000000-29.000000)/2 = 3

   (When the doc is varied we may get a fractional value of ‘n’. If the fractional value is more than the value of doc then, the mod value of n is considered for number of cuts. If the fractional value is less than the doc then the fractional value itself is considered as the doc.)

2) Starting diameter (initial value of X)= (y₃-refᵧ)*2

   In this case it is (35.000000-5.00000)*2= 60.000000

3) The length of cut = (x₃-refₓ)

   In this case it is (100.000000-154.00000)= -54.000000
4) Final diameter = \( (y_2 - \text{ref}_y)^2 \times 2 \)

In this case it is \( (29.000000 - 5.00000)^2 = 48.000000 \)

5) Initial position for the tool \( (y_3 - \text{ref}_y)^2 + 2, (x_1 - \text{ref}_x) + 1 \)

In this case it is \( (61.000000, 1.00000) \) i.e. the tool comes to this position with fast traverse. And this value is followed by the G00 code.

6) As the number of cuts is three, the initial diameter (60) is reduced to the final diameter (48) in three steps (56, 52, 48). And the same is incorporated in to the CNC code.

The same logic is used to generate the code for the other step turning operations of the component.

6.6.2 CODE FOR PROFILE TURNING:

The term profile turning relates to the generation of convex and/or concave surfaces on the component. The arc profiles in design features of the component are to be interrelated to the profile turning in manufacturing. And accordingly the CNC code for the same has to be generated. In this work all the profile turnings are performed after performing the simple plain and step turning operations, that lead to the profile turning operation. Initially the curve surfaces are assumed to be edged sections (as shown in Figure 5.15). For generating the CNC code for profile turning the following information is required, in addition to the
process parameters which are assumed to remain the same as in the previous section.

1) Initial point \((x_1, y_1)\)
2) Final point \((x_2, y_2)\)
3) Radius of the arc \(r\)
4) Type of profile (clockwise, counter clockwise)

As explained earlier in the overview of the dxf file format, the arc profile may be drawn using a simple Arc command or using a Polyline command. Thus the database could be different depending on the type of command. In either of the case the proposed system is capable of extracting the feature details and mapping the same to generate the CNC code. The code generation methods for the two cases are given below.

**Profile with Arc features.**

When the profile is drawn using an arc command the following data is generated in the dxf file. The feature extraction module of this work extracts the following details of the data.

**CENTRE POINT  RADIUS  STARTANGLE  ENDANGLE**

The initial point for the arc will be the final point of the previous feature and the final point of the arc will be initial point of the next feature. The radius of the curve is directly obtained from the data under the column “RADIUS”. The start and end angles will help in deciding that whether the profile should have the convexity or concavity, in other words
it is to see that whether the tool should move in a clockwise direction (G02) or in the counter clockwise direction (G03). Thus in this case, it is the only task of mapping the design features in to the manufacturing features and then in to the code.

**Profile with Polyline feature:**

When the arc in the profile of the component is drawn using the Polyline command the following data is obtained.

The initial point of the arc is the vertex point given before the flag value 42, in a dxf file, which indicates that the value below this flag line is a “Bulge value”. Polyline command doesn’t provide the radius and the angle details separately. The radius of the arc and the direction is obtained by converting the bulge value, as explained below.

**CONVERSION OF BULGE INTO RADIUS:**

When the POLYLINE command is used to draw the arc segment, the corresponding output in the DXF file is given, below the DXF code 42, as an integer value, called Bulge. If the line segment connecting the vertices would have length d, and the perpendicular distance from the midpoint of that segment to the arc is h, then the magnitude of the Bulge is \((2 \times h / d)\). The sign is negative if the arc from the first vertex to the second is clockwise. A semicircle thus has a bulge of 1 (or -1). If the number mode is 0 (integer), Bulge items are scaled by 216. If the number mode has been set to floating-point, then the floating-point value supplied is just \(2^*h/d\) (not scaled)[15].
the feature recognition system is to be used for the downstream applications in manufacturing, the Bulge value has no meaning. Hence its value has to be resolved into the RADIUS of the arc. However, the preceding and succeeding vertices of the bulge indicate the starting and ending points of the arc segment. The following procedure is developed to convert the bulge value of an arc segment of Polyline (dark line), in Figure 6.1 given below, into its equivalent radius. The bulge value “b” along with the sign is extracted from the DXF file.

Figure 6.1. Elements of bulge on arc segment of a polyline

The distance “d” between the vertices of the arc is calculated as following

\[ d = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2} \]  

---- (1)

The value of ‘h’ is obtained by substituting value of ‘d’ in the equation (2).

\[ \text{Bulge; } b = \frac{2h}{d} \]  

---- (2)

Further the radius of the curve (r) is obtained by substituting the values of ‘h’ and ‘d’ in the equation (3)

\[ \text{Radius of the curve } r = \frac{(h^2 + (d^2 / 4))}{2h} \]  

---- (3)
6.6.3 CODE FOR GROOVE ELEMENTS:

There may be different types of groove profiles on the outer surface of a component. Let us consider the groove element “BCD”, that is being present on the given component. In such case, the following data will be obtained from feature extraction module.

B – 112.0000000 29.0000000 112.0000000 27.0000000 ((x₁, y₁) (x₂, y₂))
C - 112.0000000 27.0000000 106.0000000 27.0000000 ((x₂, y₂) (x₃, y₃))
D –106.0000000 27.0000000 106.0000000 29.0000000 ((x₃, y₃) (x₄, y₄))

Reference value 154.000000, 5.000000 (refₓ, refᵧ)

Let the feed (F), speed (S) and depth of cut (doc) be the same as given by the user.

For this operation ‘BCD’ the required calculations are made as following.

1) Number of cuts \( n = (y₂-y₁)/\text{doc} \)

If \( \text{doc} \) is 2 then, in this case \( n = (29.000000-27.000000)/2 = 1 \)

2) Starting diameter (initial value of \( X \))= \((y₁-\text{ref}_y)\)*2

In this case it is \((29.000000-5.000000)*2= 48.000000 \)

3) The length of cut = \((x₃-x₂)\)

In this case it is \((106.000000-112.000000)= -6.000000 \)

4) Final diameter = \((y₂-\text{ref}_y)\)*2

In this case it is \((27.000000-5.000000)*2= 44.000000 \)

5) Initial position for the tool \(((y₁-\text{ref}_y)*2+1) , ((x₁-\text{ref}_x)+1) \)
In this case it is (49.000000, -43.00000) i.e. the tool comes to this position with fast traverse. And this value is followed by the G00 code.

6) As the number of cuts is one, the initial diameter (48) is reduced to the final diameter (44) in single step. And the same is incorporated in to the CNC code.
Figure 6.2 CNC part program for the TEST1 component
Start

Enter the file name

Run feature extraction module

Generate text files for output & ordered files

Do you require Mnfg features?

Yes

Do you want to Generate CNC code?

Yes

Display component dimensions

Enter the Billet Dimensions (≥ Component dimensions)

Do want to enter Machining parameters ?

No

Enter the Job and Tool materials

Yes

Enter the values of Feed, Speed & d.o.c

Mapping of Operations

Generate the part program

EXIT

Print the features and EXIT

Print Manufacturing Features and EXIT

Figure 6.3 Flow chart for the overall program
6.8 VARIOUS SCREENS GENERATED DURING SIMULATION RUN

Figure 6.4 Initial screen showing the billet

Figure 6.5 Screen showing the Facing operation
Figure 6.6 Screen showing the Step turning operation

Figure 6.7 Screen showing the grooving operation
Figure 6.8 Screen showing the final component & end of the program
6.9 RESULTS & DISCUSSION

This module is used to generate the automated part program for the rotary components. The part program for the TEST1 component is presented in the Figure 6.2 shown above. The same program is being given as an input for the simulation package and tested for the process sequence, process parameters and for tool collisions. Upon the satisfactory results from the simulation runs the program is executed in a real environment and the component was produced.

The proposed system thus, shows the integration of feature recognition and CNC code generation for the axi-symmetric rotary parts in a typical CIM environment. Several other components are presented as case studies in the appendices.