

# Automatic CNC Code Generation for Profiles and Pockets

T.R. Kannan<sup>1</sup>, T.V. Rajakumar<sup>2</sup>, T.J. Praveen<sup>2</sup>

<sup>1</sup>Assistant Professor,      <sup>2</sup>U.G.Students

Dept. of Mechanical Engineering, K.L.N. College of Engineering, Pottapalayam-630611.

<sup>1</sup>[kthoguluva@yahoo.co.in](mailto:kthoguluva@yahoo.co.in)

**Abstract:** *The present market scenario necessitates programmable flexible automated systems to cope up with the global competition. CNCs play a vital part in such systems. But for manufacturing different parts using CNCs, it requires CNC programming expertise. Though systems that can automatically generate CNC programs from the 3D models are commercially available, they cost a fortune to medium and small-scale industries. Most of these industries use low end drafting softwares like AutoCAD. If a system that can interpret data from AutoCAD and that can generate CNC programs automatically is developed, then it will be much useful. Such a system is proposed in this paper. The system is developed using 'C'. It takes the AutoCAD filename in the DXF format as the input and extracts the data of the entities to generate CNC programs (for FANUC Controller) automatically. This reduces the programming time, avoids the need for a skilled programmer and increases the production rate at lower costs. The system can generate CNC programs for different profiles and pockets. For pocketing, the system uses zigzag tool path. The developed system is able to generate CNC programs effectively for various profiles and pockets*

**Keywords:** *DXF File, FANUC Controller, CNC, Profile milling, Zigzag pocketing.*

## 1. INTRODUCTION

Globalisation has necessitated products to be manufactured with high quality at lower costs. Also, the manufacturing systems should be able to manufacture different parts at shorter lead times

due to the quick obsolescence of products. This trend necessitates programmable flexible automated systems. To achieve this, CNCs are being widely used nowadays. The number of CNC controllers available in the market has increased dramatically and requires sufficient CNC programming

expertise. Commercial systems that can automatically generate CNC programs from the 3D models are too costly and hence are not affordable by the medium and small-scale industries. Such industries normally use low end drafting softwares like AutoCAD. A system that can automatically generate CNC programs by interpreting data from AutoCAD would be much useful and cost effective. Such a system is proposed in this paper. The general structure of the system is as shown in Fig 1.

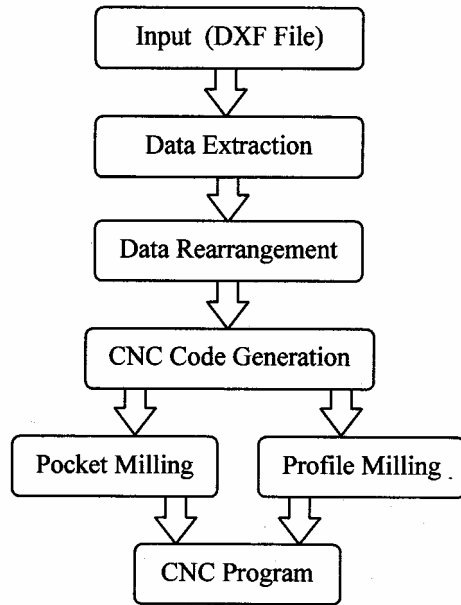


Fig 1. General Structure of the System

The system is developed using 'C'. The AutoCAD filename in the DXF format is given as the input. From the DXF file, the data of different entities in the drawing are extracted. The extracted data is rearranged in a form suitable for generating CNC programs (for FANUC Controller). Then, the cutter location data is developed based on the profile or pocket and the complete CNC program is obtained as output from the system. Apart from the DXF filename, tool diameter and depth of cut are given as inputs to the system.

## 2. DATA EXTRACTION

DXF file is a neutral file format by which the drawing data can be retrieved and used for any application program development [1]. It consists of seven sections. The entity section contains the geometric data of all the entities. A typical data in

the entities section for a line and arc shown in Fig 2 are given in Table 1 and Table 2 respectively. For lines, the starting and endpoint coordinates are extracted. Similarly for arcs, the centre point coordinates, radius, start and end angles are extracted.

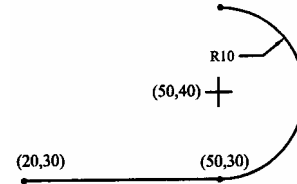


Fig 2. Typical Entities in a Drawing

Table 1. Line Data in the Entities Section of DXF File

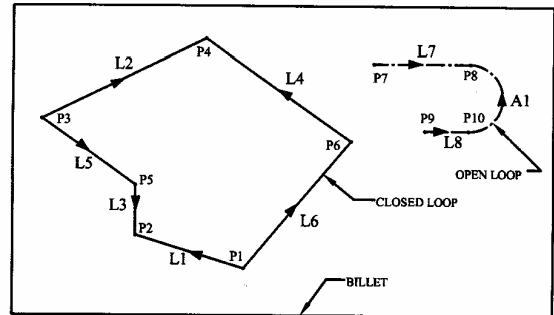
LINE	Description
ENTITIES	
0	
LINE	
5	
34	
330	
1F	
100	
AcDbEntity	
8	
0	
100	
AcDbLine	
10	X coordinate of Starting Point
50.0	X value of Starting Point
20	Y coordinate of Starting Point
30.0	Y value of Starting Point
30	Z coordinate of Starting Point
0.0	Z value of Starting Point
11	X coordinate of Ending Point
20.0	X value of Ending Point
21	Y coordinate of Ending Point
30.0	Y value of Ending Point
31	Z coordinate of Ending Point
0.0	Z value of Ending Point
0	

**Table 2. Arc Data in the Entities Section of DXF File**

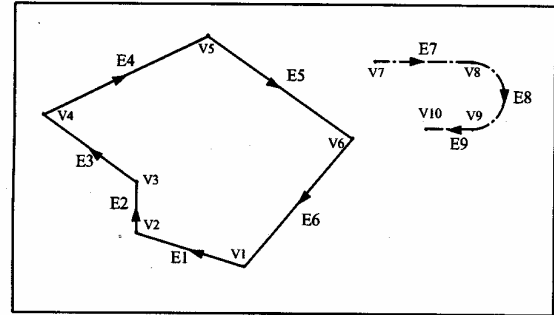
ARC	Description
ENTITIES	
0	
ARC	
5	
36	
330	
1F	
100	
AcDbEntity	
8	
0	
100	
AcDbCircle	
10	X coordinate of Centre Point
50.0	X value of Centre Point
20	Y coordinate of Centre Point
40.0	Y value of Centre Point
30	Z coordinate of Centre Point
0.0	Z value of Centre Point
40	Radius of the arc
10.0	Radius value
100	
AcDbArc	
50	Start angle
270.0	Start angle value
51	End angle
90.0	End angle value
0	

### 3. DATA REARRANGEMENT

A typical input drawing is shown in Fig 3. The drawing consists of an open loop and a closed loop. Open loops can be considered as profiles whereas closed loops may be profiles or pockets. In order to differentiate between profiles and pockets, profiles are drawn using centrelines and pockets are drawn using solid lines. In Fig 3(a), we can see that the lines are not in proper order to form either an open loop or closed loop and hence the lines have to be rearranged as shown in Fig 3(b). It involves rearranging the sequence of lines in the database as well as rearranging the start and end points. i.e. line L1 starts from P1 and ends at P2, whereas line L3 starts at P5 and ends at P2. Lines L1 and L3 have to be renamed as E1, E2 and the start and end points of E2 have to be rearranged as V2 and V3 respectively.



(a)



(b)

**Fig 3(a) Unarranged Lines and Data  
Fig 3(b) Rearranged Lines and Data**

Loops have entities that share their start and end points with other entities within the loop. In Fig 3(a), line L1 shares one of its points P2 with L3 and shares the other point P1 with L6. This information is used for determining loops and rearranging data. Loop determination procedure is done as follows. The first entity say L1 is taken and is renamed as E1. It is checked whether any other entity is having the end point (P2) of line L1 as its start or end point. In this case, line L3 is having P2 as its end point. Hence line L3 is renamed as E2 and the start and end points (P5 and P2) are renamed as V2 and V3 respectively. This procedure is continued till no other connected entity is available. By this procedure, open or closed loops are identified and rearranged in a form suitable for generating cutter location data. Entities E1, E2, E3, E4, E5 and E6 form a closed loop and the rearranged data are shown in Table 3. Then it is checked whether there are any other entities remaining. If so, an entity among the remaining entities is taken arbitrarily and the check for open or closed loop is again performed among the remaining entities. In this case, lines L7, L8 and arc A1 are remaining and they form an open loop. Thus all the entity data are rearranged.

**Table 3. New Entity Tags with Rearranged Start and End Points**

Entity	Tag	Start & End Point	New Tag	New Start & End point
Line	L1	P1, P2	E1	V1, V2
Line	L3	P5, P2	E2	V2, V3
Line	L5	P3, P5	E3	V3, V4
Line	L2	P3, P4	E4	V4, V5
Line	L4	P6, P4	E5	V5, V6
Line	L6	P1, P6	E6	V6, V1
Line	L8	P9, P10	E7	V7, V8
Line	A1	P10, P8	E8	V8, V9
Arc	L7	P7, P8	E9	V9, V10

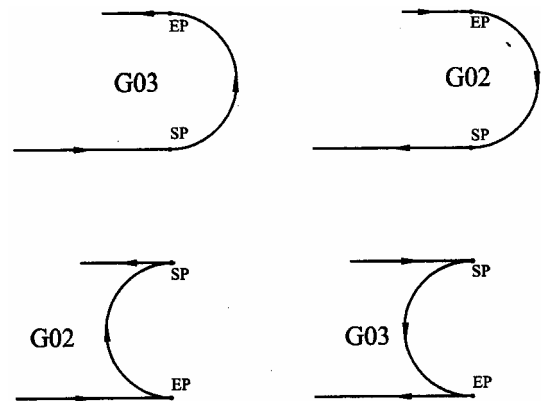
#### 4. CNC CODE GENERATION

The CNC code generation is carried out by two separate modules namely profile milling module and pocket milling module. The rearranged data is split into open loops and closed loops. Open loops are processed by profile milling module and depending on the line type, the closed loops are processed either by profile milling module or pocket milling module. First the start up of the CNC program is generated and written in a separate file. Then the appropriate body of the program is supplied by the profile and/or pocket milling modules and finally the end of the program is appended to the CNC program. The file name of the CNC program is taken as the drawing file name itself except the extension type.

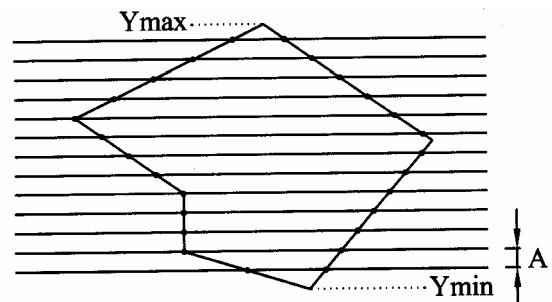
##### 4.1. Profile Milling

The rearranged data of profiles provide the information necessary to generate the body of the CNC program. Line entities are milled using G01 and arc entities are milled using G02 or G03 depending on the direction. The tool is moved rapidly using G00 to the start point of the first entity above the Z level (say Z1). After that the tool is inserted into the billet using G01 up to the

specified depth (say Z-1). Then the tool is moved to the start point of the consecutive entities till the last entity using G01 or G02 or G03. Finally the tool is moved to the end point of the last entity and then moved above the Z level (say Z1). If any other profiles are there, then again the same procedure is followed. After completing all the profiles, the control is given to the pocket milling module if any pockets are to be milled. Otherwise, the tool is returned to the home position. In profile milling, determining the direction of the arc entities is an important task. In DXF files, the arc start and end angles are always stored in the counter clockwise direction regardless of the direction of arc generation. Hence to find out the direction of arcs, a rule is used. If the connecting point of any entity with the arc is at the end point of the arc then the arc direction is clockwise (G02) else counter clockwise (G03). The start and end points, the directions and the corresponding codes used for various arc configurations are shown in Fig 4.



**Fig 4. Arc Directions and Codes for Different Configurations**



**Fig 5. Generation of Horizontal Lines for Pocket Milling**

## 4.2. Pocket Milling

The pocket milling module uses zigzag pocketing. After obtaining the rearranged data of pockets, the maximum ( $Y_{max}$ ) and minimum ( $Y_{min}$ ) Y coordinate value of the loop is determined (Fig 5). Horizontal lines are generated at a distance of  $A$  mm apart as shown in the Fig 5.  $A = Y_{min} + 0.8 * D$ , where  $D$  is the tool diameter,  $0.8$  is constant for tool overlap. A total number of  $(n-1)$  lines will be generated where  $n$  is given by  $n = (Y_{max} - Y_{min}) / (0.8 * D)$ . The intersection points of the first horizontal line with the profile entities are found out. Intersection point is determined only when the Y value of the horizontal line is lying within the range of Y values of the particular pocket entity. Similarly the intersection points of consecutive horizontal lines are found out. The intersection point between any two lines as shown in Fig 6 can be found out as follows:

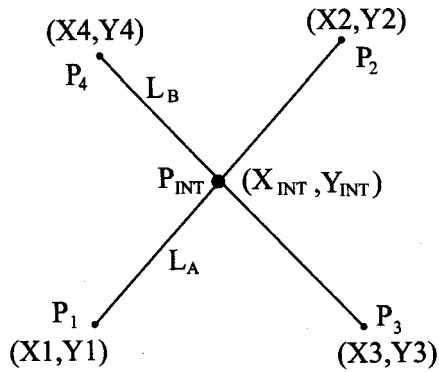


Fig 6. Intersection of two lines

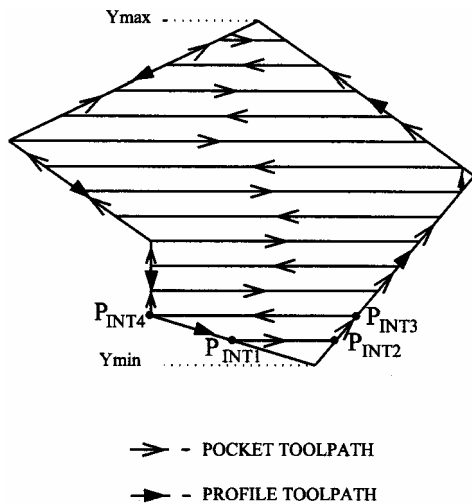


Fig 7. Tool Path Centre for Pocket Milling

The parametric equations of lines  $L_A$  and  $L_B$  are  $P_A = P_1 + uP_2$ ; where  $u$  varies from  $0$  to  $1$ ;  $P_1$  &  $P_2$  are the start and end points of line  $L_A$ ;  $P_B = P_3 + vP_4$ ; where  $v$  varies from  $0$  to  $1$ ;  $P_3$  &  $P_4$  are the start and end points of line  $L_B$ ;  $P_A$  and  $P_B$  are any point on the lines  $L_A$  and  $L_B$  for the corresponding parameter values  $u$  and  $v$  respectively;

Given the parametric equations of lines  $L_A$  and  $L_B$ , the value of the parameter  $u'$  for the line  $L_A$  at the point of intersection of lines  $L_A$  and  $L_B$  [2,3] is given by

$$u' = - \left( \frac{X1*(Y3-Y4) - Y1*(X3-X4) + (X3*Y4 - X4*Y3)}{(Y3-Y4)*(X2-X1) - (X3-X4)*(Y2-Y1)} \right)$$

The point of intersection  $P_{INT}$  is obtained by

$$P_{INT} = P_1 + u'P_2$$

For pocket milling, first the pocket profile is milled and then the tool is taken to the intersection point ( $P_{I1}$ ) having the lowest X coordinate value of the first horizontal line (Fig 7). The tool is moved to the second intersection point ( $P_{I2}$ ) of the first horizontal line. The tool is then moved to the second intersection point ( $P_{I3}$ ) of the second horizontal line and then to first intersection point ( $P_{I4}$ ) of the second horizontal line and so on.

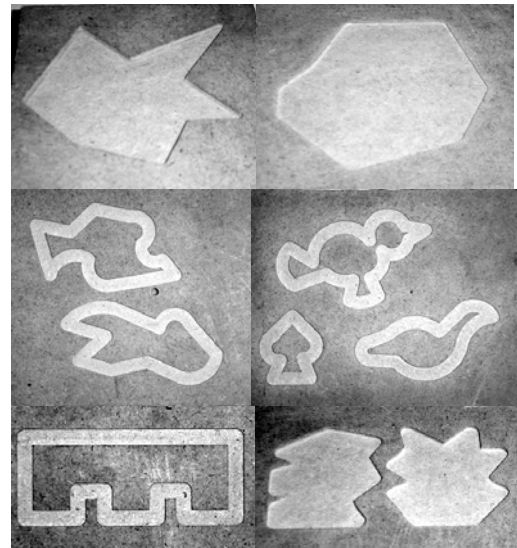


Fig 8. Sample Profiles and Pockets Milled Using the Developed System

## 5. CONCLUSION

The developed system effectively generates CNC programs for different profiles and pockets. Few sample profiles and pockets for which the system automatically generated CNC programs are milled and shown in Fig 8. This reduces the programming time, avoids the need for a skilled programmer and increases the production rate at lower costs. The system demonstrates that even with low end drafting softwares like AutoCAD, application softwares such as automatic CNC programming can be developed. With the present availability of cheap and skilled personnel, such applications can be developed in-house at an affordable cost. The system generates CNC programs for 2½D profiles and pockets only for FANUC controller. The same work can be extended to include 3D pockets and profiles by extracting data from STL formats. Also,

with slight modifications the system can be converted to generate CNC programs for any other controller.

## 6. REFERENCES

1. [www.autodesk.com](http://www.autodesk.com).
2. Rogers. D.F. and Adams. J.A., “Mathematical Elements in Computer Graphics”, McGraw Hill book Company, Newyork, 1976.
3. Ibrahim Zeid, “CAD/CAM Theory and Practice”, Tata McGraw-Hill, 1998.