MANUFACTURING FEATURE RECOGNITION OF PARTS USING DXF FILES

Nafis Ahmad* and A.F.M. Anwarul Haque
Department of IPE, BUET, Dhaka

Abstract-- This paper explains manufacturing feature recognition of a rotational component using DXF file. Feature recognition integrates two independent systems: CAD and CAPP. CAD files contain detail geometric information of a part. Different CAD software uses different types of databases to store the part information. There is no common structure of the databases and as such it is difficult, sometimes impossible to interpret or make any necessary change in the databases. In this work geometric information of a rotational part is translated into manufacturing information through a Data Interchange Format (DXF). A feature recognition algorithm is used to recognize different features of the part from its DXF file, where geometric information of the part is stored after respective DXF codes. Finally, using the data extracted from DXF file, each feature of the part is recognized.

Keywords-- CAD, Feature Recognition, CAPP,DXF

INTRODUCTION

Creating link between CAD and CAPP is one of the most difficult tasks in concurrent design and manufacturing. Without proper interface between CAD and CAPP it is impossible to generate a process plan efficiently. Feature recognition or feature extraction is the key to achieve this objective. In mechanical assembly or machining processes a feature is, usually, defined as a set of constituent faces. The geometric information related to a feature is obviously subset of object. In addition to the geometric information, some non-geometric information associated with a feature is also essential for process planning. Process planning, has very specific objectives- to identify the detailed processes required for the production of a piece of part. Product design is the input to the planning system. The engineering design representation of a part describes the part and planning system, interprets it to find the appropriate machining processes, process parameters, and so on to satisfy the design. While feature recognition from the design fills the gap between these two systems. Feature recognition is the process of extracting manufacturing features from a CAD database[1].

DESIGN TO MANUFACTURING

Engineering design is the partial realization of a designer's concept. When the idea of a part first comes to the mind of the part designer, he tries to put his ideas into an engineering drawing to represent the part completely. But the designer normally cannot directly transform a concept into a physical item. Instead the

designer conveys his idea to other people through engineering design for further process.

Different CAD packages use different types of database structures to store the information of the part in a CAD file. The design file contains all the information the part designer wants the part should have. Usually designer does not have any control over the processes when he completes the design and sends it for manufacturing. Figure 1 shows the information flow for this process. First designer's idea of a product is put into a design file (part design). After the completion of the part design, product design is completed by identifying different features (hole, slot etc) of the part. Then process plan of the part is created to manufacture the part efficiently.

In most cases parts are designed in separate CAD environment, which has no direct link with manufacturing. In this work AutoCAD is used for design purpose. But other CAD software, which has the option to convert design file into DXF format, can also be used. Here, the CAD package is used with some design guidelines to facilitate design, integration and transformation of design information into manufacturing information.

In this work only the rotational parts having external surfaces symmetrical about its axis are considered. A 3D view of a rotational part is presented in figure 2.

Design guidelines: Different CAD or geometric modeling packages store the design related information in their own databases. Structures of these databases are different from each other. As a result no common or standard structure has so far been developed that can be used by all CAD packages.
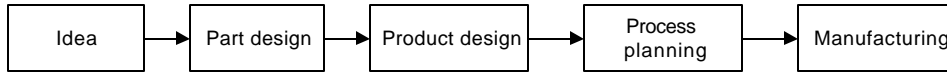
*Email: nafis@IPE.BUET.EDU

Figure 1: Information flow of a product

For this reason researchers are trying to develop a standard database structure, which will communicate with the different CAD/CAM systems.

Data Interchange Format (DXF) is a common structure, which is widely used for this purpose. But it is yet to be matured enough to transform design information into manufacturing information automatically. In this work DXF file is used with some design guidelines. These guidelines are as follows:

1. As the machined surfaces of a rotational part are symmetrical about their axis. They can be designed easily by revolving a line similar to the profile of the upper half of the part about the axis, i.e. a 2D profile of upper half of the part will represent the 3D part completely. So, 2D profile should be used to design the part. The figure 3 is the 2D profile of the 3D part shown in figure 2.

2. The 2D profile may have several numbers of lines and arcs. Each line/arc in the 2D profile represents a surface or feature of the 3D part. Lines may be perpendicular, parallel, inclined to the axis or it may be curved. 2D profile in figure 3 has 12 different features. Figure 4 (a) and 4 (b) shows four different types of lines and corresponding types of surfaces. Though it is possible to create the 2D profile in different ways in CAD environment, 'polyline' command should be used because lines, arcs can be drawn within a single entity by 'polyline' command and all feature related information can be extracted from a single 'ENTITIES' section in the DXF file.

It is already mentioned that in the polyline, line or arc connecting two consecutive vertexes represents an individual machined surface of the part [figure 3]. Each line or arc is connected at its two ends with other line or arc sequentially except the starting and ending ones. So if there are 3 lines and 1 arc, the number of surface/feature in the part is 4 and the number of vertexes is 4x1+1=5. In the example in figure 3, the number of lines and arc is 12 and therefore, there are 13 vertexes in that 2D profile.
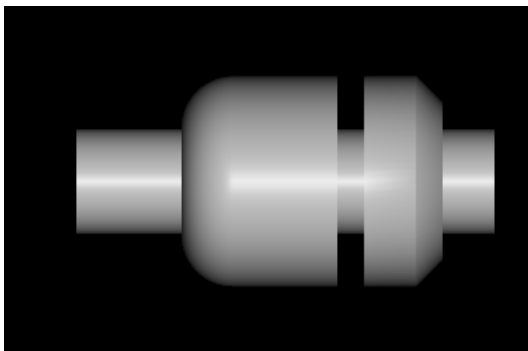
DATA EXTRACTION FROM THE CAD FILE

It is difficult for the end user to int erpret or alter the databases used by different CAD packages. Commercial CAD package developers designed their CAD system for design purpose only, not for manufacturing systems. As stated earlier, one CAD, CAM or CAPP system cannot completely understand a design file created by different software and for this reason, it is not possible for a third party to develop a CAPP system that will completely compatible with other CAD systems i.e. understand and extract all the information kept in a design file. To resolve the problem some standard formats like Initial Graphics Exchange Specification (IGES), Data Interchange Format (DXF) are being used for transferring design information from one CAD system to another CAD, CAM or CAPP system. In this research work, Data Interchange Format (DXF) is used for its rigidly defined structure [8]. It is also easy to write programs to extract data from these files.

Structure of the DXF file

There are six sections in a DXF file [7]. They are: HEADER, CLASSES, BLOCKS, TABLES, OBJECT and ENTITIES. Among them, only ENTITIES section contains geometric information regarding any object drawn in the design file. So, part feature information is extracted from the section.

Geometric information regarding each entity is stored after different DXF numbers or codes in the ENTITIES section. ENTITIES section of the DXF file of the 2D profile (figure 3) is presented in table 1. Here the DXF file is presented in a tabloid format to explain different codes and its structure. But in actual DXF file all data (DXF codes and respective values) are in a single column one after another.

DXF code '0' in the first row of the first column in table 1 indicates that a new section is started. Name of the section (ENTITIES in this case) is available in the
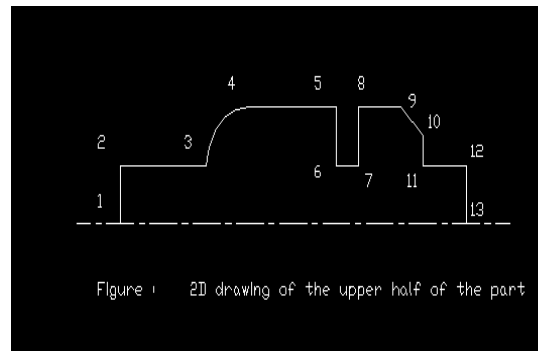

Figure 2: 3D model of the part

Figure 3: 2D profile of upper half of the part

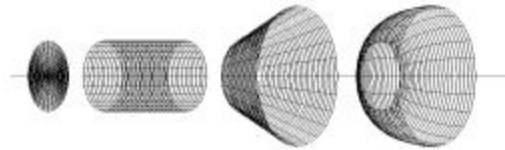*Figure 4(b) Surfaces generated from the lines*

Figure 4(a) different types of lines

fourth row of the first column after DXF code 2 (third row in the first column) as shown. Similarly name of an entity ('POLYLINE'), handle number (BC), name of the layer ('0') are after DXF code '0', '5' and '8' in the fifth, seventh and ninth row of the first column.

X, Y, and Z coordinate values of different points are available after DXF code 10, 20 and 30 respectively. In table 1 columns 2 through 14 contains the coordinates of 13 vertexes of the polyline. First column contains the starting point of the polyline. In the fourth column DXF code 42 indicates that line from third point to fourth point is an arc instead of a straight line.

The ending of the current ('ENTITIES') section is identified by 'SEQEND' in the last column of the table. Similarly 'ENDSEC' is for ending of all six section of the DXF file. Detail structure, DXF code and their description are available in AutoCAD online help.

Part data extraction: In this work, a C/C++ program is used to extract the information from the DXF file. Appendix-1 shows the flow chart of the C/C++ program. As already mentioned in a DXF file data is stored in a single column. So, the program opens the DXF file and search 'ENTITIES' section line by line until a match is found because data regarding the part geometry is available in this section. As the profile of the part is drawn by a polyline, when the program reaches the 'ENTITIES' section (table 1, column 1, row 3) it searches for 'POLYLINE' (table 1, column 1, row 6). After this, X and Y coordinates of each 'VERTEX' which represents a point in the profile are collected following DXF code 10 and 20 respectively from the entire 'ENTITIES' section. To know whether the line between two points is curved or not, the program search DXF code 42 after DXF code 10, 20, and 30 for individual vertex. DXF code 42 indicates that line between the current point and the next point is curved. The program continues until it reaches 'ENDSEQ', i.e.

the end of the entity section. When the process completes all the data extracted by the program is saved in a data file so that these data can be used for feature recognition.

FEATURE RECOGNITION

Typically, CAD and CAM are developed separately within the design and manufacturing divisions of companies, with each division seeking to exploit computers in its own way. The effect of the separate applications of CAD and CAM has increased productivity where the individual CAD and CAM application resulted in an effective solution by itself. However two separate computer systems, which cannot communicate, degrade overall effectiveness[3]. Often, product can be sophisticatedly designed with CAD, but then the designed results cannot be automatically passed to the manufacturing system. Manufacturing companies are learning that this lack of communication is a major obstacle to the achievement of the full benefits of CAD and CAM applications.

A CAD model contains all the detailed information about a part and thus provides information of all manufacturing functions. However the language of CAD is geometry based with geometric entities such as 'line', 'arc', 'circle' and so on to represent the finished part. Down-stream in CAM, features such as 'face', 'taper', 'groove', 'chamfer' and associated attributes are common language[3]. Unfortunately, computers are not intelligent enough to recognize the CAM language from the CAD language. Hence, one of the major challenges is to translate the CAD language to a CAM language and this requires a part feature recognition system (figure 5).

Part data are extracted from the DXF file consist of some coordinates of different points and have very little

Table 1: 'ENTITIES' section of the DXF file

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	SECTION	VERTEX	VERTEX	VERTEX	VERTEX	VERTEX	VERTEX	VERTEX	VERTEX	VERTEX	VERTEX	VERTEX	VERTEX	VERTEX	SEQEND
3	2	5	5	5	5	5	5	5	5	5	5	5	5	5	5
4	ENTITIES	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5
5	0	8	8	8	8	8	8	8	8	8	8	8	8	8	8
6	POLYLINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	5	10	10	10	10	10	10	10	10	10	10	10	10	10	0
8	BC	0	0	2	3	5	5	5.5	5.5	6.5	7	7	8	8	ENDSEC
9	8	20	20	20	20	20	20	20	20	20	20	20	20	20	0
10	0	0	1	1	2	2	1	1	2	2	1.5	1	1	0	EOF
11	66	30	30	30	30	30	30	30	30	30	30	30	30	30	
12	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	10			42			X-coordinate after DXF code 10								
14	0			-0.414			Y-coordinate after DXF code 20								
15	20						Z-coordinat after DXF code 30								
16	0						Value after DXF code 42 indicates the								
17	30						next line is curved								
18	0														

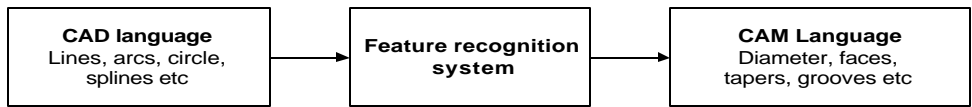| CAD language Lines, arcs, circle, splines etc	→	Feature recognition system	→	CAM Language Diameter, faces, tapers, grooves etc

Figure 5: Feature recognition system

significance for manufacturing until different features and their attributes of the part are identified. Each line straight or curved (i.e. each segment of the polyline) represents a feature on the part. To identify any feature and its attributes it is necessary to get the information of those two points which are at the two ends of the corresponding line. Length, direction, diameters etc. are determined by examining the coordinates of these two points.

Table 2 presents the logics used to determine different types of lines and their orientation[4]. A code, D-type is assigned for each type of line as shown in table 2. Here (x1, y1) and (x2, y2) are two points that represent two ends of a line. As shown in the first row of table 2, if the value of X-coordinate of the first point is less than that of the second point and Y-coordinates of these two points are equal then the line connecting the two points is horizontal from left to right. So, the surface corresponding to that line is horizontal and parallel to the part axis. D-type of this line is assigned as 1. Similarly information regarding all other surfaces are determined and coded as shown in table 2. Curved surfaces are exceptional. DXF code 42 must be present after Z -coordinate of the first of the curve.

Other attributes of each feature such as length, radius etc. are calculated from two points of each line corresponding to different surfaces. Here starting point of the polyline is on the centerline (figure 2) and the difference between Y coordinate of each point and Y-coordinate of centerline is equal to the radius of the surface at that point. When all the feature attributes are known, appropriate machining techniques like facing, turning, chamfering etc can be selected.

Table 2: Different logics used to determine line property

Logic	Direction	Line	D-type
if y1=y2 and x1<x2	Horizonta	→	1
if y1=y2 and x1>x2	Horizontal	←	2
if y1<y2 and x1=x2	Vertical	↓	4
if y1>y2 and x1=x2	Vertical	↑	8
if y1<y2 and x1<x2	Inclined	↗	16
if y1>y2 and x1<x2	Inclined	↘	32
if y1>y2 and x1>x2	Inclined	↙	64
if y1<y2 and x1>x2	Inclined	↖	128
DXF code 42 exists	Arc/Curved	↗	256

Table 3: Part database

VERTEX				Feature					
NO.	X	Y	R	No.	D type	Length	Radius I	Radius 0	Surface Type
1	0	0	0	1	4	1	0	1	Vertical U
2	0	1	0	2	1	2	1	1	Horizontal R
3	2	1	0	3	256	1.7	1	2	Curved
4	3	2	1	4	1	2	2	2	Horizontal R
5	5	2	0	5	8	1	2	1	Vertical D
6	5	1	0	6	1	0.5	1	1	Horizontal
7	5.5	1	0	7	4	1	1	2	Vertical up
8	5.5	2	0	8	1	1	2	2	Horizontal R
9	6.5	2	0	9	32	0.71	2	1.5	Inclined
10	7	1.5	0	10	8	0.5	1.5	1	Vertical D
11	7	1	0	11	1	1	1	1	Horizontal R
12	8	1	0	12	8	1	1	0	Vertical D
13	8	0							

CONCLUSION

In concurrent manufacturing systems, CAD and CAM plays the most important part for design and manufacturing. The complete benefit of the modern CAD/CAM systems are yet to be experienced due to the lack of integration between the two independent systems. This work presents an integration technique between them using DXF file. This work can also be improved for two-way interaction between product design and process planning.

REFERENCES

1. Gindy, N. N. Z.,"A hierarchical structure for form features", International Journal of Production Research, Vol. 27, No. 12, pp. 2089-2103
2. Jaekoo, J., et. Al., "Efficient feature-based process planning for sculpture pocket machining", Computers and Industrial Engineering.
3. Li, R. K and Adiga, S.,"Part feature recognition system- A vital link in the integration of CAD and CAM", Proceedings of the IXth ICPR, pp-198-205
4. Murphy, F. and Lewis, H. , 1995, " AtoM -Linking CAD to CAPP", Twelfth conference of Iris Manufacturing Committee, ICM12, pp. 55-62
5. S. Linardakis and A.R. Mileham, 1995, " A Methodology for extracting manufacturing features for prismatic components from CAD DXF files", Proceedings of the twelfth conference of the Irish Manufacturing Committee. Pp.71-77
6. Ahmad, N., "A dynamic model of Computer Aided Process Planning for rotational components", M.Engg. thesis,
7. Online help AutoCAD 2000, Autodesk Inc.
8. Hill, A.E. and Pilkington, R.D.,"A Completed AutoCAD Databook", Prentice Hall Inc (UK) ltd 1990, pp 246-250

Appendix-1: Flowchart for feature extraction

```
                        ┌─────────┐
                        │  Start  │
                        └─────────┘
                             │
                        ┌─────────┐        ┌─────────┐
                        │  i=1    │        │ i=i+1   │
                        └─────────┘        └─────────┘
                             │                  ▲ No
                             │◄─────────────────┘
                             ▼        No              No
              ┌────────────────────┐      ┌──────────────┐   Yes   ┌──────┐
              │ Line_i='ENTITIES'? │─────►│ End of file? │────────►│ END  │
              └────────────────────┘      └──────────────┘         └──────┘
                             │ Yes
                        ┌─────────┐
                        │  i=i+1  │
                        └─────────┘
                             ▼        No              No
              ┌────────────────────┐      ┌──────────────┐   Yes   ┌──────┐
              │ Line_i='POLYLINE'? │─────►│ End of file? │────────►│ END  │
              └────────────────────┘      └──────────────┘         └──────┘
                             │ Yes
                        ┌─────────┐
                        │  i=i+1  │◄──────┐
                        └─────────┘       │
                             ▼        No  │           No
              ┌────────────────────┐      ┌──────────────┐   Yes   ┌──────┐
              │ Line_i='VERTEX'?   │─────►│ End of file? │────────►│ END  │
              └────────────────────┘      └──────────────┘         └──────┘
                             │ Yes
                        ┌─────────┐
                        │  i=i+1  │◄──────┐
                        └─────────┘       │
                             ▼   Yes  ┌─────────┐       ┌──────────┐
              ┌────────────┐──────────►│ i=i+1   │══════►│ X=Line_i │
              │ Line_i='10'?│          └─────────┘       └──────────┘
              └────────────┘
                             │ No
                        ┌─────────┐
                        │  i=i+1  │◄──────┐
                        └─────────┘       │
                             ▼   Yes  ┌─────────┐       ┌──────────┐
              ┌────────────┐──────────►│ i=i+1   │══════►│ Y=Line_i │
              │ Line_i='20'?│          └─────────┘       └──────────┘
              └────────────┘
                             │ No
                        ┌─────────┐
                        │  i=i+1  │◄──────┐
                        └─────────┘       │
                             ▼   Yes  ┌─────────┐       ┌──────────┐
              ┌────────────┐──────────►│ i=i+1   │══════►│ R=Line_i │
              │ Line_i='42'?│          └─────────┘       └──────────┘
              └────────────┘
                             │ No
                        ┌─────────┐
                        │  i=i+1  │
                        └─────────┘
                             ▼        No
              ┌────────────────────┐
              │ Line_i='SEQEND'?   │──────────►
              └────────────────────┘
                             │ Yes
                        ┌─────────┐
                        │  END    │
                        └─────────┘
```