Automatic Recognition of Turning Features Using 2-D Drawing Files*

Saad M. A. SULIMAN** and Khalil A. AWAN***

A CAD/CAM link is developed for rotational components. The link consists of three modules, namely, structured modeling, feature recognition, and feature sequencing for machining. An algorithmic approach is used to generate a structured component model. A feature recognition knowledge base is developed for identifying turning surfaces on the structured component model without any human intervention. The automatic recognition process is achieved by matching entities of the component model with predefined entities to identify the types of turning features required to produce the component. The recognized features are sequenced in an operational machining sequence that produces best results.

Key Words: CAD/CAM, Feature Recognition, Feature Sequencing, Turning

1. Introduction

CAD/CAM implies an interrelated process continuum where computer is integrated in the design and manufacture of products. The latest trend in the generation of a part program using CAD/CAM is based on production of these programs automatically or semi-automatically. In the semi-automatic program generation some intervention and information are required from the user to generate an efficient program.

In CAD/CAM systems the geometric information related to an object is extracted from the solid modeling system using techniques such as design by features or automatic feature extraction/recognition^{(1),(2)}. The technological information can be obtained either automatically by the use of rule based knowledge system or semi-automatically where the user will provide the technological information related to machining of an object like operation sequencing, cutting parameters, etc.^{(3),(4)}.

A number of CAD/CAM and CAPP systems have been developed for rotational components^{(4)–(10)}. Some of these are technologically oriented and provide a generous margin for human intervention to define geometry of the component and prepare the process plan^{(4),(7)}. Other systems fill the gap between CAD and GT applications by providing automatic recognition of machined surfaces^{(1),(9)} and automatic generation of NC data⁽⁵⁾. Continuous research efforts are made to establish optimal solutions for set-up, sequencing of features, and parameters of the process⁽⁶⁾.

The aim of most of the research during the last two decades is to generate part programs automatically, i.e. without the intervention of the human operator, few systems fulfill the requirements up to date. In this work a modular CAD/CAM link is developed for rotational components. The link consists of three modules, namely, structured modeling, feature recognition, and feature sequencing for machining.

2. Structured Modeling

Using orthographic projections, a single elevation view, with internal features shown as intermittent lines, can be used to fully describe the features of any axis-symmetrical rotational component. In this work AutoCAD, the widely known CADD system, is used to

* Received 9th February, 2000

** Department of Mechanical Engineering, University of Bahrain, Box 32038, Isa Town, Bahrain. E-mail: ssuliman@eng.uob.bh

*** University of Technology, Mechanical Engineering Department, The Papua New Guinea

produce the 2-D drawing file of the rotational component. Example components are shown in Figs. 1(a), 2(a), and 3(a). The AutoCAD drawing file is then converted into DXF file for extraction of entities. Prior to the conversion of the AutoCAD drawing file, the component drawing must be oriented for left chucking.

2.1 Extraction of entities

The geometric information of each entity in the DXF file is extracted and written into an entity link list using a specially developed C-program (DXF.CPP). Entity information extracted includes entity type, style, starting point, end point, and center point of an arc entity.

2.2 Generation of profile entities

To generate the structured model that will be used for the automatic recognition of the turning features, a three stage algorithmic approach is developed. In the first stage, all entities below the component centerline, i.e. the component rotational axis, are removed from the entity link list. This is done by deletion of the entities lying below the centerline, and chopping the lower parts of those crossing the centerline (Figs. 1(b), 2(b), and 3(b)).

In the second stage of the profile generation, each of the vertical lines of the upper half is processed to chop the segment(s) that is not required for the structured profile (Figs. 1(c), 2(c), and 3(c)). The

following vertical line-chopping algorithm achieves this chopping process.

Vertical line chopping algorithm: For each vertical line (continuous or intermittent), do the following:

- Identify the entities that have one of their end points lies on the vertical line under consideration, i.e. entities having one of their end points of the same x -coordinate as that of the vertical line and of y -coordinate lying between the y -coordinates of the line end points. These points will be referred to as T-intersections in the following paragraphs.

- Number sequentially these T-intersections starting with the one nearest to the component centerline.

- If the total number of the T-intersections is odd, then the vertical line is processed by chopping the segment(s) lying between the T-intersection points alternatively upward starting with the segment connecting the first and second T-intersection points.

- If the total number of the T-intersections is even, the vertical line is processed by chopping the segment(s) lying between the T-intersection points alternatively upward starting with the segment connecting the lower end of the line (the end on or nearest to centerline) and the first T-intersection point.

- To determine the line type of each segment of the vertical line, check the types of the two entities having T-intersections with the segment. If the two entities are of the same type, then use that entity type

(a) An example of a simple solid component

(b) Upper half of a simple solid component

(c) Structured model of a simple solid component

Fig. 1 Simple solid component

(a) Example of an average component

(b) Upper half of an average component

(c) Structured model of an average component

Fig. 2 An average component

(a) An example of a complex component

(b) Upper half of a complex component

(c) Structured model of a complex component

Fig. 3 A complex component

for the segment. Otherwise, use intermittent line except for the right-most and left-most vertical lines where continuous line type is always used.

To illustrate the above algorithm three vertical lines of Fig. 2(b) are considered. Vertical line I has two T-intersections. Since the number of intersections is even, then the segment connecting the lower end of line I (i.e. the end point on the center line) and the first T-intersection point is chopped as shown in Fig. 2(c). Similar treatment is performed on vertical line II that has four T-intersections. Two segments are chopped; the first one is between the lower end of the line and the first T-intersection point. The second chopped segment is between the second and third T-intersection points. Two vertical line segments are resulting from the chopping process. The lower segment is of intermittent line type because the types of

the entities connected to the ends of the segment are both of intermittent type. Using similar analysis, the entity type of the upper segment is determined. Since three T-intersections are shown in Fig. 2(b) for vertical line III, the line segment chopped is between the first and second T-intersection points. Two segments are left, an intermittent segment between the lower end of the line (the end on the centerline) and the first T-intersection point, and a continuous segment between the second and third T-intersection points as shown in Fig. 2(c).

The third stage of the profile generation is the arrangement of the structured profile entity list.

Structured profile entity list: Having finished processing of all vertical lines, the structured profile (Figs. 1(c), 2(c), and 3(c)) is arranged in a **profile entity link list**. The resulting list includes all the continuous (visible) entities ordered according to their linking sequence from right to left followed by the intermittent (hidden) entities ordered from left to right. This is done using the following algorithm:

- **For the external profile:** Starting with the right-most entity, follow the external profile by linking its entities from right to left. As you move along the profile, use the following end point convention: the first end point of an entity is the entity starting point while the second end point is the entity ending point. Thus, the ending point of an entity is the starting point of its left neighboring entity.
- **For the internal profile:** Starting with the left-most entity, follow the internal profile by linking its entities from left to right. Using the end point convention described for the external profile, the common end point of two neighboring entities represents the ending point of the left entity and the starting point of the right entity.

If the component has a blind cavity on the left end, start with the left-most entity of the blind cavity on the right end of the component, if any, and end up with the blind cavity on the left end.

3. Feature Recognition Knowledge Base

A major input to any computer-aided process planning system is the definition of the component geometry, i.e. component geometrical model. The geometrical model can be used for:

- Human-assisted feature definition
- Automatic recognition of features.

Geometric features provide a high-level description of any component. Such description is a fundamental requirement for reasoning to determine processes, operations sequence, machine and tool selection, etc. Methods can be developed for automatic feature recognition or interactive definition of

features using geometric models. Automatic feature recognition methods interrogate and process the geometric model of a component to extract form features corresponding to some predefined geometric patterns. The geometric reasoning approach to automatic feature recognition typically includes classification of geometric entities, determination of geometric relationships between adjacent entities, and pattern matching.

In this section a feature recognition knowledge base is developed for the following external and internal turning features: cylinder, end face, shoulder face, taper, chamfer, fillet, corner, and undercut turn. Other turning features are not included at this stage. Each turning feature is represented by a **major entity** defined by the following set of properties:

1. Entity style: It is the entity drawing pattern that is either continuous or intermittent. The standard convention of the orthographic projection theory is adopted where a continuous entity represents an external (visible) feature while an intermittent entity represents an internal (invisible) feature.
2. Entity type: The type of an entity can be either a line or an arc. A line entity type represents the major entity of a cylinder, end face, shoulder face, or tapered turn feature. An arc entity type represents the major entity of a fillet or a corner.
3. Entity convexity: It is a property describing the shape of an arc entity. An arc entity with convex shape represents a corner feature while an arc entity with a concave shape represents a fillet. A line entity has no convexity; i.e. its shape is neither convex nor concave.
4. Entity change in x -coordinate (Δx): This is the difference between x -coordinates of the starting and ending points of a specific major entity. It can be used as a measure of the major entity orientation relative to the component axis. An entity with $\Delta x=0$ implies an orientation perpendicular to the component axis, e.g. a major entity representing an end face feature.
5. Entity change in y -coordinate (Δy): This is the difference between y -coordinates of the starting and ending points of a specific major entity. It can be used as a measure of the major entity orientation relative to the component axis. An entity with $\Delta y=0$ implies an orientation parallel to the component axis, e.g. a major entity representing a cylinder turn feature. Also, it can be used for classification of the turning feature into right or left feature. An entity with $\Delta y > 0$ implies a right feature, e.g. a right taper, while an entity with $\Delta y < 0$ implies a left feature, e.g. a left taper.
6. Entity peripherality: An external entity is described as a peripheral entity if it is the right-most or

left most entity, e.g. right end face or left end face. An internal entity is described as a peripheral entity if it is the right innermost or left innermost entity and if the starting or end points of the entity lies on the component axis. Any other entity that is not a peripheral entity is a non-peripheral entity.

7. Threshold relativity: This property measures the change in x -coordinates (Δx) of an entity relative to a threshold set value. If $\Delta y=0$ and if Δx is less than or equal to the threshold value, the entity is an undercut, otherwise it is a cylinder. If $\Delta y \neq 0$ and if Δx is less than or equal to the threshold value, the linear entity is a chamfer, otherwise it is a taper.

8. Entity vertical and horizontal approaches: These are the directions that the tool can take in order to approach the major entity. They are established by determining the sign of Δx and Δy . If $\Delta x < 0$, then the vertical approach is from top to bottom indicating that the tool is riding on the material. Otherwise, if $\Delta x > 0$, then the vertical approach is from bottom to top indicating that the tool is below the material, i.e. internal turning. If $\Delta x=0$, then the entity can not be approached vertically. The horizontal approach is determined by the sign of Δy where negative Δy indicates an approach from left to right, positive Δy indicates an approach from right to left, whereas zero Δy implies that the horizontal approach is not recommended. The direction of feed motion to generate a specific major entity is perpendicular to its direction of approach.

Table 1 shows the different levels of each property and the codes associated to these levels. Based on these properties, some of the common turning features are defined and coded as shown in Table 2. Thus, 2-D drawing entities are used to develop a feature recognition knowledge base that is subsequently used for the automatic recognition of the turning features.

The automatic recognition process is achieved by matching entities of the component model with the predefined major entities to identify the types of turning features required to produce the component. The structured model is used to code the major entities of the component. Consequently, the codes are used to identify the required turning features, and the relative locations of these features are established by the profile entity link list.

4. Sequencing of Turning Features

Having recognized the turning features of a specific component and their relative relationships, the next step to be carried out is to sequence these features. Sequencing of a feature implies a decision on the tool travel limits to generate the major entity of

Table 1 Levels given to the different properties and their associated codes

Digit No.	Property Code	-1	0	1
	Entity Property			
1	Style	—	Intermittent	Continuous
2	Type	—	Arc	Line
3	Convexity	Concave	None	Convex
4	Δx *	Negative	Zero	Positive
5	Δy *	Negative	Zero	Positive
6	Peripherality	—	Non-peripheral	Peripheral ** (Right-most and Left-most entities)
7	Threshold Relativity	Δx can take any value	$\Delta x \leq \text{Threshold}$	$\Delta x > \text{Threshold}$
8	Vertical Approach	Top to Bottom if $\Delta x < 0$	Not Applicable for $\Delta x = 0$	Bottom to Top if $\Delta x > 0$
9	Horizontal Approach	Left to Right if $\Delta y < 0$	Not Applicable for $\Delta y = 0$	Right to Left if $\Delta y > 0$

** This is for external entities, while right-inner-most and left-inner-most for internal entities.

* Standard coordinate system is assumed where x increases from left to right and y increases from bottom to top- chunking of the component is assumed to be on its left end.

Table 2 Properties of the major entities of external turning features and their codes

No.	Turning Feature	Properties of the major entity:									Feature (code)
		Style (code)	Type (code)	Convexity (code)	Δx (code)	Δy (code)	Peripherality (code)	Threshold Relativity (code)	V. Approach (code)	H. Approach (code)	
1	External cylinder	Continuous (1)	Line (1)	None (0)	Negative (-1)	Zero (0)	Non-periph. (0)	$\Delta x > \text{Thres.}$ (1)	T to B (-1)	NA (0)	110-1001-10
2	Right end face	Continuous (1)	Line (1)	None (0)	Zero (0)	Positive (1)	Peripheral (1)	$\Delta x > \text{Thres.}$ (1)	NA (0)	R to L (1)	110011101
3	Left end face	Continuous (1)	Line (1)	None (0)	Zero (0)	Negative (-1)	Peripheral (1)	$\Delta x > \text{Thres.}$ (1)	NA (0)	L to R (-1)	1100-1110-1
4	Right shoulder face	Continuous (1)	Line (1)	None (0)	Zero (0)	Positive (1)	Non-periph. (0)	$\Delta x > \text{Thres.}$ (1)	NA (0)	R to L (1)	110010101
5	Left shoulder face	Continuous (1)	Line (1)	None (0)	Zero (0)	Negative (-1)	Non-peri ph. (0)	$\Delta x > \text{Thres.}$ (1)	NA (0)	L to R (-1)	1100-1010-1
6	Right taper *	Continuous (1)	Line (1)	None (0)	Negative (-1)	Positive (1)	Non-periph. (0)	$\Delta x > \text{Thres.}$ (1)	T to B (-1)	R to L (1)	110-1101-11
7	Left taper *	Continuous (1)	Line (1)	None (0)	Negative (-1)	Negative (-1)	Non-peri ph. (0)	$\Delta x > \text{Thres.}$ (1)	T to B (-1)	L to R (-1)	110-1-101-1-1
8	Right chamfer	Continuous (1)	Line (1)	None (0)	Negative (-1)	Positive (1)	Non-periph. (0)	$\Delta x \leq \text{Thres.}$ (0)	T to B (-1)	R to L (1)	110-1100-11
9	Left chamfer	Continuous (1)	Line (1)	None (0)	Negative (-1)	Negative (-1)	Non-periph. (0)	$\Delta x \leq \text{Thres.}$ (0)	T to B (-1)	L to R (-1)	110-1-100-1-1
10	Right fillet	Continuous (1)	Arc (0)	Concave (-1)	Negative (-1)	Positive (1)	Non-periph. (0)	Any value (-1)	T to B (-1)	R to L (1)	10-1-110-1-11
11	Left fillet	Continuous (1)	Arc (0)	Concave (-1)	Negative (-1)	Negative (-1)	Non-periph. (0)	Any value (-1)	T to B (-1)	L to R (-1)	10-1-1-10-1-1-1
12	Right corner	Continuous (1)	Arc (0)	Convex (1)	Negative (-1)	Positive (1)	Non-periph. (0)	Any value (-1)	T to B (-1)	R to L (1)	101-110-1-11
13	Left corner	Continuous (1)	Arc (0)	Convex (1)	Negative (-1)	Negative (-1)	Non-periph. (0)	Any value (-1)	T to B (-1)	L to R (-1)	101-1-10-1-1-1
14	Under cut	Continuous (1)	Line (1)	None (0)	Negative (-1)	Zero (0)	Non-periph. (0)	$\Delta x \leq \text{Thres.}$ (0)	T to B (-1)	NA (0)	110-1000-10

* A right taper whether external or internal can be approached from right to left, whereas a left taper can be approached from left to right.

T = Top

B = Bottom

R = Right

L = Left

that feature, i.e. a decision on the amount of material removed. A C++ routine is written for sequencing turning operations using the following algorithms that are flowcharted in Fig. 4.

4.1 Rough turn sequencing algorithm

- For each taper and each fillet, introduce a cylin-

dricl turning operation to generate maximum material condition for the external features and minimum material condition for internal features, i.e. to generate the large diameters of external tapers and fillets and the small diameters of internal tapers and fillets.

- For each left feature that requires a left tool,

introduce, if required, an under-cut turning operation.

- Group features to external and internal features.
- Subgroup feature groups based on feature similarity to : face, cylinder, taper, round, fillet, and under-cut subgroups.
- First, sequence external features subgroups in the following manner : end face turning, cylindrical and associated undercut turning, right taper turning, right round and right fillet turning, undercut turning, left taper turning, left round and left fillet turning. Second, sequence internal features subgroups in the same manner.
- Sequence operations within each external feature subgroup according to the descending order of the major entity diameter. Sequence operations within each internal feature subgroup according to the

ascending order of the major entity diameter.

- Identify the tool travel limits for each operation based on its position in the sequence using the tool travel limits algorithm.

4.2 Finish turn sequencing algorithm

External features requiring finish turning are sequenced first followed by the internal features using the following algorithm :

- Sequence finishing operations requiring a right tool from right to left.
- Sequence finishing operations requiring a left tool from right to left.

4.3 Tool travel limits algorithm

Extend the major entity of the feature in both directions right and left (or up and down) :

- If the feature or its right or left extension inter-

Fig. 4 Sequencing of turning features

sects with other feature(s), then the nearest right and/or left intersection represents the tool travel limits.

- If the feature or its extension in one direction does not intersect with any other feature then the free length of the component represents the tool travel limit in that direction.

The last turning operation to be sequenced is a parting-off operation, if it is required, to complete the machining operational sequence required for production.

5. Conclusion

To further computer-aided process planning (CAPP) research, a CAD/CAM link is developed for rotational components. The link consists of three modules, namely, structured modeling, feature recognition, and feature sequencing for machining. Based on a 2-D engineering drawing, the first module uses an algorithmic approach to generate a structured component model in C-language. The model defines the profile by a set of connected entities comprising lines and arcs. A feature recognition knowledge base is developed for identifying rotational surfaces on the structured component model without any human intervention. Each turning feature is represented by a major entity defined by a set of rules that describes its type, orientation, directions, approach, size and neighborhoods. The automatic recognition process is achieved by matching entities of the component model with the predefined major entities to identify the types of turning features required to produce the component. In the final module, the recognized features are sequenced in an operational machining sequence that produces best results.

References

- (1) Wang, H.P. and Lin, C.A., Automated Generation of NC Part Programs for Turned Parts Based on 2-D Drawing Files, *International Journal of Advanced Manufacturing Technology*, Vol. 2, No. 3 (1987), pp. 23-35.
- (2) Shah, J., Sreevalsan, P. and Mathew, A., Survey of CAD Feature Based Process Planning and NC Programming Technique, *Computer Aided Engineering*, Feb.(1991), pp. 25-33.
- (3) Shyu, J.M. and Chen, Y.W., A Mini CIM System for Turning, *Ann. of CIRP*, (1987), pp. 227-280.
- (4) Hinduja, S. and Barrow, G., TECHTURN: A Technologically Oriented System for Turned Components, *Proc. Int. Conf. on Comp. Aided Prod. Eng. Edinburgh*, April (1986), pp. 255-260.
- (5) Davies, B.J., Joseph, A. and Kalta, M., An Expert System to Generate NC Data from CAD Product Models, *Journal of Materials Processing Technology*, Vol. 26 (1991), pp. 227-236.
- (6) Iran, S.A., Koo, H.Y. and Raman, S., Feature Based Operation Sequence Generation in CAPP, *Int. J. Prod. Res.*, Vol. 33, No. 1 (1995), pp. 17-39.
- (7) Korde, U.P., Bora, B.C., Stelson, K.A. and Riley, D.R., Computer Aided Process Planning for Turned Parts Using Fundamental and Heuristic Principles, *Journal of Engineering Industry, Transaction of ASME*, Vol. 114 (1992), pp. 31-40.
- (8) Leung, H.C., Hines, E.L. and Raja, V.H., Computer-Aided Process Planning in Contemporary Manufacturing Environment, *The 11th ISPE/IEE/IFAC International Conf. on CAD/CAM Robotics and Factories of the Future '95*, Pereira, Colombia, (1995), pp. 302-307.
- (9) Pradhan, P. and Zeid, I., Automatic Generation of Tool Paths for NC Turning Operations from Solid Models, *Current Advances in Mechanical Design and Production, Fifth Cairo University MDP Conf.*, Cairo, Dec.(1991), pp. 251-260.
- (10) Thakar, G., Sankar, K. and Jain, V.K., An Integrated Process Planning and NC Part Programming System for Rotational Components, *Computers in Industry*, Vol. 21, No. 3 (1993), pp. 341-357.