Proceedings of IDETC/CIE 2005
ASME 2005 International Design Engineering Technical Conferences
& Computers and Information in Engineering Conference
September 24-28, 2005, Long Beach, California USA

DETC2005-85431

RECOGNITION OF INTERACTING TURNING FEATURES FOR MILL/TURN PARTS

Shiqiao Li
Design Automation Laboratory
Department of Mechanical & Aerospace Engineering
Ira A. Fulton School of Engineering
Arizona State University, Tempe AZ 85287-6106
Shiqiao.li@asu.edu

Jami J. Shah
Design Automation Laboratory
Department of Mechanical & Aerospace Engineering
Ira A. Fulton School of Engineering
Arizona State University, Tempe AZ 85287-6106
Jami.shah@asu.edu

ABSTRACT

This paper focuses on efficient automatic recognition algorithms for turning features. As with other domains, recognition of interacting features is a difficult issue, because feature interaction removes faces and alters the topology of the existing turned features. This paper presents a method for efficiently recognizing both isolated (without interaction with other features) and interacting rotational features from geometrical CAD model of mill/turn parts. Additionally, the method recognizes Transient Turned Features (TTFs) that are defined as maximal axisymmetric material volumes from a non-turning feature that can be removed by turning. A TTS may not share any faces with the finished part. First, the rotational faces on a solid model are explored to extract isolated rotational features and some of the interacting ones. Then portions of the 3D model where no rotational faces can be used to recognize turning features are cut out and processed by a novel algorithm for finding their transient turning features.

1. INTRODUCTION

Feature recognition is the key to seamlessly integrating CAD and CAM. It can be used to find manufacturing features that are required by CAM, from a solid model that usually is the output of CAD. Much research has been conducted in this area and many feature recognition methods have been devised. [17, 6] Among the issues of feature recognition, recognizing interacting features is the most critical. Once features interact, their stereotypical geometrical/topological data are altered. Vandenbrande and Requicha [20], Gao and Shah [5], Marefat and Kashyap [13], and Regli et al. [17] proposed hint-based approaches to solve problems generated by feature interaction. Among them, Gao and Shah [5] proposed a method based on universal hints, rather than hints defined for each feature. It defined a hint as a feature's topological and/or geometric entity left in the nominal geometry of a part after an interaction takes place. These remnant entities may contain enough information to rebuild the whole shape of interacting features. Insufficient remnant entities may result in misinterpretation of features.

Mill/turns are machine tools that can do milling and turning. Usually many feature interactions between turning and milling features occur on mill/turn parts. Since on a mill/turn machine tool, turning is more efficient to remove material than milling [21], turning is preferred in the stage of rough machining. To create an efficient process plan, as many as possible turning features should be identified and removed by turning. Feature interaction, however, may result in insufficient hints (rotational faces) left in the model of the part. As Figure 1 shows, we have a cylindrical stock (Figure 1.a) to machine into the part in Figure 1.b. To create efficient process plans it is reasonable to extract two forms of feature: turning feature (the feature volume showed in Figure 1.c) and milling features (the four exploded feature volumes in Figure 1.d). Conventional methods, however, cannot recognize the turning feature that has a high level of preference over milling [19]. So far, little research has been done on feature interactions between turning and milling feature of this category of part.

The major objective of this paper is to develop an efficient recognition method that can extract not only isolated turning features (features not interacting with other features) and interacting turning features, but also turning features that enclose a milling volume (such as the volume in Figure 1.c). We label such feature as Transient Turning Feature (TTF). To reach this goal, two sets of novel algorithms are put forward. One recognizes the first two types of above features, while the other supplies an efficient approximate solution to finding the TTFs on a part. A special case of turning feature, facing feature (planar turning feature), is also taken into consideration in this paper.

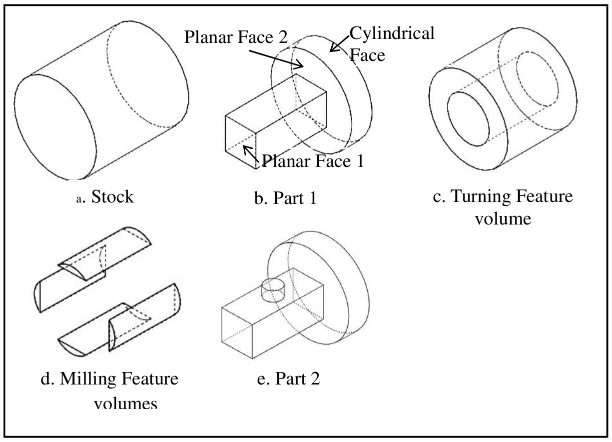


Figure 1 Misrecognition of feature

This paper is organized as follows. In section 2 an overview of related work is presented. This is followed by section 3 that gives an overview of the feature recognition method. Section 4 describes algorithms for recognizing features when at least a portion of the rotational faces of turning features remain, section 5 presents an algorithm for finding the transient turning features of non-turnable portions of the part respectively. Section 6 aggregates profile segments into complex turning features. The next section demonstrates the preliminary implementation and a case study of the method. The discussion is concluded in section 8.

2. RELATED WORK

Much research has been done on mill/turn parts. Miska [14], Noaker [15], and Owen [16] provided surveys of mill/turns, describing their capabilities and characteristics. Levin, et al. [10], Henderson, et al. [7], Li et al. [11], Madurai, et al. [12], and Dutta, et al. [3] presented methods of process planning and feature extraction for mill/turn parts. But very few of them dealt with interacting turning features and milling features. The methods of recognizing turning features in these papers are primarily face-based, which means that finding turning features to some extent is equivalent to finding rotational faces. These face-based approaches are accurate and efficient only when no feature interaction occurs.

To our knowledge, the first attempt to recognize interacting turning and milling features was made by Tseng and Joshi [19]. Yip-Hoi and Dutta [21] developed an algorithm for finding the Maximum Turnable State (MTS) of a part. This algorithm for finding MTS of a part was used by the author in [22] as a comprehensive turning feature recognizer. In this section, we primarily review the research conducted by [19] and [21] that are most closely related to ours. An extensive overview of earlier rotational feature recognizers can be found in [19].

The feature recognition methods supplied by present commercial CAD/CAM software dealing with turned or mill/turn parts lack comprehensive consideration of feature interaction. Most of them can be labeled as face-based methods. A few take feature interaction into account, but can only handle of limited cases. For example, FeatureCAM [23] can recognize turning features interacting only with depression features.

2.1 Recognition of rotational features interacting with other features

Tseng and Joshi [19] provided a comprehensive overview of rotational feature recognizers and concluded that none of them can deal with feature interaction. Then they put forward a new algorithm called machining volume generation method to recognize interacting features in mill/turn parts. In this algorithm, the volumes of rotational features are generated by sweeping boundary faces along the direction of axis. Different types of machining features can be recognized by generating different forms of machining volumes using various machining operations. The generated machining volumes are then classified using face adjacency relationships of the bounding faces. For the *Part 1* in Figure 1.b, this algorithm is able to recognize feature volumes shown in Figure 1.c and 1.d. But it has the following shortcomings: (1) all outer loops of the boundary faces of the involved mill/turn part must consist of either linear edges or circular edges. This restriction seriously narrows the application range of this algorithm. The algorithm identifies *Planar Face 1* and *2* in Figure 1.b as boundary faces. If the outer loops are not composed of linear edges or circular edges, no feature could be recognized; (2) all information between two boundary faces is ignored and the rotational feature that lies between two boundary faces can only have straight line as its revolving profile. This results in two problems. Firstly, features like the cylindrical protrusion in Figure 1.e will be ignored, because no boundary faces can be extracted from this form of feature; secondly, rotational features with curved profile, like toroidal and spherical features, could not be recognized.

2.2 Maximum turnable state of part

Yip-Hoi and Dutta [21] introduced the concept of Maximum Turnable Status (MTS) for mill/turn parts: an intermediate state from which no more material can be removed by turning without gouging faces of the final part. From the MTS, the Maximum Turnable Volumes (MTVs) of the mill/turn part can be computed. An approach was devised to find the MTS of a part. By this approach, turning feature like the one in Figure 1.c can be extracted as a feature independent of the milling features. This approach supplied an algorithm based on model slicing method that is widely used in rapid prototyping. The algorithm consists of four major steps: (1) model slicing: intersection loop(s) are generated by slicing the model with a plane; (2) the MTS profile points are found for each slice: this is equivalent to the problem of finding the extreme distances from a vertex to an set of edges; (3) determination of slice step: in order to obtain more accuracy and more efficiency, slice step must be carefully determined; (4) Point aggregation into the MTS profile: each slice of the part will contribute at least one point to the MTS profile.

This algorithm has several shortcomings: (1) it does not work well with parts that generate B-spline curves on intersection with a slicing plane; (2) the algorithm obtains all profile segments by slicing without investigating the geometry and topology of the part. For the part in Figure 1.b, the algorithm slices the part throughout the whole axial range. It is an inefficient and approximate method to recognize the cylindrical face as a turning feature. Although the algorithm adopted a method of adaptive slicing step selection, usually only recognition of conical turning features (with linear profile) can be effectively benefited. When dealing with parts composed of non-linear analytical faces, like spherical faces and toroidal faces, the algorithm would slice the face for many steps. As a result, this algorithm is computationally expensive in comparison with the above mentioned face-based methods mentioned. Furthermore, this algorithm can only obtain approximate revolving profiles composed of straight line segments for these analytical faces. High accuracy of the profiles can be acquired only at high expense of computation time; (3) the axis of the part is predefined. The algorithm cannot extract the axis of the turning feature, whereas the parameter of the axis for a mill/turn part is crucial.

In the process of building the boundary faces, Tseng and Joshi [19] encountered the problem of calculating the extreme distances from a point to a set of edges. The purpose of finding these extreme distances is to extract rotational feature volume that is identical to the MTV generated by [21]. But no further effort is undertaken. As a result, MTV can be generated only in limited cases if the cross-sections of the part are composed of straight line segments and circular arcs.

3. OVERVIEW OF THE METHOD

In this paper, turning features are classified into three categories in terms of feature interactions between turning and other features:

- turning feature interacting with depression features
- turning features interacting with protrusion features
- transient turning features that do not have enough original geometries/topologies left after feature interaction

For turning features in the first and the second categories, the method we devised takes advantage of face-based recognition method. All revolved faces on mill/turn parts are identified to extract these two categories of turning features. As to the third category of turning feature, we developed an algorithm based on surface approximation to implement feature extraction. The program can be conceptually thought to consist of six main phases as shown in Figure 2. The input to the program is the B-rep solid model of the part.

(1) Preprocessing: finding all axes of revolution of the turning features on the solid model, transforming the part to make each axis of revolution lie on the global Z-axis respectively, finding the revolving profiles of all the rotational faces belonging to each axis, and rotating all profiles about the axis of revolution along the same circumferential direction onto the global Y-Z plane. As a result, all profiles segments will be on the same side of the global Z-axis. In this phase, rules for preliminary manufacturability analysis are applied and rotational faces that cannot pass these rules are labeled as milling features;

(2) Checking and merging intersecting profile segments. The result of interaction turning feature and depression feature could be that an originally complete revolving face is separated into multiple faces. In this phase, all separated faces are rejoined together to form a single turning feature and the feature interactions on the first category of turning feature are dealt with;

(3) Checking feature interactions. This phase deals with the second category of feature interaction. Protrusion interacting with turning features cause portion(s) of the turning features to be non-turnable. These portions are eliminated in this phase;

(4) Cutting out the non-turnable portions of the part and finding the transient turning features. The third category of feature interaction is done in this phase;

(5) Integrating the profile segments obtained from phase (2) and (3) with the ones obtained from phase (4). A continuous profile of the part in question is obtained;

(6) Aggregating profile segments into complex turning features.

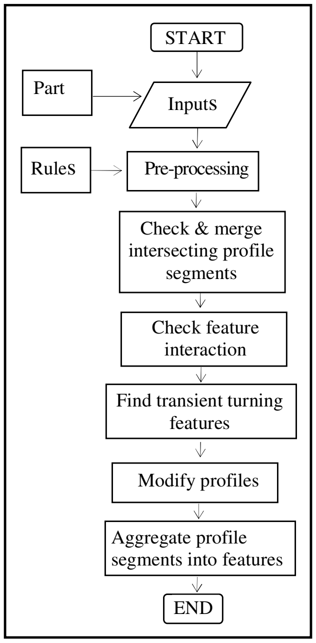


Figure 2Flowchart of the method

This paper only considers turning features in mill/turn parts. Generally, interacting milling features can be recognized by hint-based methods in [9], [17], and [20]. Tseng and Joshi [19] have put forward an effective method specifically for recognition of interacting milling features in mill/turn parts. This paper does not consider turning features like thread and knurl. These types of turning feature are usually viewed as attributes attached to rotational faces. To recognize them geometrically is beyond the interest of this research.

4. RECOGNITION OF TURNING FEATURE AND ITS INTERACTION WITH OTHER FEATURES

This section describes how to find the turning features interacting with depression or protrusion features. First of all, we need to design a data model that facilitates achieving the objectives of this paper. A Turning feature is a subset of revolved feature class. ISO STEP AP224 [8] defines revolved feature as a type of machining feature that is a sweeping of a planar shape one complete revolution about an axis. A turning feature can be completely represented by a profile with an axis. We follow this definition and establish the data model for representing turning feature.

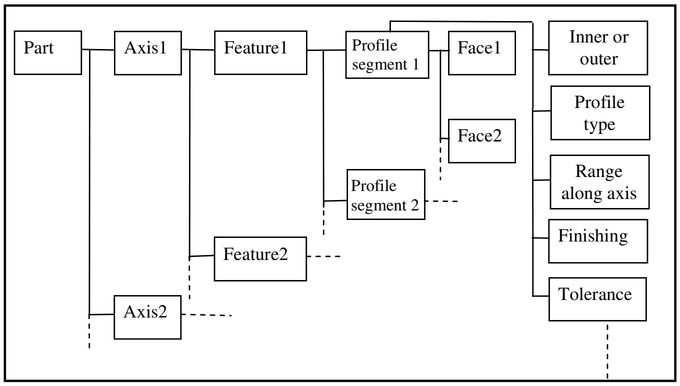


Figure 3 Hierarchical data model of turning features

As Figure 3 shows, this is a typical tree data structure. Each input solid model has a single root record called *Part* and all data will be extended from this root. For each *Part*, all possible axes of revolution are extracted and each axis is stored as an independent record. Under the record of *Axis* are Features. Each feature consists of either single or multiple profiles; turning features like a groove may have more than one profile segment. Each *Profile segment* in the data model is related to at least one *Face*. Without feature interaction, each *Profile segment* contains only one complete revolved *Face*. But once it interacts with other features, the complete *Face* could be cut into pieces. In this paper, these separated *Faces* would be joined together to form one feature. Other attributes of the profile segments are also stored in the data model. The number and contents of the attributes are both reconfigurable. This guarantees that data of turning features can be extracted and processed in a quite flexible way. This also allows high semantic level of expression of turning features. For each transient features represented by the data model, there is no data about *Face* and attributes.

4.1 Preprocessing

Before being processed in terms of feature interaction, the solid model should be preprocessed. The phase of preprocessing includes the following major tasks:

(1) Identifying all revolved faces on the part in question;
(2) Finding revolving profiles of identified revolved faces and extracting basic parameters of the faces;
(3) Checking if each revolved face can be turned.

4.1.1 Identifying revolved faces

All revolved faces are potential turning features. We classify them into five categories:

- conical (cylindrical is a special case of it)
- toroidal
- spherical
- NURBS (this type of revolved face cannot be represented in analytical format)
- planar (planar faces perpendicular to an axis of turning feature(s) could be machined by facing, a special case of turning)

4.1.2 Finding the revolving profile

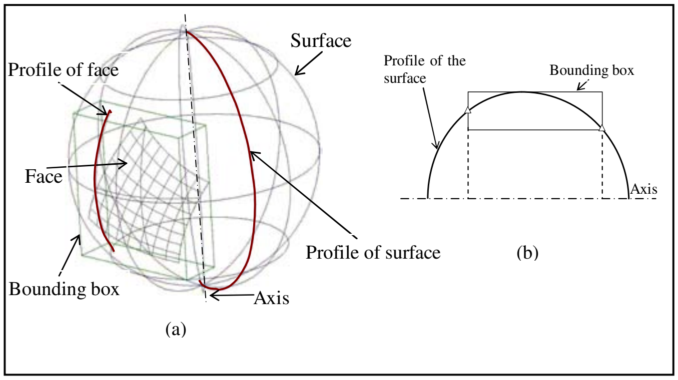


Figure4 Finding the profile of a face

Conventionally, the profile of a revolved face is obtained by slicing it and retrieving the intersection curve. Once feature interaction occurs, however, this approach will probably be unable to calculate the exact profile. As Figure 4.a shows, the profile of the face cannot be obtained by this approach. An algorithm is developed to solve this problem:

Find the profile of the surface that defines the face in B-rep;

Get the bounding box of the face and its Z-coordinate range, assuming that the axis of revolution is coincidental with the global Z-axis. (The bounding box is an axis-dependent, axis-aligned rectangular box. Since the axis of revolution of the face has been transformed and is coincidental with the global Z-axis, the top and bottom faces of the bounding box are perpendicular to the axis)

Obtain the portion of profile within Z-coordinate range of the bounding box

The portion of profile obtained with the above algorithm is the profile of the face in question. Finding the profile for a planar revolved face can be viewed as the problem of finding the extreme distances from a point to the boundaries of the planar face. This problem will be discussed in detail in sub-section 4.3.2. For outer boundaries of the planar face, maximum distance from the axis needs to be calculated. For inner boundaries, minimum distance is required. Once the minimum and maximum distances are found, the profile of the planar turning feature can be determined. The revolving profile of a planar face is a straight line segment perpendicular to the axis on the Y-Z plane.

4.1.3 Rules for preliminarily manufacturability analysis

The adoption of the face-based method arouses a problem. It regards each rotational face as a turning feature, whereas some turning features recognized by this way might be non-turnable. For example, patterned holes on a mill/turn part usually should be recognized as milling features. Since each patterned hole is composed of rotational face(s) and is identical to a turning feature geometrically and topologically in the B-rep solid model, the face-based method might recognize it as turning feature. Thus, turning features recognized by the face-based method need to be filtered to save computational time. We set up rules for preliminary manufacturability analysis to recognize milling features composed of rotational faces. The following is an example:

> *(Assuming that the maximum radius of the turning features of Axis 1 is R1 and that the maximum radius of turning features on the part is* R_{max} *)*
>
> *If* $R_{max}/R1 > 5$*, then the turning features along Axis 1 is non-turnable and should be recognized as milling features.*

The reason behind the above rule is that the fixture of the part will be difficult or even impossible if the above listed assumptions are all valid. This rule should be used together with other rules to obtain a more reasonable result of feature filtering, though we use it as an independent rule to demonstrate the concept of the method in this paper.

4.1.4 Procedures of preprocessing

Figure 5 shows the flowchart of the preprocessing. The preprocessing can be decomposed into the following steps:

(1) Finding revolved faces: once a revolved face is identified, its axis of revolution is calculated. If there is an identical axis in the data model, the face will be added to the existing axis. If no identical axis exists and the face is not planar, a new axis record will be created;

(2) Extracting parameters of the faces: tools like the bounding boxes of the faces are used to obtain parameters of the faces for preliminary manufacturing analysis;

(3) Checking turnability of the faces with the parameters and the rules;

(4) Transforming the part: After all revolved faces and the axes of revolution have been extracted, they are saved in the data model. The part is transformed to make each axis coincidental with the global Z-axis respectively;

(5) Finding and rotating the profiles of the revolved faces: For each axis record, the profiles of the faces are obtained and rotated about the axis along the same circumferential direction onto the global Y-Z plane.

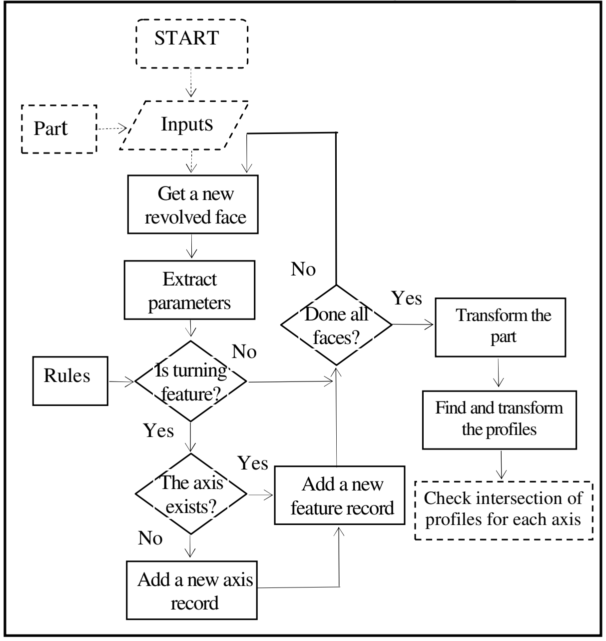


Figure 5 Flowchart of Preprocessing

The step (1), (2), and (3) need to be iteratively executed till all faces in the model have been investigated. The step (4) and (5) are also iteratively executed and the number of iteration equals to the number of axes in the data model. At the end of the preprocessing, each axis of revolution on the part has a 2-D profile on global Y-Z plane that represents all turning features sharing this axis. Figure 6.a gives an example part and Figure 6.b shows the result of the preprocessing. Totally seven revolved faces are found during the preprocessing. But *face 7* does not pass the rule in sub-section 4.1.3 and hence it is eliminated. Eventually we have only one axis extracted from this example part.

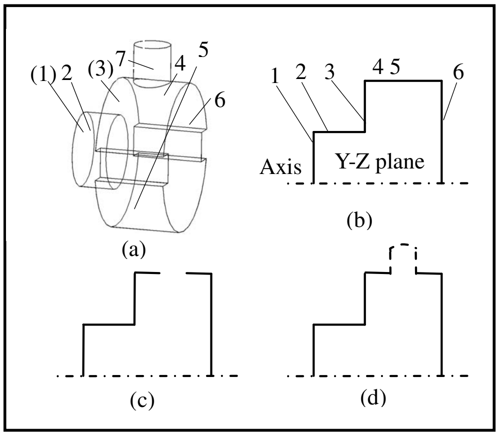


Figure 6 Example part

4.2 Merging profile segments that overlap each other

The purpose of profile merging is to solve the problem of feature interaction between rotational features and depression features. In Figure 6.a, face 4 and 5 should be recognized as one feature. After preprocessing, we obtain overlapping profile segments (4 and 5) from both faces. By merging these profile segments, we get one profile segment to represent both faces in the data structure.

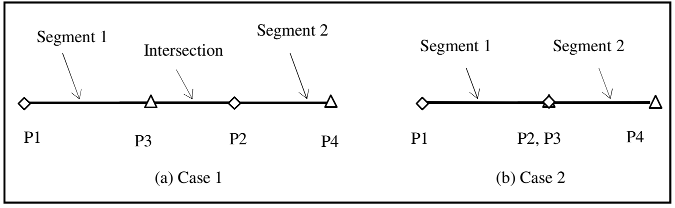


Figure 7 Intersection of profiles

We regard overlapping as a special case of intersecting. There are two possibilities of profile intersection as Figure 7 shows. The *profile segment 1* is represented by the curve between *P1* and *P2* and the *profile segment 2* is by the curve between *P3* and *P4*. These two profile segments overlap in *Case 1*. We assume that they have the same attributes, like finishing and tolerance, and merge them into one profile segment. The result profile segment is the whole curve between *P1* and *P4*. In *Case 2*, the two profile segments intersect at a point. We view them as different rotational features. All profile segments obtained by preprocessing are investigated and those that overlap as *Case 1* are merged. If one profile segment is merged into another, we need to delete the former as a feature record in the data model and add the face(s) under it to the latter. In Figure 6.b, segment 4 and 5 overlap and are merged into one profile segment in the data model.

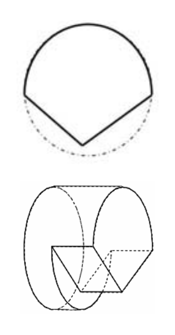
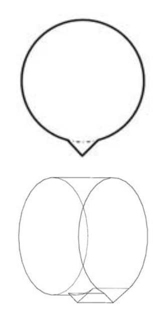
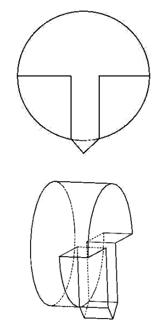
4.3 Feature interaction

The algorithms in this sub-section are developed to eliminate the influence of feature interaction between turning features and protrusion features. The protrusion features make a portion of, or even the whole turning feature non-turnable. We devise two algorithms. One is for non-planar (the first four categories of turning features classified in sub-section 4.1.1) and the other for planar turning features.

4.3.1 Interaction of cylindrical features

Table 1 shows three typical cases of feature interaction. *Case 1* is a feature interaction between depression feature(s) and a turning feature (the cylindrical face). In this case, the turning feature intersects with other feature(s) at convex edges. Feature interaction in *Case 1*can be solved by profile merging described above. *Case 2* shows a feature interaction between protrusion(s) feature and a turning feature. The turning feature intersects with other feature(s) at concave edges. These concave edges can be used to efficiently determine which portion of the turning feature can not be produced by turning. *Case 3* shows another feature interaction between protrusion feature(s) and a turning feature. In this case, the intersection edge between features does not supply sufficient information about feature interaction. Thus the feature interaction problem cannot be solved only by directly investigating the intersection edges. This principle is applicable to both non-planar and planar turning features. We develop an algorithm that can find the non-turnable portions of the turning features for *Case 2* and *3*. The basic idea beneath this algorithm is to use an accessorial geometry interesting with the protrusion features to obtain virtual intersection edges equivalent to the concave intersection edges in the *Case 2*. Since it is difficult to automatically classify a feature interaction into the three cases above, we apply this algorithm to all turning features. The algorithm consists of two steps: (1) using Boolean operations to obtain the intersection edges; (2) projecting the edges onto planes twice to get the non-turnable axial ranges. In practice, the *Case 3* does not occur often. We can directly find the intersection edges in the *Case 2* and execute the step (2) to solve the problem of feature interaction in a simplified way.

Table 1 Cases of feature interaction of cylindrical features

Case 1	**Case 2**	**Case 3**
		

4.3.1.1 Obtaining virtual intersection edges

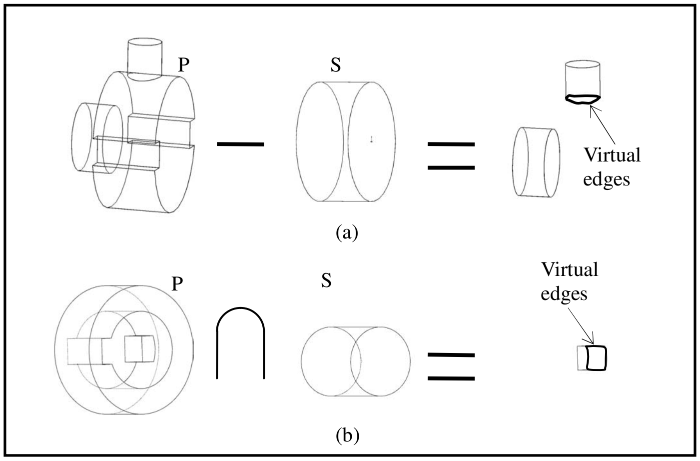


Figure 8 Boolean operations to find the intersection edges

To know which portion(s) of the turning feature are non-turnable due to interaction with protrusion features, we create a complete revolved solid (S) using the known profile segment and the axis of the turning feature. A certain Boolean operation will be applied to the NS and original part (P). As Figure 8 shows, for outer rotational feature, cutting operation, P-S, is undertaken. The cutting operation generates single or multiple new revolved faces (not including planar ones) if any protrusion interacts with the turning feature. The boundaries of these new revolved faces are the virtual intersection edges we need for further work. For inner rotational feature, intersection operation, $P \cap S$ is performed. The boundaries of the new revolved faces on the resulted solid object are virtual intersection edges.

4.3.1.2 Getting portion of turning feature affected by interaction

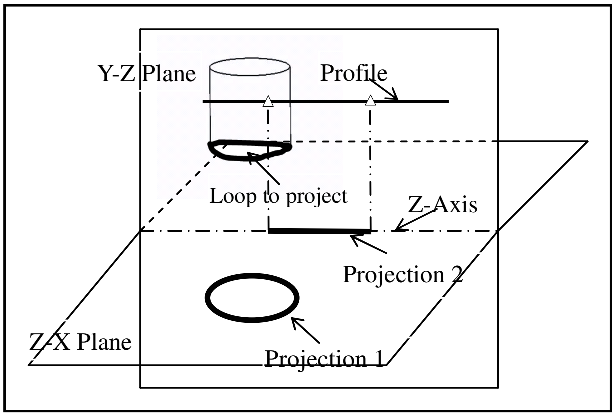


Figure 9 Non-turnable range of profile

The virtual intersection edges obtained above are projected onto one plane and the resulted projections are projected onto another plane. The planes are mutually perpendicular and both parallel to the axis of revolution of the turning feature. The result of the two projections is an edge. The Z-axial range of the resulted edge is the axial range where the portion of turning feature in question is non-turnable. In Figure 9, the virtual intersection edges of the part in Figure 8.a are used to illustrate the algorithm. These edges are projected onto *Z-X plane* along Y-axis, then onto *Y-Z plane* along X-axis. The result of the two projections is an edge (*projection 2*) on the *Z-axis*. Two straight lines perpendicular to the Z-X plane go from either ending points of the *projection 2* and intersect with the profile at the points represented by hollow triangles. The portion of the profile between the two triangles cannot be produced by turning. To obtain a certain solution, both lines must intersect the profile only at one point. This constraint still allows the algorithm to meet the requirements of most practical cases. If a portion of a profile segment is non-turnable, we split it into multiple pieces of profiles. Each piece is a feature record in the data model. In the end of this step, profile might become discontiguous. We will recover the contiguity of the profile in the step of finding the TTF of non-turnable portions of the part. Figure 6.c shows the resulted profile of the example part after checking feature interaction.

4.3.2 Feature interaction of planar turning feature

In this sub-section, we describe algorithms for solving feature interactions of planar turning features. The problem of feature interactions consists of two major tasks: (1) Obtaining the virtual intersection edges of the protrusion features; (2) Finding the extreme distances from the axis of revolution to the intersection edges.

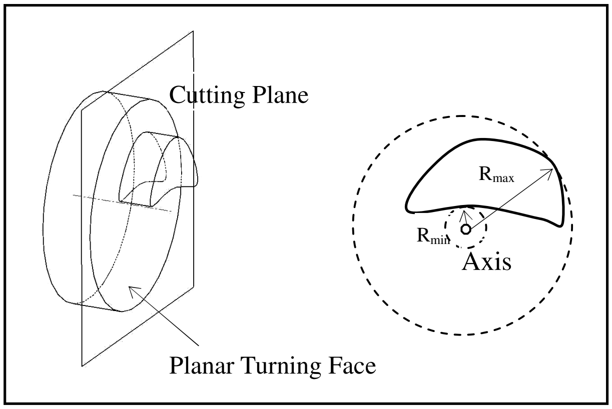


Figure 10 Cutting plane and resulted loop at planar face

As Figure 10 shows, to check feature interaction on the planar turning face on the part, we create a cutting plane perpendicular to the axis of revolution at the axial location of the planar turning face to cut the part. If the result is null, the whole planar face is turnable. If there is any intersection, the planar face cannot be turned in some radial ranges. The problem of feature interaction turns into the problem of finding the minimum and maximum distance from a point to an edge. The virtual intersection edges consist of three types of curves, namely, straight line segment, elliptical (including circular) and NURBS. Figure 11 illustrates how to find the extreme distances from a point to a straight line and an elliptical edge. On the left side of the figure, the case of straight line segment is illustrated. $P2$ is the projection of the *Axis* point onto the line segment or its extension. $P1$ and $P3$ are ending points of the straight line segment. $R1$, $R2$ and $R3$ are distances from *Axis to* $P1$, $P2$ and $P3$ respectively. If $P2$ is on the line segment instead of its extension, radial range [$R2$, *max ($R1$, $R3$)] on the planar turning face is non-turnable as result of being influenced by the straight line segment. If $P2$ is on the extension of the straight line segment, the radial range is [*min ($R1$, $R3$), max ($R1$, $R3$)] correspondingly. On the right side, the general case of circular and elliptical edge is shown. C is the center point of the arc. $P1$ *and* $P2$ are intersecting points between the edge and the straight line that determined by C and *Axis*. $P3$ *and* $P4$ are ending points of the elliptical edge. $R1$, $R2$, $R3$, and $R4$ are corresponding distances from the point of *Axis*. If $P1$ and $P2$ lie on the arc, $R1$ and $R2$ are extreme radii. For the case in the figure, the non-turnable radial range is [$R1$, max ($R2$, $R3$)]. (*: max (a, b) returns the largest number between braces and min (a, b) returns the smallest number.)

As to other types of intersection edges, we convert them into NURBS and approximate them by curve subdivision. The result of subdivision is a set of straight lines. An iterative format of the algorithm for straight line segment in Figure 11 is used to get the non-turnable radial range. Chaikin's algorithm [4] is used to do the approximation. In our experiment, usually subdivision for only three steps will produce satisfactory result.

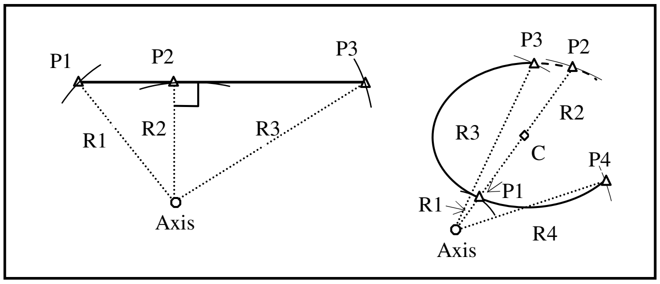


Figure 11Finding the extreme distances from a point to an analytical curve

A special issue about turned planar face is that it could be turned about different axes. We need to determine which axis to choose. Two preliminary principles are made in this paper to choose the axis: (1) obtain maximum volume removal; (2) keep the maximum radii of facings as small as possible. Principle 1 guarantees that the turned planar face is assigned to the axis about which maximum volume of material on the face would be removed. As to principle 2, the larger the maximum radius of facing is, the larger the force on the chuck is because of leverage. Pseudo-code of the algorithm to choose the right axis is as follows:

```
Apply principle 1;
If volumes of removed material for different axes are unequal
    select the axis with maximum volume;
Else
    Apply principle 2;
    If maximum radii of the same facing about different axes don't equal
        Assign the facing to the axis with smaller maximum radius.
    Else
        Assign the facing feature to any of axis, if no other principle takes effect.
```

Figure 12 shows the selection algorithm to a simple part. *Planar Face* 1 and 2 are both assigned to *Axis 2*.

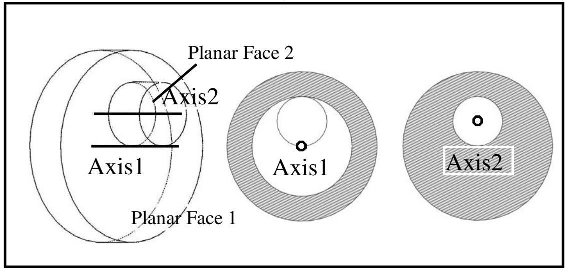


Figure 12 Selection of axis for planar rotational feature

5. FINDING THE TRANSIENT TURNING FEATURES FOR THE NON-TURNED PORTIONS ON A PART

This section describes the algorithm for finding the TTFs of a part. The concept of TTF is similar to the one of MTV in [20]. The major difference between a TTF and a MTV is: a MTV is all the amount of material removed from the stock while a TTF is the volume removed at a non-turnable portion of the part. The MTV is the difference between the stock and the MTS of a part. [21] Once the profile of the MTS of a part is obtained, the profile of the MTV/TTF can be determined.

5.1 Description of the problem and the algorithm

We describe the problem of finding the MTS of a part as shown in Figure 13: we put the part into a Cylindrical Coordinate System (CCS) with the axis of revolution being equivalent to the z-axis of the CCS. Then if we project the part along the azimuthal direction of the CCS onto a plane (θ =0. This plane is equivalent to the y-z plane of the global Cartesian coordinate system), the boundary of the 2-D projection is the generator to generate the MTV of the part. To obtain this generator, we do not necessarily project the solid body of the part, but project all the faces of it instead. Since no software tool is available, we devised an approximate method to projecting the faces onto a plane along the azimuthal direction. In this method, firstly, the faces on the part are approximated by face meshing. Secondly, we project all vertices of the meshes onto the plane and obtain a 2-D point cloud. Projection of a vertex along the azimuthal direction in CCS is easy to do. All we need to do is to rotate the vertex about the axis by a certain degree. Thirdly, we find the boundary of the point cloud. This boundary is the generator of the MTV. If we apply this method to the non-turnable portions of the part, the profiles of transient turning features can be obtained. An algorithm for this method is illustrated in detail below. The algorithm makes simplification in rotating the vertices and combines projecting vertices and finding the profile into one single step.

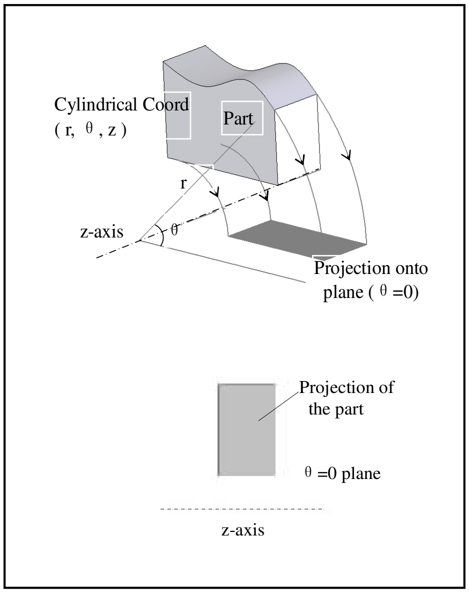


Figure 13 Finding the MTS by projecting in cylindrical coordinate system

```
Approximate all faces of the non-turnable portion or the part by a mesh;
Divide the z-axis coordinate range of the non-turnable portion into even and contiguous z-coordinate zones. For each zone, setup two variables, one for maximum R and one for minimum R;
Loop
{
    Find the distance R from a mesh vertex to the axis of revolution (assuming it is global z-axis);
    Find the z-coordinate zone where the current mesh vertex falls into;
    Compare the R value of the current mesh vertex with the value of the maximum R and the minimum R of the z-coordinate zone;
    Update the maximum R and the minimum R of the z-coordinate zone according to the result of comparison;
    Go to next mesh vertex;
}
Draw connected straight lines through either all maximum R's or all minimum R's. Two strings of line segment are obtained;
Check the angle between every two adjacent straight line segments. If the angle is smaller than a specified value, abandon the through point and connect the maximum R or minimum R point with next point;
Integrate the strings of line segment into the profile of the part. If the resulted profile is still discontiguous, connect the discontiguous vertices by extra straight line segments perpendicular to the axis.
```

The strings of straight line segment are the profiles of the transient turning feature. The dashed edges in Figure 6.d show the profile segments of transient turning features. The two dashed edges perpendicular to the axis are extra edges added by the last step of the above algorithm. If we apply this algorithm to the whole part, we will obtain the profile of the MTS of the part. This algorithm has the following characteristics by comparison with the one in [21]:

- Computationally efficient: it consists of only two major steps, creating the mesh of the faces and calculating the distances from the node on the mesh to the axis;
- Comprehensive: it can deal with any type of shape.
- Highly extendable: methods of surface approximation are being improved. That will make the algorithm faster and more accurate.

5.2 Finding non-turnable portion of a part

In this paper, the algorithm of finding the MTS for a part is only applied to non-turnable portions of the part to save time and computational expense. Whether a non-turnable portion exists on a part can be determined by checking the contiguity of the profile generated after the steps of profile merging and feature interaction check. Each profile has an axial range. A profile vertex would be regarded as starting vertex, if it (1) lies on the axis and belongs to only one profile segment; or (2) lies on the starting or ending axial range of the profile and belongs to only one profile segment. If a profile vertex belongs to only one profile segment and is not a starting vertex, it is a vertex for identifying non-turnable portion of the part. In Figure 14, the two triangular vertices are recognized as starting vertices and the four circular ones are marked. Cutting planes will be created at the position of these marked vertices. Discrete non-turnable portions will be cut out from the part.

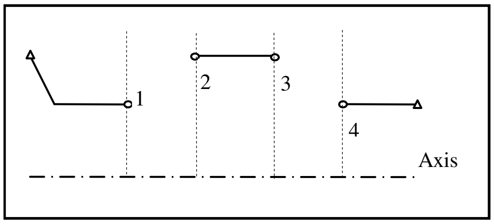


Figure 14 Cutting out non-turnable portions of the part

6. PRELIMINARY IMPLEMENTATION AND CASE STUDY

The implementation of the method in this paper follows the flowchart in Figure 2 with APIs and class function from ACIS Geometric Modeling Kernel. While calculating the transient turning features, we need to approximate the faces of non-turnable portions of the part in meshes. Many tools are available to do this. Among them, surface subdivision based on control mesh, auto FEM meshing tools for surface, and faceting tools used in computer graphics work efficiently and are strong candidates for the implementation of this paper. We currently use the faceting subroutine supplied by ACIS to generate the approximate mesh.

A case study is used to illustrate the procedures of the system in this section. The part in Figure 16.a has all types of turning features and turning feature interaction. It has some features (NURBS faces) that cannot be recognized by the methods in [19] and [21]. The difficulties of the part also include that it has turning feature belonging to multiple axes. The following are detailed procedures by which the part is processed.

(1) Identify all revolved faces on a part. As shown in Figure 16.a, 34 faces are potential turning features. These faces are classified according to their axes of revolution. The rules in section 4.1.3 works as the filter to eliminate axes. In the example, four axes are extracted. But the two axes of the blind holes (16, 17) should not be machined by turning, because these two features have small radii and would better be machined by milling according to the rule in 4.1.3. Eventually two axes are recognized;

(2) Extract revolving profile of each revolved face. All profiles are rebuilt on the global *Y-Z plane* with the axis of revolution being on the *Z-axis*. Overlapping profiles are merged into one profile. Figure 16.b shows all merged profiles of revolved faces belonging to one of the two axes (*axis* 1);

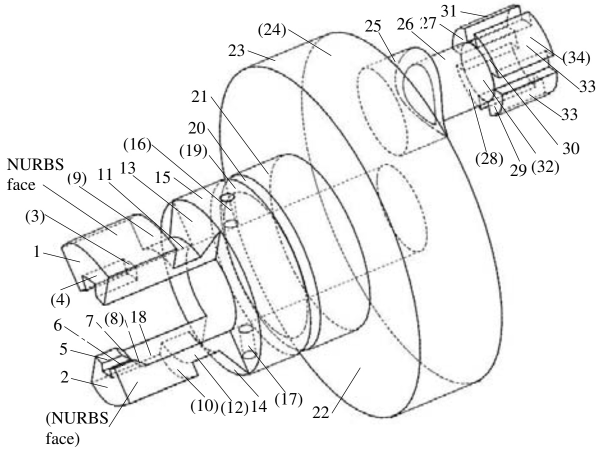
(3) Check interaction of each profile segment. Figure 16.c is the result of interaction checking. All non-turnable profile

segments are removed. Ideally, we would obtain a contiguous profile after the interaction checking, but the contiguity might be broken due to feature interaction;

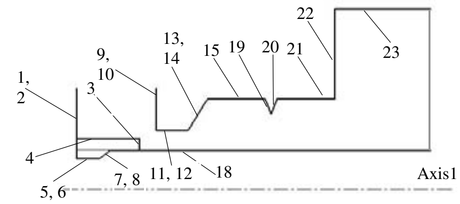
(4) Cut out the non-turnable portion. Two planar disks perpendicular to the axis are created at each discontiguous profile vertices. The disks should be large enough and their sizes can be determined by the bounding box of the part. Figure 16.d indicates the two cutting disks and Figure 16.e is the non-turnable portion of the part;

(5) Approximate all faces on the non-turnable portion by the faceting subroutines of ACIS. Each node on the meshes is projected along the circumferential direction to form a 2-D point cloud. The boundary of the 2-D point cloud is found. Figure 16.f illustrates the projection and the axial zones generated to calculate the boundary;

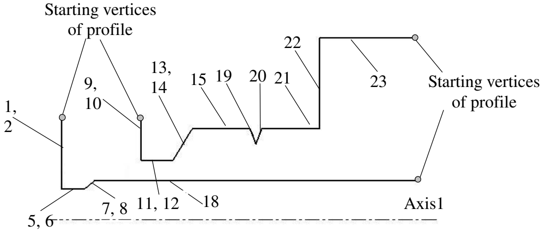
(6) Add the boundary of the non-turnable portion in the profile. A contiguous profile is obtained in Figure 16.g;



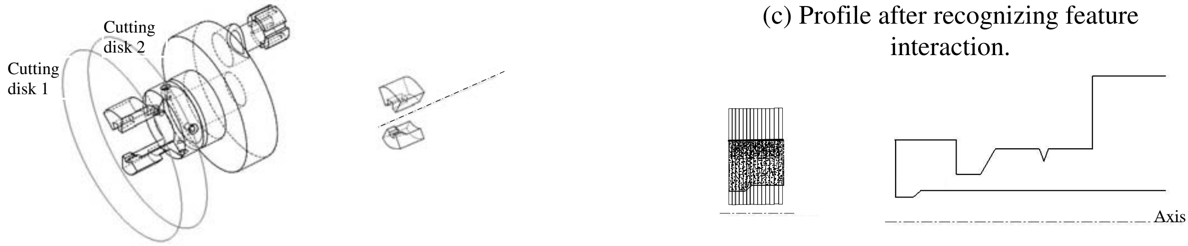
(a) All revolved faces on the part.



(b) Profile after emerging intersecting segments.



(c) Profile after recognizing feature interaction.



(d) Cutting out the non-turnable portion.

(e) Non-turnable portion.

(f) Projected vertices and axial zones.

(g) Result profile.

(h) Profiles.

(i) Profile after checking interaction.

(j) Resulted profile.

Figure 16 Case stud

Figure 16.h, 16.i, and 16.j briefly shows recognition of features for the other axis.

7. CONCLUSION

This paper presents a new method for recognizing turning features that interact with other forms of features in mill/turn parts. All turning features, including transient turning features, can be recognized no matter how they interact with other features. This is a comprehensive solution to the recognition of tuning features. The method overcomes all shortcomings of related work in section 2 and brings some new functions: (1) it uses a novel face-based algorithm to recognize turning features and an efficient algorithm to find TTFs on a part. Using different algorithms to solve these two types of feature improves computational efficiency of the method; (2) it can deal with mill/turn parts that have multiple axes of revolution; (3) it supplies a data model that can express turning features in high semantic level; (4) it considers requirements of future work, like feature customization and manufacturability analysis, at the stage of feature recognition.

The method will be developed into a customizable turning feature recognizer and integrated into the DFM system developed in the Design Automation Lab of the Arizona State University. The approach to the problem of finding the boundary of a planar point cloud in this paper would be replaced by methods, like the one in [2], which supplies more general solutions.

REFERENCES

[1]. *ACIS Geometric Modeler Version 6.0 Programming Reference*, Boulder, CO, Spatial Technology Inc.

[2]. Braquelaire, J-P, Vialard, A., "Euclidean Paths: A new representation of boundary of discrete regions", *Graphical Models and Image Processing*, 61, 16-43, 1999.

[3]. Dutta, D., Kim, Y., S., Kim, Y., Wang, E., Yip-Hoi, D., "Feature extraction and operation sequencing for machining on Mill-Turns", *Proceedings of 1997 ASME Design Technical Conferences*, Sacramento, CA, CIE-4276.

[4]. Farin, G., *Curves and surfaces for CAGD: A practical guide*, fifth edition, Morgan Kaufmann Publishers Inc., 2001.

[5]. Gao, S., Shah, J. J., "Automatic recognition of interacting machining features based on minimal condition subgraph", *Computer-Aided Design*, 1998, Vol. 30, No. 9, 727-739.

[6]. Han, J. Y., Pratt, M., Regli, W. C., "Manufacturing feature recognition from solid models: a status report", *IEEE Transaction on Robotics and Automation*, Vol. 16, No. 6, December, 2000.

[7]. Henderson, D. L., Anderson, D. C., "Feature-based design for Mill/Turn and 3-axis machining", *In Proceeding of the 1994 ASME Design Technical Conferences*, DE Vol. 69-1, Minneapolis, MN, 1994, 564-573.

[8]. *STEP AP224, ISO Standard 10303-224, Product data representation and exchange: Application protocol: Mechanical product definition for process planning using machining features.*

[9]. Joshi, S., Chang, T. C., "Graph-based heuristic for recognition of machined features from a 3D solid model", *Computer-Aided Design*, 1988, Vol. 20, 58-65.

[10]. Levin, J., Dutta, D., "Computer-aided process planning for parallel machines". *Journal of Manufacturing System*, 1992, Vol. 11, No. 2, 79-92.

[11]. Li, R-K., Kwang, T-Z., "A part-feature recognition system for rotational parts — non-turning features", *International Journal of Computer Integrated Manufacturing*, 1989, Vol. 2, 257-267.

[12]. Madurai, S., S., Lin, L., "Rule-based automatic part feature extraction and recognition from CAD data", *Computer Industrial Engineering*, 1992, Vol. 22, 49-62.

[13]. Marefat, M., Kashyap, R. L., "Geometric reasoning for recognition of 3-D object features", *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 12, No. 10, 1990.

[14]. Miska, K. H., "Driven tools turn on turning centers", *Manufacturing Engineering*, 1990, May, 63-66.

[15]. Noaker, O., "Turning machines and systems", *Manufacturing Engineering,* 1990, August, 44-48.

[16]. Owen, J., "Turning machines and systems", *Manufacturing Engineering*, 1992, August, 54-56.

[17]. Regli, W. C., Gupta, S. K., Nau, D. S., "Toward multiprocessor feature recognition", *Computer –Aided Design*, Vol. 29, No. 1, 1997.

[18]. Shah, J. J., Anderson, D., Kim, Y. S., Joshi, S., "A discourse on geometry feature recognition from CAD models", *Journal of Computing and Information Science in Engineering*, Vol. 1, 2001, March, 41-

[19]. Tseng, Y. J. and Joshi, S. B., "Recognition of interacting rotational and prismatic machining features from 3-D mill-turn parts", *International Journal of Production Research*, 1998, Vol. 36, No. 11, 3147-3165.

[20]. Vandenbrande, J. H., Requicha, A. A. G., "Spatial reasoning for the automatic recognition of machinable features in solid models", *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 15, No. 12, December 1993.

[21]. Yip-Hoi, D., Dutta, D., "Finding the maximum turnable state for mill/turn parts". *Computer-Aided Design*, 1997, Vol. 29, No. 12, 879-894.

[22]. Yip-Hoi, D., Dutta, D., Huang, Z., "A customizable machining feature extraction methodology for turned components", *Journal of Manufacturing Systems*, 2003, Vol. 22, No. 2, 82-98.

[23]. http://www.enggeo.com/