RESEARCH ARTICLE OPEN ACCESS

Manufacturing Computer Aided Process Planning For Rotational Parts. Part 1: Automatic Feature Recognition From STEP AP203 Ed2

Jaider Oussama*, Elmesbahi Abdelilah*, Rechia Ahmed*
*(Research team in Engineering, Innovation and Management of Industrial Systems, Abdelmalek Essaadi University, Faculty of Sciences and Technics of Tangier, Morocco)

ABSTRACT
In response to the urgent need to reduce the cost of manufacturing components, various CAD, CAPP, CAM systems have been designed to automate every step involved in the life cycle of a product. One of most difficult steps is to generate automatic process plans. This task is impossible unless that a link between CAD and CAPP is established. Thus, the implementation of a feature recognition module in CAPP systems is primordial. Research in feature recognition has received significant attention, however, the majority of feature recognition systems for rotational parts treat isolated features. This paper presents, a new system of recognizing both isolated and interacting features for rotational parts taking STEP AP203 Ed2 as an input to the system. The methodology works in three main phases. The first phase addresses extraction of geometric and topological information from STEP file. The second phase consists of analyzing the extracted geometric and topological data to recognize turning features. The third phase takes the recognized features as input to generate all possible combinations of interacting features. An illustrative example is presented to test and validate the method.
Keywords – CAD/CAM, CAPP, Manufacturing Feature Recognition, STEP format, turning process

I. INTRODUCTION

The main goal of any manufacturing organization is to produce high quality products at a competitive price, at the same time, the continuous changes in customer needs are to be satisfied. To suit these requirements, several software spanning various disciplines have been designed to ensure an easier, a faster and a flexible workflow [1]. In the recent years, the integration of Computer Aided Process Planning (CAPP) has received significant attention since it provides a vital link between computer Aided Design (CAD) and Computer Aided Manufacturing (CAM). CAPP selects the necessary processes, tools and generates automatic sequences of operations and instructions to manufacture the part, taking into account, surface roughness, Geometric Dimensioning and Tolerancing (GD&T), economic and technological precedence constraints. To achieve these tasks, CAPP has to extract and recognize manufacturing information such as machining features, directly from 3D solid model.

In spite of using advanced automation technology, the link between CAD and CAPP systems is still not integrated as desired [2]. On one hand, the data of the neutral files such as STEP, IGES generated by CAD systems consist of geometric and topological information, these data cannot be used for direct application to process planning since CAPP systems require part form feature information, not geometric and topological information, otherwise, CAPP systems do not understand the three dimensional geometry of the designed parts in term of their engineering meaning related to other product information, such as material properties, technological parameters, and required manufacturing precision [3]. Many research efforts have been done for automatic feature recognition for rotational parts and the majority of authors have focused on recognizing isolated features. However, feature recognition becomes more complex when features interactions occur since some surfaces of features are lost by interactions. On the other hand, Geometrical and Dimensional Tolerancing data transferred to downstream applications such as process planning are not embedded in the geometric model for the most of the current CAD systems, which are lacking of appropriate data structure to admit them. CAD models seem to include these data as seen in the drawings, nonetheless, these data are not real attributes of CAD models but simply represented as text on the drawing [4]. This is seen as a hindrance in the flow of information between CAD and CAM.

To solve the CAD and CAPP interface problems, the implementation of a feature recognition module in CAPP systems is imperative, in which manufacturing information, geometric and topological data are extracted, recognized and stored together. For the purpose, a neutral format for the representation is required for facilitating an interface between different CAD/CAPP/CAM systems. There are different neutral files available, some of them are IGES, STEP, DXF, STL files etc. The Standard for the Exchange of Product (STEP) model data, which is defined as the international standard ISO 10303-203 [5], is worldwide recognized neutral file format where it is available for almost all commercial CAD software, given the fact that it contains coordinates data that are simple to extract, includes not only geometric and topological data, but also Geometric Dimensioning and Tolerancing (GD&T) data related to the part. This gives a clue to the solution.

In this paper, the authors have focused on the development of an automatic feature recognition system for rotational parts by taking STEP AP203 Ed2 neutral file as an input of the system. The proposed system consists of three modules namely Geometric and Topological Data Extraction module, Feature Recognition module and Feature Generator module. In the first module, dimensional, topological and geometric information of the part features are extracted from STEP file using C++ programming. The extracted dimensional, topological and geometric information are reordered and stored in a database and these data are analyzed by downstream activities. The second module consists in analyzing geometric and topological data obtained from the GTDE module, such as circle centers coordinates, edges, loops, type of surfaces, and so on. A library which consists of turning-pre-defined manufacturing features is elaborated to enable the automatic recognition. The third module takes the recognized features as input to generate all possible combinations of interacting features. The elaborated algorithm permits to handle both isolated and interacting manufacturing feature with multiple possible interpretations.

II. LITERATURE REVIEW

Automatic Feature Recognition (AFR) directly from a CAD model is the first and the most difficult task in a CAPP system to achieve downstream activities such as automatic process planning. There have been many previous attempts to recognize form features for manufacturing purposes, which can be broadly classified into four major categories: Syntactic pattern recognition, Graph based, Rule based, and artificial neural network method.

Ismail & Abu Bakar (2005) [6] used syntactic pattern recognition method for feature recognition. An upper half of the 2D profile information of a part is given, which is a series of lines and arc segments that represent semantic primitives written in some description language. A set of grammar using a sequence of characters, which consists of some rules, defines a particular feature. The parser for input sentence analysis has been then used to apply a grammar to the part description (features connected to form a part). If the syntax agrees with the grammar, then the description can be classified in a corresponding class of features.

Mehalawi & Miller (2003) [7] used an attributed adjacency graph (AAG) for building a database that captures the geometric and topological similarity in order to facilitate extraction of machining features. Workpieces are represented using attributed graph-based on STEP file, in which the nodes correspond to the surfaces of the Workpiece and the links correspond to the edges of the Workpiece. The main limitation of this approach is its inability to detect features with non-planar surfaces.

Abouel Nasr & Kamrani (2006) [8] proposed a methodology for 3D prismatic parts that are modeled using constructive Solid Geometry (CSG) technique as a drawing tool. The system takes a neutral file in Initial Graphics Exchange Specification (IGES) format as input and translates the information in the file into manufacturing information. The boundary (B-rep) geometrical information of the part design is analyzed by a feature recognition program that is created specifically to extract the features from the geometrical information based on a geometric reasoning approach, by using object oriented design software. A feature recognition algorithm is used to recognize different features of the part such as step, holes, etc.

Sivakumar & Dhanalakshmi (2012) [1] developed a system that uses a simplified and generalized methodology of extracting manufacturing features from STEP AP203 file for cylindrical parts. The dimensional and geometric information of the part features and their positions are extracted from STEP file using feature extraction process. The extracted dimensional and geometric information are stored in a text file and these data are analyzed using logic rules for identifying turning features. An example logic rule to recognize a cylinder feature is used as follows:

IF STEP data contain CILYNDRICAL_SURFACE and the radius is same at both end surfaces
THEN the feature is identified as straight cylinder

This method lack of extracting topological information, and classifies features as elementary surfaces as shown in STEP file, like tapers, plans, cylinders and so on. Therefore, features formed by several elementary surfaces such as grooves are not recognized. Another drawback is that the method does not handle interacting features.

Malleswaria & Sarcar (2013) [3] developed a new feature recognition software for rotational parts which uses STEP file as input. The software analyses various strings and entities (# number) in the STEP

file and extracts dimensional, geometric and topological information including, EDGE_CURVE construction of surfaces, circle centers, radius of the circles, type of surfaces, surface radius and axis coordinates of surface. The software evaluates and interprets the exacted data in terms of manufacturing and recognizes features for turned components. Rule based technique is used during recognizing process.
Nawara & Atia (2009) [9] presented a methodology of 3D prismatic parts classification based on the geometry of their machining features. The methodology works in three main phases. The first phase takes a neutral file in STEP-AP203 format as input, restructures it and extracts the geometric information of the machining features. The second phase recognizes the machining features through training neural network (NN) with a large set of feature patterns. The third phase classifies parts based on the variation in geometry of their machining features using self-organized map (SOM) NN.

III. PROPOSED METHODOLOGY

In this research, the STEP AP203 file is taken as an input to the developed system. STEP is an emerging international standard protocol for the exchange of technical product data. It enables all individuals contributing to the design, manufacturing, marketing and supply of a product and its components to contribute, to access, and to share information. Many major multinational companies have investigated significant resources for its development and implementation [3]. The main objective of the work presented in this paper is to recognize features for rotational parts. The developed system includes three modules. They are as follows:

Module 1: Development of a Geometric and Topological Data Extraction module to extract data from STEP AP203 file, to reorder, and to store them in a database.

Module 2: Development of Automatic Feature Recognition system which takes the extracted data as input, and consists in recognizing features according to certain prespecified rules that are characteristic to each feature.

Module 3: Development of a Feature Generator that analyses frontier surfaces of each feature surfaces and build new features from interacting to generates all possible combinations of interacting features.

IV. GEOMETRIC AND TOPOLOGICAL DATA EXTRACTION FROM STEP FILE

4.1. STRUCTURE OF STEP (AP-203) FILE

The STEP file is a text file that contains geometric and topological data of a component including boundary representation data such as shells, faces, edge loops, vertices; surface geometric data such as planes, cylinders, cones, toroidal, spherical; curve geometry such as lines, circles, B-splines, ellipses. The STEP text file is begun by the keyword ISO-10303-21and is terminated by keyword END-ISO- 10303-203. The high level of description in STEP is the shell. A shell is a topological item that is constructed by joining faces along edges. In Fig. 1, we have designed a Face, an Edge and a Vertex by F, E, and V respectively. There index are taken to be the same that of its pointers in STEP file. The domain of the shell is connected, oriented, finite, non-self-intersecting surface. A shell together with its domain is given explicitly in STEP physical file. For example, the shell in a partial STEP text file in Fig. 2 of the example part in Fig. 1 is given by the following records:
#51=CLOSED_SHELL('Closed Shell',(#91,#108,#148,#165,#179,#193)) where #91…#193 are pointers to the boundary faces that form the shell.

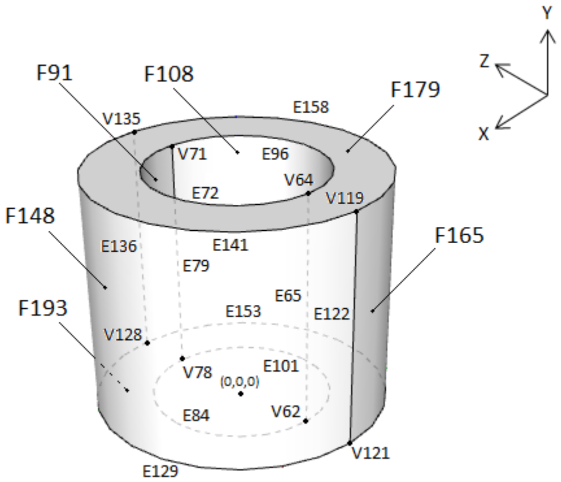


Figure 1: An example part for explanation of STEP file.

The second level of description is the face. A face is a topological entity that corresponds to the intuitive notion of a piece of surface, bounded by edge-loops. A face in STEP file is defined in terms of geometric and topological entities. A face in STEP file can be a plan, a cylinder, a cone, a torus, a hemisphere, or a surface of revolution. For example, the planar surface F179 of the part in Fig. 1 is given by the following records:
#179=ADVANCED_FACE('Corps principal',(#174,#178),#170,.F.) where #174, #178 are pointers to the face-bounds that bound the planar surface F179. Face bounds might be more than one. #170 is a pointer to the surface description of F179.

```
#51=CLOSED_SHELL('Closed Shell',(#91,#108,#148,#165,#179,#193))
;
#91=ADVANCED_FACE('Corps principal',(#90),#56,.F.) ;
#179=ADVANCED_FACE('Corps principal',(#174,#178),#170,.F.) ;
#56=CYLINDRICAL_SURFACE('generated cylinder',#55,30.) ;
#55=AXIS2_PLACEMENT_3D('Cylinder Axis2P3D',#52,#53,#54) ;
#52=CARTESIAN_POINT('Axis2P3D Location',(0.,30.,0.)) ;
#53=DIRECTION('Axis2P3D Direction',(0.,-1.,0.)) ;
#54=DIRECTION('Axis2P3D XDirection',(0,0.,1)) ;
#170=PLANE('Plane',#169) ;
#169=AXIS2_PLACEMENT_3D('Plane Axis2P3D',#166,#167,#168) ;
#90=FACE_OUTER_BOUND('',#85,.T.) ;
#174=FACE_OUTER_BOUND('',#171,.T.) ;
#178=FACE_BOUND('',#175,.T.) ;
#85=EDGE_LOOP('',(#86,#87,#88,#89)) ;
#171=EDGE_LOOP('',(#172,#173)) ;
#175=EDGE_LOOP('',(#176,#177)) ;
#86=ORIENTED_EDGE('',*,*,#65,.T.) ;
#87=ORIENTED_EDGE('',*,*,#72,.F.) ;
#88=ORIENTED_EDGE('',*,*,#79,.F.) ;
#89=ORIENTED_EDGE('',*,*,#84,.T.) ;
#65=EDGE_CURVE('',#62,#64,#60,.F.) ;
#72=EDGE_CURVE('',#71,#64,#69,.T.) ;
#79=EDGE_CURVE('',#78,#71,#76,.F.) ;
#84=EDGE_CURVE('',#78,#62,#83,.T.) ;
#62=VERTEX_POINT('',#61) ;
#64=VERTEX_POINT('',#63) ;
#71=VERTEX_POINT('',#70) ;
#78=VERTEX_POINT('',#77) ;
#63=CARTESIAN_POINT('Vertex',(0,60.,-30)) ;
#60=LINE('Line',#57,#59) ;
#69=CIRCLE('generated circle',#68,30.) ;
#76=LINE('Line',#73,#75) ;
#83=CIRCLE('generated circle',#82,30.) ;
#57=CARTESIAN_POINT('Line Origine',(0,30.,-30)) ;
#59=VECTOR('Line Direction',#58,1.) ;
#58=DIRECTION('Vector Direction',(0.,-1.,0.)) ;
#68=AXIS2_PLACEMENT_3D('Circle Axis2P3D',#66,#67,$) ;
#66=CARTESIAN_POINT('Axis2P3D Location',(0.,60.,0.)) ;
#67=DIRECTION('Axis2P3D Direction',(0.,-1.,0.)) ;
```

Figure 2: A partial STEP file of the part of Fig. 1.

Surface description is given explicitly in STEP file, for example, surface descriptions corresponding to the planar surface F179 and to the cylindrical surface F91 in Fig. 1 are given respectively by the following records:
#170=PLANE('Plane',#169)
#56=CYLINDRICAL_SURFACE('generated cylinder',#55,30.) where #169 is a pointer to the axis information of the planar surface F179, and #55 is the pointer to the axis information of the cylindrical surface F91. The last number without hash (#) in the CYLINDRICAL_SURFACE denotes the radius (30) of the cylindrical surface F91. Axis information is denoted as
#55=AXIS2_PLACEMENT_3D('Cylinder Axis2P3D',#52,#53,#54) where #52 is the pointer to the Cartesian point corresponding to the origin of the local coordinate system. A Cartesian-Point is an Address of a point or vertex in Cartesian space, denoted as:
#52=CARTESIAN_POINT('Axis2P3D Location',(0.,30.,0.)).
#53 is a pointer to the direction of the local z-axis, denoted as:
#53=DIRECTION('Axis2P3D Direction',(0.,-1.,0.)), and #54 is the pointer to the direction of the local x-axis necessaries to define completely the local system, which is given as:
#54=DIRECTION('Axis2P3D XDirection',(0,0.,1)). Directions of the x-axis and z-axis are given with respect to the global coordinate system of the part. The local z-axis corresponds to the normal of planar and spherical surfaces and to the axis of cylindrical, conical and toroidal surfaces.

A face-bound is a loop of connected edges used for bounding a face. Two sub-types called face-outer-bound and face-bound that define respectively an outer edge-loop and an inner edge-loop of a face. For example, face-bounds that bound F179 are given by following records:
#174=FACE_OUTER_BOUND('',#171,.T.)
#178=FACE_BOUND('',#175,.T.) where #171, #175 are pointers to the edge-loops.

Edge-loop is a topological entity that starts and ends at the same vertex point, it is formed by joining oriented edges. For example, edge loop that bounds F91is given by:
#85=EDGE_LOOP('',(#86,#87,#88,#89)) where #86…#89 are pointers to oriented-edges.

An oriented-edge is an edge constructed from another (original) edge and containing the direction (orientation) information. An edge can be a line, a circle, a Bspline curve with knots of an ellipse. For example, an oriented edge among four that form the cylindrical surface F91 are given by:
#86=ORIENTED_EDGE('',*,*,#65,.T.) where #65 is the pointers to the edge curve description.

Edge curve description is a geometric entity which has its geometry fully defined. For example, edge curves description in STEP file are given by the following records:
#65=EDGE_CURVE('',#62,#64,#60,.F.) where #62 is a pointer to the starting point of the edge, #64 is a pointer to the ending point, and #60 is a pointer to the type of edge. A Vertex-Point is a point defining the geometry of a vertex. Start vertex point and end vertex point are given respectively by the following records:
#62=VERTEX_POINT('',#61)
#64=VERTEX_POINT('',#63) where #63 is a pointer to the coordinates of the vertex point, denoted as
#63=CARTESIAN_POINT('Vertex',(0,60.,-30))
Types of edges are given explicitly by:
#60=LINE('Line',#57,#59);
#69=CIRCLE('generated circle',#68,30.) where#57 is the pointer to the point corresponding to the center of the line,#68 is the pointer that contains the direction and the point corresponding to the center of the circle. The last number without hash (#) in the CIRCLE denotes the radius (30) of the circle. #59 is the pointer to the vector of the line. The vector is a geometric entity related to the direction of a line or a surface of linear extrusion, denoted as:
#59=VECTOR('Line Direction',#58,1.) where #58 is the pointer to the direction of the line which is

denoted as #58=DIRECTION('Vector Direction',(0.,-1.,0.)).

4.2. GEOMETRIC AND TOPOLOGICAL DATA EXTRACTION (GTDE) FROM STEP AP203 FILE

The STEP file starts with a string CLOSED_SHELL and ends with the coordinates of points such as circle centers, line centers, cylinder centers, vertices and so on. Extraction of various strings and entities (# number) from STEP file is done according to the hierarchical structure shown in Fig. 3. In this research, C++ language is used to search various strings and entities in the STEP file. C++ is based on object oriented programming, as a result, representation and extraction of geometric and topological data from STEP file become simple and easy. The command string::size_type loc = str.find(" CYLINDRICAL_SURFACE ", 0) is an example of one of the functions that can be used to locate a specified text string. Like this all the strings are located and stored in a database in such a way that the extracted data are coherently ordered.

V. AUTOMATIC FEATURE RECOGNITION (AFR)

The proposed system for feature recognition has its own method for the evaluation of the extracted geometric and topological data obtained from the GTDE module. geometric data analysis addresses the evaluation of circle centers coordinates, cylinder centers coordinates, taper centers coordinates, torus centers coordinates, hemisphere centers coordinates, circle radii, cylinder radii, circles, lines, surfaces, and so on. Topological data analysis concerns the evaluation of loops, edge curve construction, inner bounds, outer bounds, vertices and so on. This system possesses a database of thirty turning-pre-defined manufacturing features including internal features, external features, and special features such as threads and recesses.

The developed system adopts the rule-based approach that uses algorithms to identify a feature according on certain prespecified rules that are characteristic to the feature. It must be noticed that the coordinate system that was adopted is XY (X for the diameter and Y for the length). Some features and their rules for recognition are illustrated in Table 1. In this paper, the groove feature has been proposed to use in feature recognition system for explanation of above aspects. An example part which contains a groove feature and a partial STEP file of the same example part are given in Fig. 4 and Fig. 5 respectively. The following set of heuristic rules is used to describe recognition of a groove.

- **Geometric rules**

Rule 1:
The STEP data must contain two CYLINDRCAL_SURFACEs with different radius (#56, #113) and circles centers (#123, #149, #137 and #154) of the cylinder with minor radius (#113) lie between centers of end circles (#80, #97, #175 and #201) of the cylinder with major radius.

Rule 2:
X and Z coordinates of cylinders centers and circles centers are the same, and Y coordinate is different.

Rule 3:
The length of the feature must be less than 16mm. The length of the groove is equal to the distance between end circle centers of the cylinder with minor radius.
The depth of the groove is equal to the difference between the big and the small radius.

- **Topological rules**

Rule 1:
Edge curve construction of CYLINDRICAL_SURFACEs must be; line, circle, line, circle (#117, #126, #133 and #140), or; circle, line, circle, line.

Rule 2:
Every CYLINDRICAL_SURFACE (ADVANCED_FACE) shares two common linear edges with another of the same type.

Rule 3:
STEP data must contain two PLANEs (#232, #246) bounded respectively each by an inner loop (#240, #254) and an outer loop (#236, #250). Circles that form each inner loop are the same circles of the cylinder with minor radius (#140, #157, #152 and #126). Circles that form each outer loop are four circles (#95, #69, #192 and #209) among eight of the cylinder with major radius.

VI. FEATURE GENERATOR (FG)

It must be noticed that the developed system for feature recognition correctly recognizes isolated features at a first stage. Feature extraction and classification become complex in the case of interacting features [10, 11]. Some surfaces of features are lost by interactions, as a result, the system can give wrong results concerning the type of features. To overcome this issue, a system for generating new features from interacting features is required. Before describing the methodology followed for construction of new manufacturing features, it will be important at this stage to define the following terms: Perfect Manufacturing Feature PMF, and Imperfect Manufacturing Feature IMF.

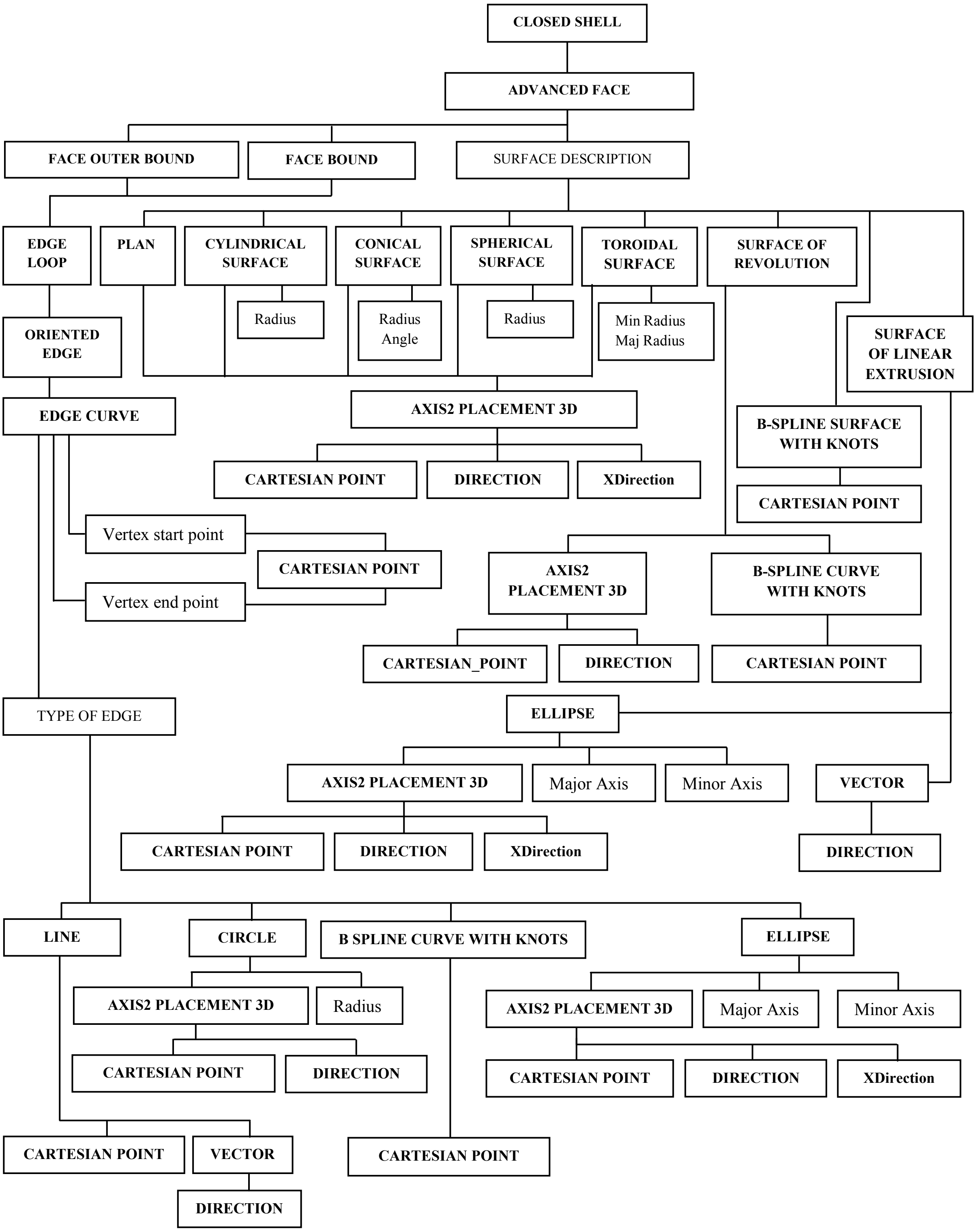
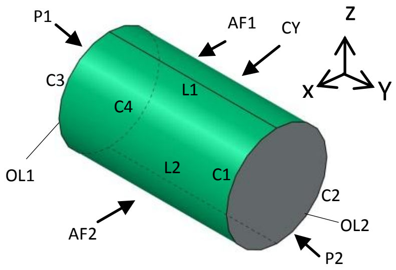
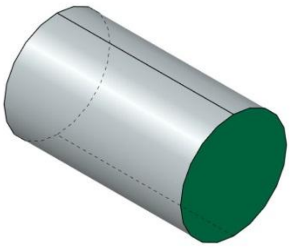
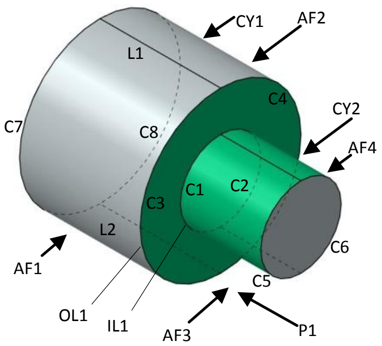


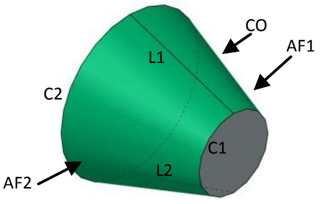
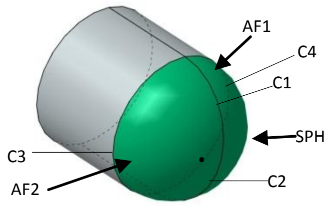
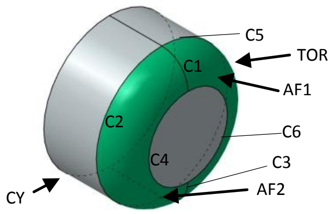
Figure 3: Hierarchical structure of STEP AP203

A MF is considered as PMF if:

a. All surfaces that constitute the feature are material surfaces.
b. All edges shared by its adjacent surfaces are material edges.
c. The topological and geometrical conditions between the surfaces that define the feature are valid.
d. All faces except those for defining MF are blank (stock) surfaces.

Table1: Some turning manufacturing features

Name of features	Features	Rules for recognition
Cylinder		**Rule 1**: The STEP data must contain **CYLINDRCAL_SURFACE** (**CY**) and the radius of the cylinder is same at both end circles. **Rule 2**: X and Z coordinate of cylinder center and circles centers are same and Y coordinate is different. **Rule 3**: The difference between the diameter of the stock and the final diameter is different to zero. **Rule 4**: Edge curve construction of cylindrical **ADVANCED_FACEs** (**AF1**,**AF2**) must be; line, circle, line, circle (e.g. **L1**, **C1**, **L2**, **C3**), or; circle, line, circle, line. **Rule 5**: Both cylindrical **ADVANCED_FACEs** share two common linear edges (**L1**, **L2**). **Rule 6**: STEP data must contain two **PLANE**s (**P1**, **P2**) bounded by outer edge-loops (**OL1**, **OL2**). Circles that form each outer loop are the same circles of the cylinder.
Face		**Rule 1**: The STEP data must contain **CYLINDRCAL_SURFACE** and the radius of the cylinder is same at both end circles. **Rule 2**: X and Z coordinate of cylinder center and circles centers are same and Y coordinate is different. **Rule 3**: The difference between the length of the stock and the final length is different to zero. **Rule 4**: Edge curve construction of cylindrical **ADVANCED_FACEs** must be; line, circle, line circle, or; circle, line, circle, line. **Rule 5**: Both cylindrical **ADVANCED_FACEs** share two common linear edges. **Rule 6**: STEP data must contain two **PLANE**s bounded by outer edge-loops. Circles that form each outer loop are the same circles of the cylinder.
Shoulder		**Rule 1**: The STEP data must contain two **CYLINDRCAL_SURFACEs** **(CY1**, **CY2)** with different radius and centers coordinates of four circles of cylinders are same (**C1**, **C2**, **C3**, **C4**). **Rule 2:** The common circles center of previous four circles lies between the remaining end circles centers (**C5**, **C6**, **C7**, **C8**). **Rule 3**: X and Z coordinates of cylinders centers and circles centers are same and Y coordinate is different. **Rule 4**: Edge curve construction of cylindrical **ADVANCED_FACE**s (e.g. **AF1**) must be; line, circle, line, circle (**L1**, **C3**, **L2**, **C7**), or; circle, line, circle, line. **Rule 5**: Every two cylindrical **ADVANCED_FACEs** (e.g. **AF1**, **AF2**) share two common linear edges (**L1**, **L2**). **Rule 6**: STEP data must contain a **PLANE** (**P1**) bounded by an inner loop (**IL1**) and an outer loop (**OL1)**. Circles that form the outer loop are the same circles of the cylinder with major radius (**CY1**). Circles that form the inner loop are the same circles of the cylinder with minor radius (**CY2**). **Rule 7:** Linear edge directions of the cylinder with

Name of features	Features	Rules for recognition
		minor radius are from the right to the left (the origin of the Cartesian coordinate is taken to be at the left end of each part).
Right hand taper		**Rule 1**: The STEP data must contain **CONICAL _SURFACE** (**CO**) with minor circle radius at the right end. **Rule 2**: X and Z coordinate of taper center and circles centers are same and Y coordinate is different**.** **Rule 3**: Edge curve construction of conical **ANDVANCED_FACEs** (e.g. **AF2**) must be; line, circle, line circle (**L1**, **C1**, **L2**, **C2**), or; circle, line, circle, line. **Rule 4**: Every **conical ADVANCED_FACE** shares two common linear edges (**L1**, **L2**) with another of the same type.
Himisphere		**Rule 1**: The STEP data must contain **SPHERICAL_SURFACE (SPH)** and the radius of hemisphere is equal to circles radius. **Rule 2**: X, Y, and Z coordinates of hemisphere center and circles centers are same. **Rule 6**: Edge curve construction of spherical **ADVANCED_FACEs** (e.g. **AF1**) must be, circle, circle, circle (**C1, C2, C4**). **Rule 7**: Every spherical **ADVANCED_FACEs** shares two common circular edges (**C1**, **C2**) with another of the same type.
Round		**Rule 1**: The STEP data must contain **TOROIDAL_SURFACE** (**TOR**) and **CYLINDRICAL_SURFACE** (**CY**)**.** **Rule 2**: X and Z coordinate of torus center, cylinder center and circles centers are same and Y coordinate is different**.** Circles centers that represent torus arcs are not included. **Rule 3**: Edge curve construction of toroidal **ADVANCED_FACEs** (**AF1, AF2**) must be; circle, circle, circle, circle (**C1, C2, C3, C4**). **Rule 4**: Every toroidal **ADVANCED_FACE** shares two common circular edges (**C1, C3**) with another of the same type. **Rule 5**: Circles with major radius (**C2, C5**) of tori are the same circles of a cylinder. Circles with minor radius (**C4, C6**) of tori are the same circles of a plan.

A MF is considered as IMF if:

e. It exists a MF for which the topologic and geometric criterions are satisfied but it exists at least one adjacent surface of that MF which is not a blank surface.
f. Conditions a to c of PMF are also satisfied for the MF concerned.

To explain the methodology for construction of interacting (imperfect) manufacturing features, the following procedural rules must be known:

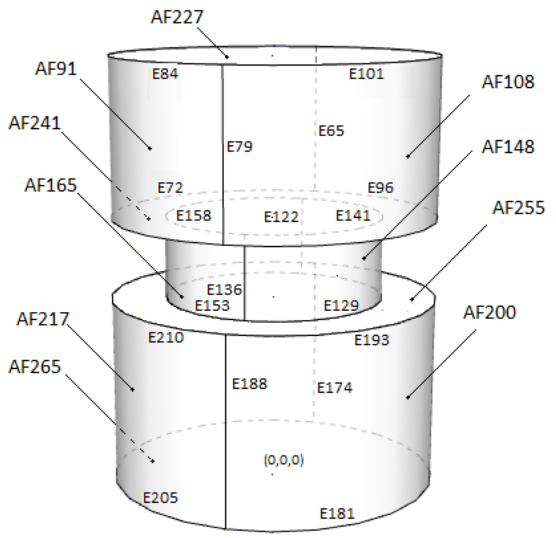


Figure 4: An example part containing a groove feature.

A blank surface is a surface that can be a cylinder having the same radius as the raw cylinder radius, or a plan having the same coordinates as two circles coordinates of the raw cylinder. An adjacent (frontier) surface to the blank is a surface for which a circular edge loop has the same radius as the raw cylinder radius, or circle centers coordinates of that loop and circles centers coordinates of the raw cylinder are same.

Taking the fact that surfaces that form each feature are known, the Feature Generator system consists in analyzing adjacent (frontier) surfaces of each feature surfaces, to distinguish between PMF and IMF. Note that some features such as recesses and nested recesses are not treated by the Feature Generator, they are added to the part as material rings to facilitate construction of IMF. Construction of a new MF is based on extending its material surfaces until the blank (stock) surfaces and/or material surfaces. By this way, an IMF is transformed to a PMF. Once a feature is built, it is removed from the blank, and after, material surfaces of a following feature are extended until the last new stock surfaces and/or material surfaces. The operation is repeated until that the stock reaches the final workpiece. Combinations of features are obtained according to the number of IMF. If the number of IMF is N, there are N! ways

```
#56=CYLINDRICAL_SURFACE('generated cylinder',#55,40.) ;
#52=CARTESIAN_POINT('Axis2P3D Location',(0.,35.,0.)) ;
#113=CYLINDRICAL_SURFACE('generated cylinder',#112,20.) ;
#109=CARTESIAN_POINT('Axis2P3D Location',(0.,35.,0.)) ;
#69=CIRCLE('generated circle',#68,40.) ;
#66=CARTESIAN_POINT('Axis2P3D Location',(0.,50.,0.)) ;
#95=CIRCLE('generated circle',#94,40.) ;
#92=CARTESIAN_POINT('Axis2P3D Location',(0.,50.,0.)) ;
#83=CIRCLE('generated circle',#82,40.) ;
#80=CARTESIAN_POINT('Axis2P3D Location',(0.,70.,0.)) ;
#100=CIRCLE('generated circle',#99,40.) ;
#97=CARTESIAN_POINT('Axis2P3D Location',(0.,70.,0.)) ;
#126=CIRCLE('generated circle',#125,20.) ;
#123=CARTESIAN_POINT('Axis2P3D Location',(0.,30.,0.)) ;
#152=CIRCLE('generated circle',#151,20.) ;
#149=CARTESIAN_POINT('Axis2P3D Location',(0.,30.,0.)) ;
#140=CIRCLE('generated circle',#139,20.) ;
#137=CARTESIAN_POINT('Axis2P3D Location',(0.,50.,0.)) ;
#157=CIRCLE('generated circle',#156,20.) ;
#154=CARTESIAN_POINT('Axis2P3D Location',(0.,50.,0.)) ;
#178=CIRCLE('generated circle',#177,40.) ;
#175=CARTESIAN_POINT('Axis2P3D Location',(0.,0.,0.)) ;
#204=CIRCLE('generated circle',#203,40.) ;
#201=CARTESIAN_POINT('Axis2P3D Location',(0.,0.,0.)) ;
#192=CIRCLE('generated circle',#191,40.) ;
#189=CARTESIAN_POINT('Axis2P3D Location',(0.,30.,0.)) ;
#209=CIRCLE('generated circle',#208,40.) ;
#206=CARTESIAN_POINT('Axis2P3D Location',(0.,30.,0.)) ;
#148=ADVANCED_FACE('Corps principal',(#147),#113,.T.) ;
#147=FACE_OUTER_BOUND('',#142,.T.) ;
#142=EDGE_LOOP('',(#143,#144,#145,#146)) ;
#143=ORIENTED_EDGE('',*,*,#122,.T.) ;
#144=ORIENTED_EDGE('',*,*,#129,.F.) ;
#145=ORIENTED_EDGE('',*,*,#136,.F.) ;
#146=ORIENTED_EDGE('',*,*,#141,.T.) ;
#122=EDGE_CURVE('',#119,#121,#117,.T.) ;
#129=EDGE_CURVE('',#128,#121,#126,.T.) ;
#136=EDGE_CURVE('',#135,#128,#133,.T.) ;
#141=EDGE_CURVE('',#135,#119,#140,.T.) ;
#117=LINE('Line',#114,#116) ;
#126=CIRCLE('generated circle',#125,20.) ;
#133=LINE('Line',#130,#132) ;
#140=CIRCLE('generated circle',#139,20.) ;
#241=ADVANCED_FACE('Corps principal',(#236,#240),#232,.T.) ;
#232=PLANE('Plane',#231) ;
#236=FACE_OUTER_BOUND('',#233,.T.) ;
#233=EDGE_LOOP('',(#234,#235)) ;
#234=ORIENTED_EDGE('',*,*,#96,.T.) ;
#235=ORIENTED_EDGE('',*,*,#72,.T.) ;
#96=EDGE_CURVE('',#64,#71,#95,.T.) ;
#72=EDGE_CURVE('',#71,#64,#69,.T.) ;
#240=FACE_BOUND('',#237,.T.) ;
#237=EDGE_LOOP('',(#238,#239)) ;
#238=ORIENTED_EDGE('',*,*,#141,.F.) ;
#239=ORIENTED_EDGE('',*,*,#158,.F.) ;
#141=EDGE_CURVE('',#135,#119,#140,.T.) ;
#158=EDGE_CURVE('',#119,#135,#157,.T.) ;
#255=ADVANCED_FACE('Corps principal',(#250,#254),#246,.F.) ;
#246=PLANE('Plane',#245) ;
#250=FACE_OUTER_BOUND('',#247,.T.) ;
#247=EDGE_LOOP('',(#248,#249)) ;
#248=ORIENTED_EDGE('',*,*,#193,.F.) ;
#249=ORIENTED_EDGE('',*,*,#210,.F.) ;
#193=EDGE_CURVE('',#187,#171,#192,.T.) ;
#210=EDGE_CURVE('',#171,#187,#209,.T.) ;
#254=FACE_BOUND('',#251,.T.) ;
#251=EDGE_LOOP('',(#252,#253)) ;
#252=ORIENTED_EDGE('',*,*,#153,.T.) ;
#253=ORIENTED_EDGE('',*,*,#129,.T.) ;
#153=EDGE_CURVE('',#121,#128,#152,.T.) ;
#129=EDGE_CURVE('',#128,#121,#126,.T.) ;
```

Figure 5: A partial STEP file of the example part of Fig. 4.

(combinations) to build N features. Otherwise, there are N! manners to machine N features. The flowchart of Fig. 6 shows the methodology of features classification and generation. It must be noticed that a workpiece can be machined in two set-ups if the maximum diameter lies in the middle of the part [12]. The maximum number of regions that a part can be subdivided in is four, left exterior region, right exterior region, left interior region,

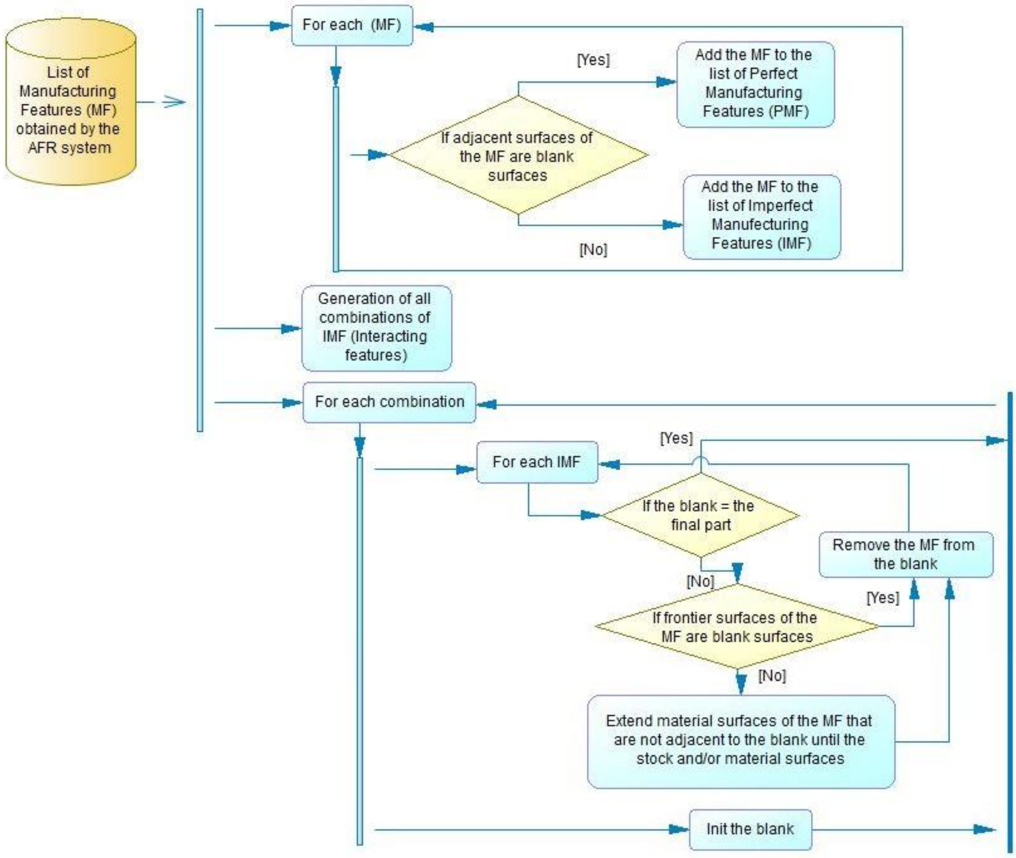


Figure 6: UML flowchart of the Feature Generator system

and right interior region, each region must be treated separately by the Features Generator system in order not to confuse interactions between features from different regions. Thus, if Ni! is the number of ways to build N features of the ith region, the total number of manners to machine the workpiece is equal to the product of Ni!. At a second stage, and after construction of new perfect manufacturing feature, which are considered as isolated features, the features library is called a second stage in order to confirm the type of features, and to extract their new dimensional parameters.

VII. CASE STUDY

The example part shown in Fig. 7 has been modeled in CATIA V5 and it is the same of that presented in Fig. 8 by Sreeramulu & Rao [2]. This part is used only to clarify the method developed and not to test its limitations. It must be known that special features such as radial and axial holes and threads are not treated by the feature generator since threading is the last operation after a turning process, and holes for being Non-axisymmetric features. Geometric and topological data of the part are extracted from STEP APA203 data file by the GTDE module. These data are analyzed by the feature recognizer module to extract features at a first stage. The feature generator module takes the extracted features (three shoulders) and distinguishes between PMF and IMF. Material surfaces of IMF are used to build new PMF. Taking the fact that three shoulders are recognized, the number of possibilities to machine the part is six (3!). Two possibilities are similar since features are the same in a dimensional point of view. Thus, five different solutions are illustrated in Fig. 7. At a second stage, the feature recognizer module is applied again to confirm the type of features and to extract the new parameters of features.

VIII. CONCLUSION

In the present work, we have explained a new methodology to interlink between CAD and CAPP systems based on features recognition for rotational parts. In a first module, geometric and topological data of the part are extracted from STEP file, coherently ordered and stored in a database. In the second module, the extracted data are used as input for the feature recognition system which consists in recognizing features according to certain prespecified rules that are characteristic to each feature. In the last module, the features generator system analyses frontier surfaces of each feature

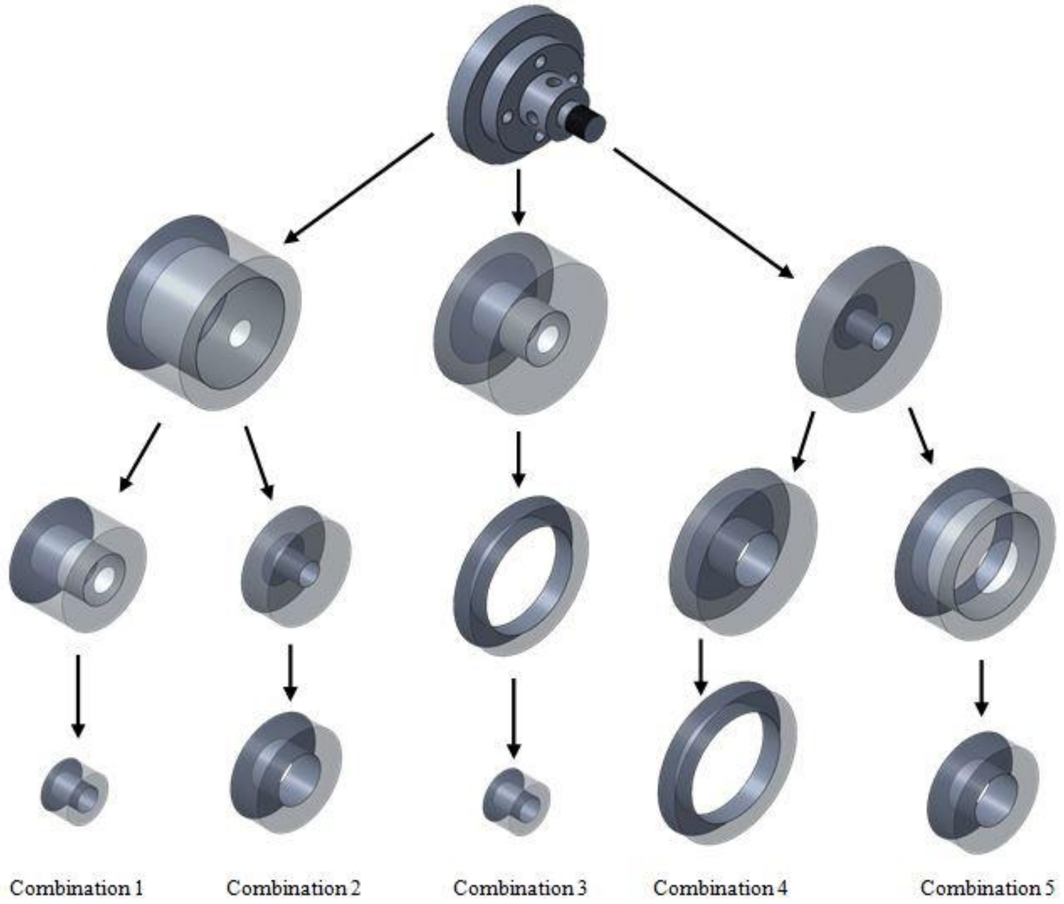


Figure 7: The five possible combinations generated by the Features Generator module for an example part of the literature.

surfaces and build new manufacturing features from interacting features by extending their material surfaces until the blank or material surfaces of the part. Finally, features recognition is applied a second time to confirm the type of features and to extract their new dimensional parameters.

It is clear that the Features Generator system gives multiple combinations of interacting features to machine a same workpiece, which can lead to a computational combinatorial explosion at the feature recognition stage, and time consuming at the tool selection stage, Thus, in a future paper, we will reduce the number of combinations by taking into consideration on one hand, metal removal principles and manufacturing rules, since some features such as facing, and longitudinal turning of the maximum diameter of the finished part are frequently performed first in any turning process, others such as grooves and recesses can be excluded from features interactions, because from the point of view of manufacturing rules applied by experts in the real manufacturing environment, two cylinders having the same diameter have to be machined sequentially which causes a long tool path into blank diameter of the groove or the recess. And on the other hand, Geometric dimensioning and Tolerancing and economical and technological constraints that create precedence between surface of the part and then, between features. By this way, combinations of feature that violate these constraints, and combinations eliminated by manufacturing rules and metal removal principals, will be deleted and only the remaining combinations will be transferred to downstream activities such as automatic tool selection.

REFERENCES

[1] S. Sivakumar, V. Dhanalakshmi, A feature-based system for CAD/CAM integration through STEP file for cylindrical parts, *Indian Journal of Engineering & Material Sciences*, *vol. 20*, February 2013, pp. 21-26.

[2] D. Sreeramulu, C.S.P.Rao, A new methodology for recognizing features in

rotational parts using STEP data exchange standard, *International Journal of Engineering, Science and Technology*, *Vol. 3, No. 6*, 2011, pp. 102-115.

[3] V. N. Malleswari, Dr. P. M.Valli, Dr. M. M. M. Sarcar, Automatic Recognition of Machining Features using STEP Files, *International Journal of Engineering Research & Technology (IJERT)*, ISSN: 2278-0181, *Vol. 2 Issue 3*, March – 2013.

[4] M. Kanga, J. Han, J.G. Moon, An approach for interlinking design and process planning, *Journal of Materials Processing Technology 139* (2003) 589–595.

[5] STEP Application Protocol (AP) 203 editions 1 & 2, Configuration Controlled 3D Designs of Mechanical Parts and Assemblies, ISO10303 1994, (2007) & (2009).

[6] N. Ismaila, N. Abu Bakar, A.H. Juri, Recognition of cylindrical and conical features using edge boundary classification, *International Journal of Machine Tools & Manufacture 45* (2005) 649–655.

[7] Mohamed El-Mehalawi, R. Allen Miller, A database system of mechanical components based on geometric and topological similarity. Part I: representation, *Computer-Aided Design 35* (2003) 83-94.

[8] Emad S. Abouel Nasr, Ali K. Kamrani, A new methodology for extracting manufacturing features from CAD system, *Computers & Industrial Engineering 51* (2006) 389–415.

[9] G. Nawara, M. A. ELBaz, A. M. Attia, Parts Classification Based on Solid Model and Neural Networks, *Contemporary Engineering Sciences*, *Vol. 3, 2010, no. 8*, 395 – 417.

[10] Tobias Dipper, Xun Xu, Peter Klemm, Defining, recognizing and representing feature interactions in a feature-based data model, *Robotics and Computer-Integrated Manufacturing 27* (2011) 101–114.

[11] A. Elmesbahi, A new methodology for recognition of interacting features using Frontier Faces of Base Face, CPI 2009, 19-21 October 2009, Fez, Morocco.

[12] Ali Oral, M. Cemal Cakir, Automated cutting tool selection and cutting tool sequence optimisation for rotational parts,” *Robotics and Computer-Integrated Manufacturing 20* (2004) 127–141.