



# A novel and smart interactive feature recognition system for rotational parts using a STEP file

Mazin Al-wswasi<sup>1,2</sup> · Atanas Ivanov<sup>1</sup>

Received: 15 November 2018 / Accepted: 6 May 2019 / Published online: 30 May 2019  
© The Author(s) 2019

## Abstract

Sharing product design information with other downstream applications, such as process planning, is a major barrier to develop an integrated manufacturing system. Part of this shortcoming is due to the difference in product data descriptions, since a design is geometry based, whereas process planning is manufacturing feature based. The implementation of automatic feature recognition (AFR) techniques is considered an indispensable concept for transferring product data between computer-aided design (CAD) and automatic computer-aided process planning (ACAPP). This is accomplished using one of the international Product Data Exchange (PDE) standards, such as DXF, IGES, or STEP files. Despite different AFR techniques and systems having been developed to serve this aim, each of them has limitations. The most important limitation is that each system is restricted to a specific set of predefined manufacturing features. This means that even when the system tries to cover as many as possible of the existing features that are predefined, it is always possible to create a new feature based on the specific requirements and designer creativity. Consequently, the new feature is not included in the compiled database and, hence, will not be recognised. This paper presents a novel and smart interactive AFR (SI-AFR) methodology for recognising the features of rotational parts, taking a STEP AP203, AP214, or AP242 file as an input to the system. This has been written using C# coding to extract the features' geometrical and topological information from a STEP file, building a database containing a set of 54 predefined features, whilst also smartly learning how to recognise new features and adding them to the set. Several examples have been processed for this paper to validate the system, and based on the results, it is anticipated that it will lead to the launching of a new generation of feature recognition systems.

**Keywords** Smart automatic feature recognition · Computer-aided process planning · STEP AP203 · Rotational parts

## 1 Introduction

Due to the continuous changes in customers' needs and the increase in global market competition, manufacturing companies aim to launch their products with the required quality, minimal time-to-market, and lower cost [1]. In order to achieve these goals, part features, as represented in CAD, must be identified and linked with CAPP, and consequently with CAM, in a fast, accurate, and controllable way. In conventional manufacturing, which includes the use of CNC machines, manufacturers translate a CAD drawing of a

product into detailed manufacturing instructions as CAPP at a given factory. The integration between CAD and CAPP systems in this manner requires significant effort, time, and cost. Part of this drawback is related to the different natures of CAD and CAPP systems, since the former is geometrical and topological based, whereas the latter is manufacturing feature based. Thus, product data provided by a CAD platform cannot be used directly in CAPP systems [2]. Nowadays, many researchers are working on ways to automate the transference of the part information among downstream applications as a part of smart manufacturing requirements. This involves upgrading CAPP systems to the next level in the form of automatic computer-aided process planning (ACAPP). In this field of research, several AFR technologies have been developed and implemented in different systems to serve this purpose [3]. The five main categories of AFR are as follows: syntactic pattern recognition, graph based, hint based, logic rule based, and artificial neural networks [4, 5].

✉ Mazin Al-wswasi  
Mazin.Al-Wswasi@brunel.ac.uk

<sup>1</sup> Department of Mechanical, Aerospace and Civil Engineering (MACE), Brunel University, London, UK

<sup>2</sup> Department of Production Engineering and Metallurgy, The University of Technology, Baghdad, Iraq

In all AFR techniques, the relations between low-level geometric entities, such as points, lines, and curves created in CAD systems, are automatically extracted and converted to high-level manufacturing features, for example, holes, grooves, and pockets [5, 6]. For this purpose, a part CAD representation is saved and exported using one of the international Product Data Exchange (PDE) standards to facilitate the interface between different CAD/CAPP/CAM systems. There are different PDE standards available, with the most recognised being DXF, IGES, and STEP files [7]. Any PDE file is analysed and translated to a part representation suitable for form feature recognition, which is achieved using an interface program written in Java, Prolog, C++, C# or some other programming language. Usually, the same programming language is utilised to create a set of predefined features to be used as a reference in recognising the part features. Whilst different PDE standards have been implemented in different AFR systems, special attention has been paid to the STEP file, which includes advantages that outweigh those of any other format. For example, the STEP file transfers both geometrical and non-geometrical information of a product through its life cycle. The definition of the physical shape including vertices, edges, curves, surfaces, and relations are examples of the former, whilst materials, view, general note, witness line, leader, and associativity entities represent the latter. Furthermore, the STEP file groups mechanical elements in a certain view [7]. However, the use of STEP files in AFR systems does not eliminate the following limitations of these systems: (i) the recognition of a specific set of predefined features and (ii) the limited ability of learning how to recognise a new feature.

In this paper, a novel and smart interactive AFR (SI-AFR) system has been developed for rotational part feature recognition. The system accepts any of the STEP AP203, AP214, or AP242 files format as an input, which can be created in many CAD packages representing the constructive solid geometry (CSG) of a part. However, the structure of the STEP AP203 will be explained in this work only as an example. The proposed SI-AFR system consists of five modules. In the first, a parser is developed for the purpose of restructuring and presenting the geometrical and topological information of STEP files in a readable way for both machines and human. The second module includes different algorithms, which are used to manipulate the result from the parser in order to facilitate the task of the feature recogniser. In the meantime, a set of 54 predefined features is created following the concept of the AFR logic rule-based technique. This predefined set of features is saved in the system's database, and represents the third module. Figure 1 shows examples of some external and internal predefined

features. The fourth module comprises an analysis of the part's geometrical and topological information by matching it with the predefined features in order to achieve feature recognition. Whilst these four steps can be found in many AFR systems, they have different abilities regarding the number and types of features they can recognise. For example, in contrast with other AFR systems, the proposed SI-AFR can recognise a square groove that has extra details and solves feature's intersecting issues without the need for extra processing as it will be shown in Section 4.1. The fifth module involves a smart interactive detector such that if an undefined feature is found, the system provides its complete geometrical and topological information, whilst also allowing the user to name it. The new feature is then saved, in terms of the geometrical and topological information, and added to the database of features. For example, if the SI-AFR system is used for the first time and a new feature is found, the total number of the predefined features will be 55 instead of the 54 that are already defined.

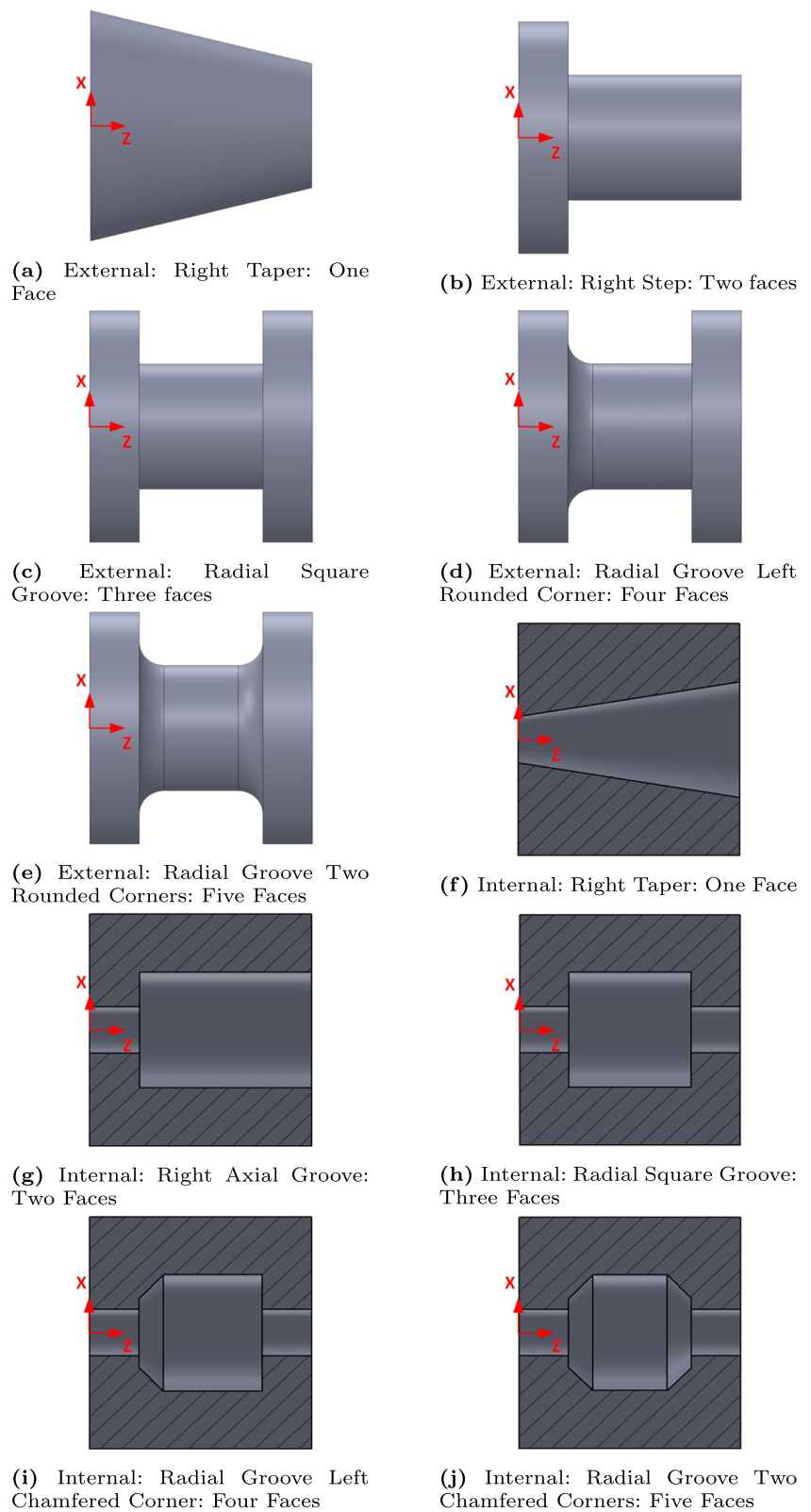
## 2 Previous work critical review

The direct recognition of features from a CAD model is an essential task in any CAPP system and consequently in CAM [8]. Hence, the integration of CAD/CAM by AFR and CAPP systems is considered one of the secrets to the success of small- or medium-sized manufacturing industry. Nowadays, small- and medium-sized factories have become prevalent and play a crucial role in production [9]. Accordingly, many AFR techniques and systems have been developed in order to recognise manufacturing features from CAD models automatically.

### 2.1 Syntactic pattern recognition

Ketan and Yaqoub [10] proposed an AFR system for recognising the symmetrical features of rotational parts. In this system, a STEP file is used to save a 2D upper profile of the part from a CAD package. The system also includes the implementation of the Sweeping Primitive Rule (SPR) algorithm, which was constructed based on three techniques: syntactic pattern recognition, sweeping operation, and logic rule based. An example was presented to validate the approach using mechanical desktop 0.9 in order to draw the part profile. Whilst a 2D part representation can be created using lines and arcs, in this work the part profile is restricted to containing only line segments. Also, the system is limited to recognising thirteen predefined symmetrical rotational features.

**Fig. 1** Examples of external and internal predefined features



Meeran and Pratt [11] applied both syntactic pattern recognition and intelligent logic techniques to recognise the features of a 2D prismatic part design using DXF

file. The entities in the DXF file are classified and resorted into three groups pertaining to the front, side, and top views. These three groups are then exported

to the recogniser, which is provided with a library of predefined features based on syntactic pattern recognition, for recognising and categorising simple isolated features. Loops of entities that correspond to predefined features are removed from the database, whilst the remaining entities are examined using intelligent logic to check whether they correspond to one of three possibilities: intersecting, protrusion, or depression features. According to the authors, the sequence of recognition processes is not in any sense optimal. Furthermore, the adopted approach is capable of recognising very specific and simple feature types along with more general protrusion and depression ones.

Yip-Hoi et al. [12] utilised a 2D area decomposition method for recognising turned features from axisymmetric parts. It starts with identifying the machining axis and determining the Maximum Turn-able State (MTS) of the part, which is the state of the part in which moving more material will gouging surfaces of the final part. The maximum turn-able volume (MTV) is the material that can be removed by turning, and it is found by subtracting the MTS from the stock workpiece. The system consists a set of predefined features with a unique combination of elements for each feature. A pattern recognition algorithm is used to search for matching between the maximum turn-able volumes (MTVS) decomposition and each predefined feature pattern. After finishing the feature recognition, three steps of precedence generation are included: determining the feature's ordering, mapping volume-to-operations, and mapping operation-to-resource. This is to be used as a part of a CAPP system. However, the implementation of this method was limited for the 2D patterns that contains only straight-lines segments. Based on the authors, it is possible to extend it to include arcs or other curved segments.

## 2.2 Rule based

Sivakumar and Dhanalakshmi [13] developed an AFR logic rule-based methodology to extract manufacturing features for cylindrical parts. The system comprises four major units, with the first including a part model created using Unigraphics and saved as a STEP file. A set of nine different cylindrical features was defined in the second unit. Next, a feature analyser was developed using the C++ language to extract a part's features and save it in a text file. Finally, the extracted data with other information, such as type of material and cutting conditions, were used to generate an NC code. During the experimental work, the authors claimed a range of deviations in dimensions between 0.082 and 4.125% from the reference data. However, eight of the nine predefined features were identified based on one surface and hence, other features that require more than one surface were not included, for example, a square groove that comprises three adjacent surfaces.

## 2.3 Attributed adjacency graph based

In a multi-tasking machine, both turning and milling machining can be performed. ZHU et al. [14] applied the concept of attributed adjacency graph (AAG) to recognise cylindrical and prismatic features, which makes it valuable in multi-tasking machines. According to the authors, the system is capable of generating process planning from a CAD model, which is achieved by saving it as a STEP file and sending it to the feature recogniser. In total, the system is able to recognise nine turning features and eight milling ones. Then, the information of the manufacturing features is returned to the user, in order to select the cutting tools manually or automatically. The final step includes the generation of process planning, which comprises the sequence of operations and the NC tool path data. The generated process planning is optimised based on the machining cost evaluation.

## 2.4 Hint based

Kang et al. [15] presented an approach to link CAD and CAPP systems via STEP files and a hint-based technique. The system includes four steps, starting with the generation of a STEP 203 file from UniGraphics CAD software. Next, the part's geometrical information of the STEP 203 is transformed into Parasolid entities. This is to check for the correctness of the STEP 203 import and to assign other information to the following step, such as surface roughness, and dimensional or geometric tolerances. In the third step, the hint-based concept is implemented to convert the geometric features into manufacturing features by applying the Integrated Incremental Feature Finder (IF2). Finally, the resultant manufacturing information is translated into a physical STEP AP224 file, which contains the relevant process planning data to manufacture the designed part. Despite the system being able to remove the main barrier between CAD and CAPP, recognition of complex shapes remains a bottleneck, which requires a mature feature recognition system.

## 2.5 Artificial neural network

Sunil and Pande [4] developed an intelligent Artificial Neural Network (ANN) system to recognise prismatic features from B-Rep CAD models. In this approach, a unique 12-node vector scheme was created in order to classify machining features as families based on their geometrical and topological information. In this system, the input is a SAT file exported from Solidworks software, which is sent to a pre-processing module to construct a part face adjacency graph and to create a feature representation vector (FRV). All the FRVs are presented to the trained

ANN unit for feature classification. Finally, all the classified features from the ANN unit, with their relevant parameters, are post-processed to create a feature-based model. Hence, the feature-based model file can be linked to a CAPP system for CNC machining.

## 2.6 Hybrid AFR systems

A hybrid AFR method was developed by Rameshbabu and Shunmugam [16] to recognise the intersecting manufacturing features of a prismatic part using a STEP AP203 file. The AFR algorithm is a combination of face adjacency graph and volume subtraction techniques. This is to simplify a feature's graph by reducing the number of faces in a feature's volume and hence, the complexity of feature representation is minimised. The proposed system is also able to generate a part of CAPP in terms of determining a setup sequence to machine a component's features. Other aspects of CAPP, such as machine tool selection, operation selection, operation sequencing and machining sequencing, have not been considered, since it is assumed that only one machine is required to manufacture the product. Li et al. [17] also presented a hybrid AFR approach for the purpose of recognising intersecting features from a design part. In this system, three AFR techniques are used, which are hint based, AAG, and ANN. In the design part, the face loops are defined as generic feature hints, which can be extracted and used as clues for interacting features based on an enhanced attributed adjacency graph (EAAG) algorithm. Next, the relationships between the face loops are determined in order to create F-Loop Graphs (FLGs). The use of FLGs simplifies the graph representation of a part in the next step, which includes a final feature recognition using an ANN unit. Although the system can recognise complex intersecting features, it is restricted to planar faces and quadric surfaces. Tseng and Joshi [18] presented a machining volume generation method for recognising rotational and prismatic features of mill-turn parts. In order to generate feature volumes, the boundary faces of a part are swept along a direction determined based on the type of machining operation. The proposed method includes four main steps. Firstly, the part's faces are classified according to their geometric shapes and topological relationships as cylindrical, planar, or other types of faces. Then, several rotational machining zones, which represent cylindrical portions of the part that can be produced in turning processes, are determined. Rotational and prismatic machining volumes are generated using rotational sweeping operations for the former and volume decomposition, maximal volume sweeping, and reconstruction for the later. Finally, two approaches are used to recognise different types of features; these are 2D profile patterns for rotational features and 3D face adjacency relationship for prismatic features. Although many

types of rotational and prismatic features can be recognised following this method, it has several limitations. For example, a complex geometry can be recognised as rotational or prismatic features, which is unreliable in such cases and requires extra work. Liu et al. [19] utilised the concept of AAG to analyse the geometrical and topological relations of parts' machined surfaces. Following certain geometric reasoning rules, the AAG net is created by adding, deleting, and improving some linking relations between the graph nodes. Then, the resultant AAG is decomposed into a number of subgraphs in order to construct local rotational primitives. The 2D section of the turning volumes between the stock workpiece and the MTS model are meshed using the local-slicing approach. Also, a 2D shape descriptor for turning feature types was developed. Both the meshes and the descriptor helped creating a layer tree-based element sorting approach, which was established for identifying turning features.

However, beside the mentioned methods, there are many feature recognition techniques that serve different manufacturing processes in addition to those of machining. For example, Shi et al. [20] presented a feature recognition approach using heat kernel signature (HKS), for manufacturability analysis in Additive Manufacturing (AM). This includes an investigation regarding unique characteristics of AM processes, such as unsupported feature, minimum feature size, maximum vertical aspect ratio, minimum spacing, and minimum self-supporting angle. The proposed method has the ability to identify geometric features and manufacturing constrains of different shapes. Such technique can be justified to be compatible with the recognition of parts' features manufactured using machining processes.

Based on the literature review of AFR techniques, it is concluded that a particular emphasis has been placed on solving some issues, such as intersecting features. Also, features that contain toroidal surfaces have to be recognised, in the same way that features containing planar surfaces are currently identified. Whilst a few systems have been able to deal with these matters, other barriers are still unsolved. For example, any AFR system requires a library comprising a set of predefined features; however, there is none that can be expected to include all the possibilities of features, which could exist in a part's design. Furthermore, a new feature recognition system is needed, which can automatically recognise new types of features and add them to its database [21].

## 3 Extracting geometrical and topological data from STEP files

STEP is an International Standard (ISO 10303) that is computer readable, which represents and exchanges product

data during the entire product lifecycle, including: design, manufacturing, quality control, inspection and support [22]. The STEP standard is categorised as a series of nearly 2,000 parts, which have been published separately. Different Application Protocols (APs) have been developed in STEP to specify the representation of product data for one or more industrial applications [23], with AP 203 being widely known as a CAD model's data exchange file. The objective of AP 203 is to describe the product data in a way that satisfies industrial needs, in order to exchange configuration-controlled 3D product design data of mechanical parts or assemblies [24]. Most of these APs use the EXPRESS language to describe the representation of the data, which is transformed into a physical text file [22].

### 3.1 Structure of a STEP (AP-203) file

The STEP AP 203 file is a language-based text file, containing strings and entities describing a product's representation [25], being divided into two main sections: "HEADER" and "DATA". The HEADER section provides general product information, such as file name, creation date, company name, and programmer name. The DATA section contains lines of entities, which represent the main geometrical and topological product data. Each entity instance in the DATA section starts with "#" followed by a unique integer, entity name, and related data. The related data can be numbers, strings, Boolean, or even a reference to another entity instance in the file.

The geometrical and topological product data is organised into a hierarchical structure, as shown in Fig. 2. The top level of description in this hierarchy is the shell, which is a topological item that bounds a region in 3D space by joining a number of faces along edges. For example, the cylinder in Fig. 3a is considered a "CLOSED\_SHELL", with the STEP AP 203 file for it containing 253 entity lines in its DATA section. Whilst the STEP file has a hierarchical structure and shows the entity lines in an ordered form, the data arrangement does not appear in a hierarchical sequence in the physical text file. Hence, the "CLOSED\_SHELL" node, which represents the top of the hierarchy, does not necessarily exist in the first line of entities. In fact, it could be in any order between the beginning and the end of the DATA section. This statement holds true for all the other levels of data.

The faces represent the second level of description in a STEP AP 203 file. A face is a topological entity that describes a piece of a surface bounded by loops, which shares an edge with exactly one other face to form a CLOSED\_SHELL. Figure 3b shows the four faces of the cylinder and some other details, including the edges and vertices of the first face #47. Each face can be defined

using two pointers: bound and surface. The bound always takes the first pointer, whereas the second pointer denotes the surface of the face. The ADVANCED\_FACE entity ends with a Boolean flag, which indicates whether the loop direction is oriented in accordance with or opposed to the surface normal [26].

#### 3.1.1 Bound

A bound is a loop of edges utilised as face borders and each face might contain one or more bounds. This can be a "face outer bound" or a "face bound", which refers to external or internal face edge loops, respectively. Regardless of the type of bound, it always contains one "EDGE\_LOOP" indicator. An edge loop is a closed path of oriented-edges around a face, for which the start and the end points are the same. An "ORIENTED\_EDGE" is derived from another original edge and contains a Boolean flag that refers to the direction orientation. Accordingly, the orientation from the start to the end vertex can be reversed by this Boolean flag. Each oriented edge refers to a single "EDGE\_CURVE", which is a topological entity bounded by a start and an end point. Based on the edge type, each "EDGE\_CURVE" has a full geometrical and topological description. For example, a line is described by a start point vertex, end point vertex, vector, and one direction, whereas a circle is defined by a start point, end point, and centre point vertices as well as a radius and two directions. A Cartesian point is the lowest level in the STEP hierarchy, which defines a vertex in Cartesian space in terms of  $x$ ,  $y$ , and  $z$  axes.

#### 3.1.2 Surface

A surface is an indicator of the face type, which can take only one prospect, such as a plane, cylindrical, conical, and toroidal. The provided information in this section is different based on the surface type. For example, a cylindrical surface is described as "AXIS2\_PLACEMENT\_3D2", which defines a reference vertex on the surface and a direction, as well as the radius of the surface. Whereas, in the case of a toroidal surface, it is defined as AXIS2\_PLACEMENT\_3D2, major radius, and minor radius (for more details about the STEP AP203, see [24]).

### 3.2 Developing a parser for a STEP file

As mentioned above, the data in a STEP AP 203 file, which is represented by entity lines, does not appear in a specific logical order. Consequently, the file is almost unreadable since it requires moving from line to line in a non-sequential way. This is unreadable not only for humans, but also for feature recognition systems. It is therefore

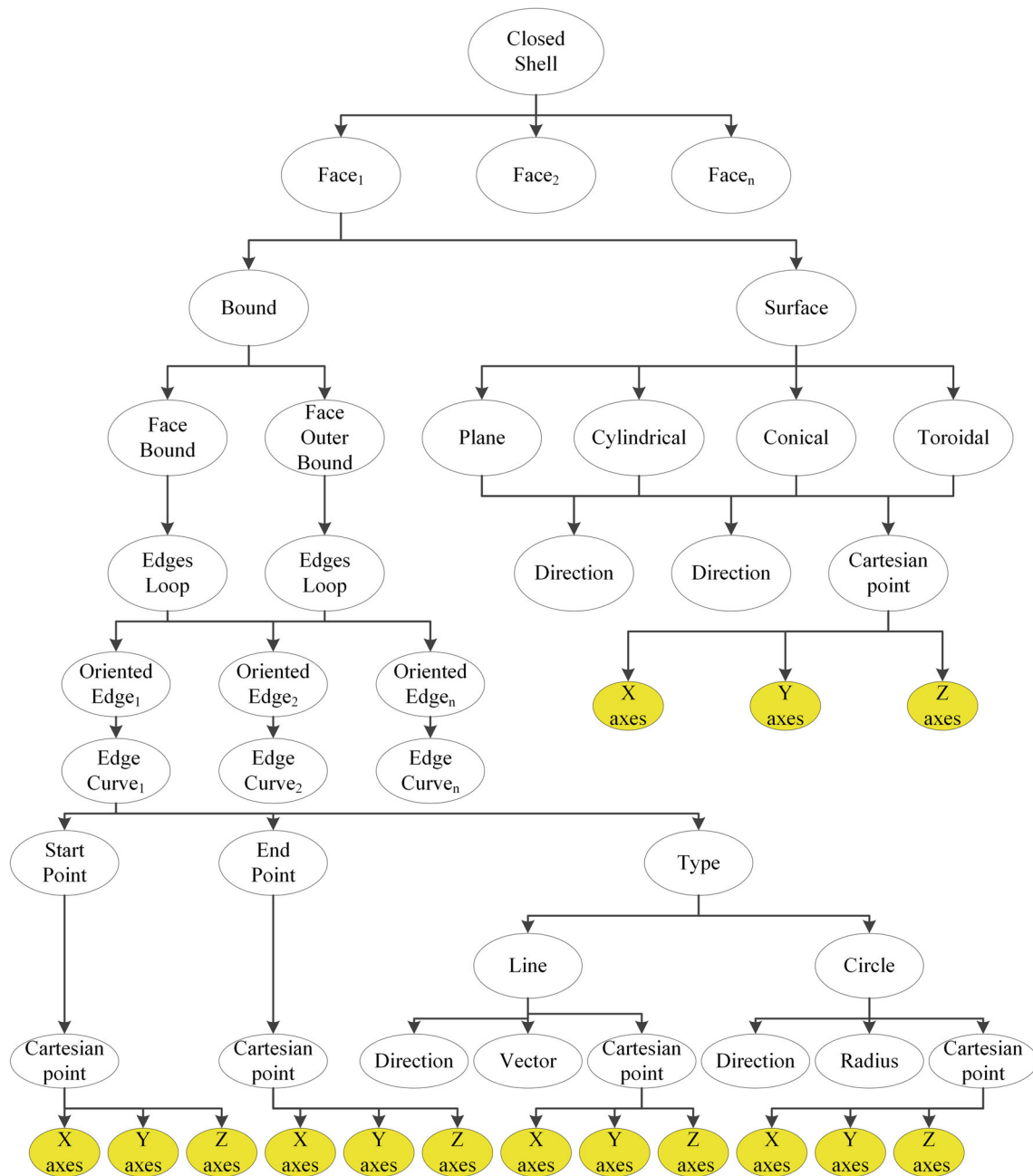


Fig. 2 The structure of a STEP AP 203 file

necessary to develop a parser so as to re-arrange the data in a comprehensible way. The parser that was developed in this work imports a STEP AP 203 file and scans all of its lines to find how many closed shells there are in the design as well as how many faces each contains. Then, each face is displayed with all of its geometrical and topological information before moving to the next face. When a face is declared, the first bound, which might be either face bound or face outer bound, will appear in the following line. Next,

the parser shows the edge-loop with indicators of the edges that form the face. All the edges are given with their related information until the lowest level of Cartesian points. If the face contains another bound, the parser displays it directly in the same way as the previous one, otherwise the surface type of the face is declared with all of its details before moving on to the next face. Figure 4a and b show part of the original and the resultant STEP file AP 203 after applying the parser, respectively. The parser is developed

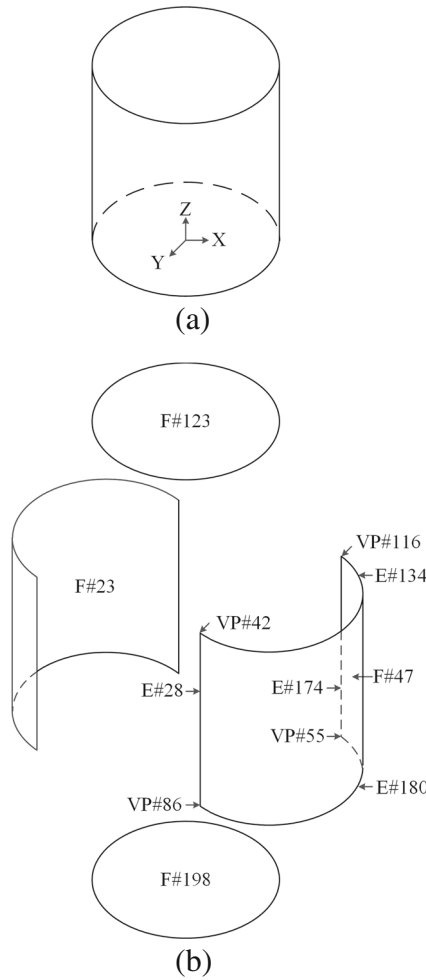


Fig. 3 a A Cylinder, and b four faces of the cylinder with some details of the first face

using the C# programming language, and it aims to facilitate the feature recognition task. However, more manipulating, editing, and filtering are needed before moving on to the feature recognition module. This will be achieved using developed algorithms that are explained in the next section in detail.

### 3.3 Algorithms to manipulate the STEP before moving on to feature recognition

Different algorithms have been proposed for this system to manipulate the results from the parser before moving on to the feature recognition module. These have the aim of facilitating the task of the feature recogniser, by delivering the information of the part's design in a more organised way.

#### 3.3.1 Find convexity or concavity of a toroidal face

A toroidal face is generated by rotating a curve around an axis in its own plane. Based on its information, such as start

```

DATA;
#1 = SHAPE_DEFINITION_REPRESENTATION ( #02, #221 ) ;
#2 = PLANE ( 'NONE', #102 ) ;
#3 = PLANE ( 'NONE', #247 ) ;
#4 = CARTESIAN_POINT ( 'NONE', ( 30.000000000000000000000000000000, 3.673040397442058400E-015, 45.000000000000000000000000000000 ) ) ;
#5 = DATE_TIME_ROLE ( 'classification_date' ) ;
#6 = FACE_OUTER_BOUND ( 'NONE', #59, .T. ) ;
#7 = DATE_AND_TIME ( #172, #216 ) ;
#8 = LINE ( 'NONE', #4, #189 ) ;
#9 = PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE ( 'ANY', '', #61, .NOT_KNOWN. ) ;
#10 = PERSON_AND_ORGANIZATION_ROLE ( 'creator' ) ;
#11 = APPLICATION_PROTOCOL_DEFINITION ( 'international standard', 'config_control_design', 1994, #244 ) ;
#12 = DIRECTION ( 'NONE', ( 0.00000000000000000000000000000000, 0.00000000000000000000000000000000, 1.000000000000000000000000000000 ) ) ;
#13 = CLOSED_SHELL ( 'NONE', ( #47, #23, #123, #198 ) ) ;
#14 = LOCAL_TIME ( 22, 52, 35.000000000000000000000000000000, #72 ) ;
#15 = MANIFOLD_SOLID_SHEP ( 'Boss-Extruded', #13 ) ;
#16 = DIRECTION ( 'NONE', ( 1.00000000000000000000000000000000, 0.00000000000000000000000000000000, 0.00000000000000000000000000000000 ) ) ;
#17 = APPROVAL_STATUS ( 'not_yet_approved' ) ;
#18 = PRODUCT ( 'Fig2', 'Fig2', ' ', ( #64 ) ) ;
#19 = COORDINATED_UNIVERSAL_TIME_OFFSET ( 0, 0, .AHEAD. ) ;
#20 = APPROVAL ( #228, 'UNSPECIFIED' ) ;
#21 = CYLINDRICAL_SURFACE ( 'NONE', #07, 30.000000000000000000000000000000 ) ;
#22 = CC_DESIGN_DATE_AND_TIME_ASSIGNMENT ( #131, #150, ( #165 ) ) ;
#23 = ADVANCED_FACE ( 'NONE', ( #6 ) , #39, .T. ) ;
#24 = EDGE_CURVE ( 'NONE', #55, #86, #213, .T. ) ;
#25 = APPROVAL_PERSON_ORGANIZATION ( #80, #52, #173 ) ;
#26 = CALENDAR_DATE ( 2017, 9, 11 ) ;
#27 = DATE_AND_TIME ( #245, #69 ) ;
#28 = EDGE_CURVE ( 'NONE', #42, #86, #8, .T. ) ;
#29 = CARTESIAN_POINT ( 'NONE', ( 0.00000000000000000000000000000000, 0.00000000000000000000000000000000, 45.000000000000000000000000000000 ) ) ;
#30 = APPROVAL_ROLE ( ; ) ;
#31 = COORDINATED_UNIVERSAL_TIME_OFFSET ( 0, 0, .AHEAD. ) ;
#32 = DATE_AND_TIME ( #85, #187 ) ;
#33 = EDGE_LOOP ( 'NONE', ( #238, #153 ) ) ;
#34 = APPROVAL_DATE_TIME ( #87, #211 ) ;
#35 = PERSON_AND_ORGANIZATION ( #44, #206 ) ;
#36 = CC_DESIGN_APPROVAL ( #20, ( #9 ) ) ;
#37 = APPROVAL_STATUS ( 'not_yet_approved' ) ;
#38 = PERSON_AND_ORGANIZATION_ROLE ( 'design_owner' ) ;
#39 = CYLINDRICAL_SURFACE ( 'NONE', #90, 30.000000000000000000000000000000 ) ;
#40 = PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE ( 'ANY', '', #18, .NOT_KNOWN. ) ;
#41 = DIRECTION ( 'NONE', ( 0.00000000000000000000000000000000, 0.00000000000000000000000000000000, 1.000000000000000000000000000000 ) ) ;
.
.
#253 = CIRCLE ( 'NONE', #83, 30.000000000000000000000000000000 ) ;
ENDSEC;
END-ISO-10303-21;

```

```

(a)
A Closed Shell Has been found
#13 = CLOSED_SHELL ( 'NONE', ( #47, #23, #123, #198 ) );

F1=#47
#47 ADVANCED_FACE #94 #21
#94 FACE_OUTER_BOUND #251
#251 EDGE_LOOP #190 #66 #71 #84
#190 ORIENTED_EDGE #134
#134 EDGE_CURVE #42 #116 #70
#42 VERTEX_POINT #110
#110 CARTESIAN_POINT X=30 Y=0 Z=45
#116 VERTEX_POINT #109
#109 CARTESIAN_POINT X=-30 Y=0 Z=45
#70 CIRCLE #69 Radius = 30
#69 AXIS2_PLACEMENT_3D #215 #12 #152
#215 CARTESIAN_POINT X=0 Y=0 Z=45
#12 DIRECTION X=0 Y=0 Z=1
#152 DIRECTION X=1 Y=0 Z=0
#66 ORIENTED_EDGE #28
#28 EDGE_CURVE #42 #86 #8
#42 VERTEX_POINT #110
#110 CARTESIAN_POINT X=30 Y=0 Z=45
#86 VERTEX_POINT #51
#51 CARTESIAN_POINT X=30 Y=0 Z=0
#8 LINE #4 #189
#4 CARTESIAN_POINT X=30 Y=0 Z=45
#189 VECTOR #218 1000
#71 ORIENTED_EDGE #180
#180 EDGE_CURVE #86 #55 #124
#86 VERTEX_POINT #51
#51 CARTESIAN_POINT X=30 Y=0 Z=0
#55 VERTEX_POINT #217
#217 CARTESIAN_POINT X=-30 Y=0 Z=0
#124 CIRCLE #46 Radius = 30
#46 AXIS2_PLACEMENT_3D #229 #101 #16
#229 CARTESIAN_POINT X=0 Y=0 Z=0
#101 DIRECTION X=0 Y=0 Z=1
#16 DIRECTION X=1 Y=0 Z=0
#84 ORIENTED_EDGE #174
#174 EDGE_CURVE #116 #55 #176
#116 VERTEX_POINT #109
#109 CARTESIAN_POINT X=-30 Y=0 Z=45
#55 VERTEX_POINT #217
#217 CARTESIAN_POINT X=-30 Y=0 Z=0
#176 LINE #108 #213
#108 CARTESIAN_POINT X=-30 Y=0 Z=45
#213 VECTOR #144 1000
#21 CYLINDRICAL_SURFACE #207 Radius = 30
#207 AXIS2_PLACEMENT_3D #135 #120 #145
#135 CARTESIAN_POINT X=0 Y=0 Z=45
#120 DIRECTION X=0 Y=0 Z=-1
#145 DIRECTION X=-1 Y=0 Z=0

F2=#23
#23 ADVANCED_FACE #6 #39
#6 FACE_OUTER_BOUND #59
#59 EDGE_CURVE #134 #107 #161 #724
(b)

```

Fig. 4 a Part of the original STEP file, and b part of the resultant STEP file after applying the parser

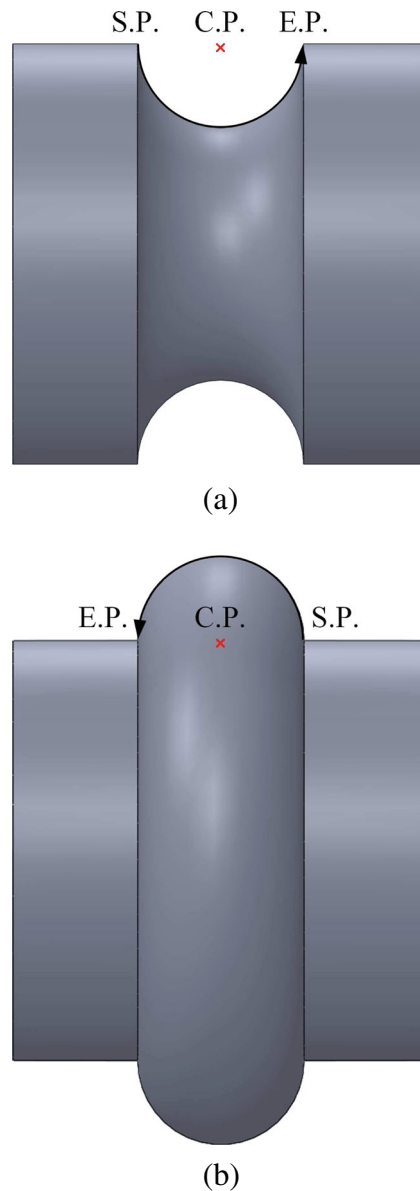
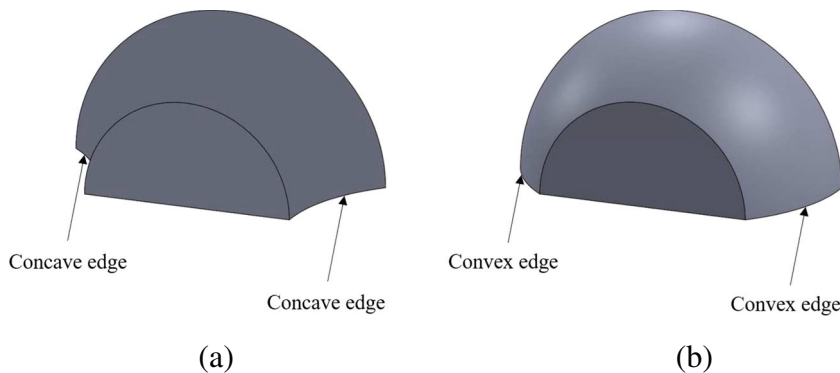


point, end point, centre, and the direction of rotation, the toroidal face might form a convex or concave shape in the part. The STEP file provides such information, but does not specify whether the toroidal face is a concave or a convex one. This is something that should be determined before moving on to the feature recognition process due to its effect in recognising features that contain toroidal faces. Below, is the information required and steps that need to be taken to distinguish the convexity or concavity of a toroidal face.

- A toroidal face has four “ORIENTED\_EDGE”, only two of which can inform whether it is concave or convex. This can be seen in Fig. 5a and b, which show concave and convex toroidal faces, respectively.
- The proposed system starts with scanning the parser and reading all the “ORIENTED\_EDGE” entities that form the toroidal face in order to check their Boolean flags. If the “ORIENTED\_EDGE” line ends with the letter “.T.”, this is an indication that the start and end points of the curve are in their natural orientation. In contrast, the letter “.F.” refers that the start and the end points are in reversed positions.
- The centre point information of the two related edges are extracted since it is also required in this algorithm.
- The importance of knowing the right place of the curve’s start, end, and centre points relates to the fact that the direction of the rotation from the start to the end point is always *counter clockwise*. Figure 6a and b show the effect of swapping positions between the start and the end points when forming concave or convex shapes.
- Finally, the algorithm takes into consideration the first direction of the two edges that form the concavity and convexity of the toroidal face. This is to distinguish between the inward-pointing and outer-pointing normal to the edge’s plane, since those of different edges in the same toroidal face might take different directions.

By providing the required information to the developed algorithm in this work, the system is able to recognise the convexity or concavity of each toroidal face. Following this methodology overcomes the issues raised by previous research. For example, in [27], the authors assumed that

**Fig. 5** **a** Concave toroidal surface, and **b** convex toroidal surface



**Fig. 6** **a** Concave shape, and **b** convex shape

“If EDGE\_CURVE last letters construction is T, F, T, F, then it is a concave external toroidal feature”, and “If EDGE\_CURVE is constructed with four circles and T, T, T,

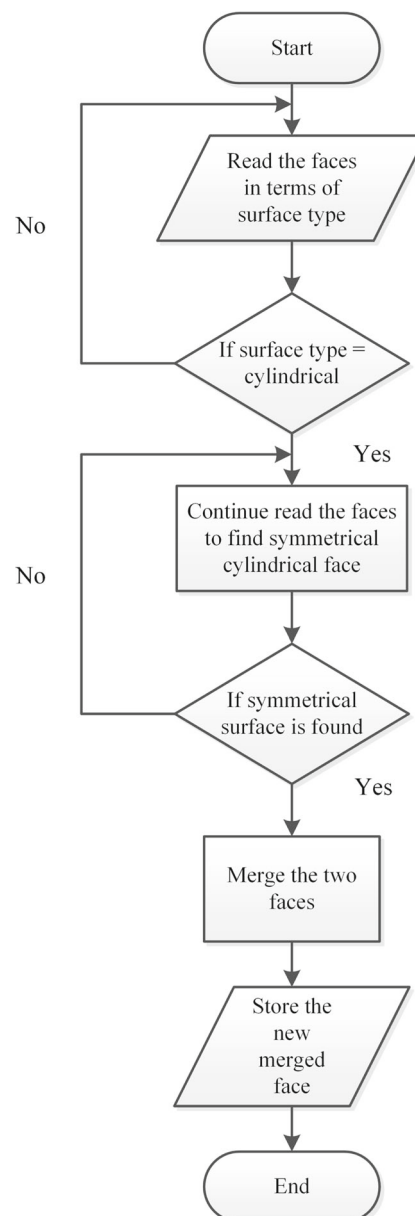
T, then it is a convex external toroidal feature”. In fact, the Boolean flags do not relate to the convexity or concavity of a toroidal face, as has been explained above. In another example, namely [28], the authors restricted their AFR system by assuming: “If the step data contains a toroidal surface with a minor radius at the left end, then the feature is identified as convex contour turning”, and “If the step data contains a toroidal surface with a major radius at the left side, then the feature is identified as concave contour turning”. However, this is impracticable, because a concave or convex feature can be on the left and/or the right in the part design.

### 3.3.2 Merging symmetrical faces

Based on the nature of the STEP file, special consideration is given to symmetrical rotational parts that have cylindrical, conical, or toroidal faces. Whilst each of these faces appears as a single face in any CAD model, they are interpreted and saved as two separated ones in the structure of the STEP file. For example, the cylinder in Fig. 3a has three faces, which are two plane circles on the sides and one cylindrical surface links them. However, the STEP file saves the cylinder as four faces of two plane circles and two halves of the cylindrical surface. Merging symmetrical faces information (MSF) algorithms have been developed to combine the information of any *two halves* of faces that are symmetrical. Figure 7 shows one that merges the information of two symmetrical cylindrical faces, and saves them as single one.

### 3.3.3 Merging adjacent toroidal faces

Whilst all the types of symmetrical faces, including toroidal faces, are declared as two separated faces in the STEP file, some of the latter have special representations based on their angular position and length. This representation is derived from the angular division concept of 2D space, which is divided into four partitions by two perpendicular axes. For instance, if the curve of the minor-radius that constructs the toroidal face is located in one angular partition, the face is represented as two symmetrical separated faces, which is the same case for cylindrical and conical faces. Figure 8a, b, and c show an example of this case. However, a different representation of a toroidal face is declared in a STEP file, if the curve is concave and located in more than one angular partition. For example, the curve of the toroidal face in Fig. 9a is located in two angular partitions and hence, it is saved in the STEP file as four separated faces, as shown in Fig. 9b. The first step includes the use of the MSF algorithm, in order to merge the information of each two symmetrical toroidal faces, as in Fig. 9c. Despite the implementation of the MSF algorithm helping to combine the information of symmetrical faces, the result is still insufficient in this

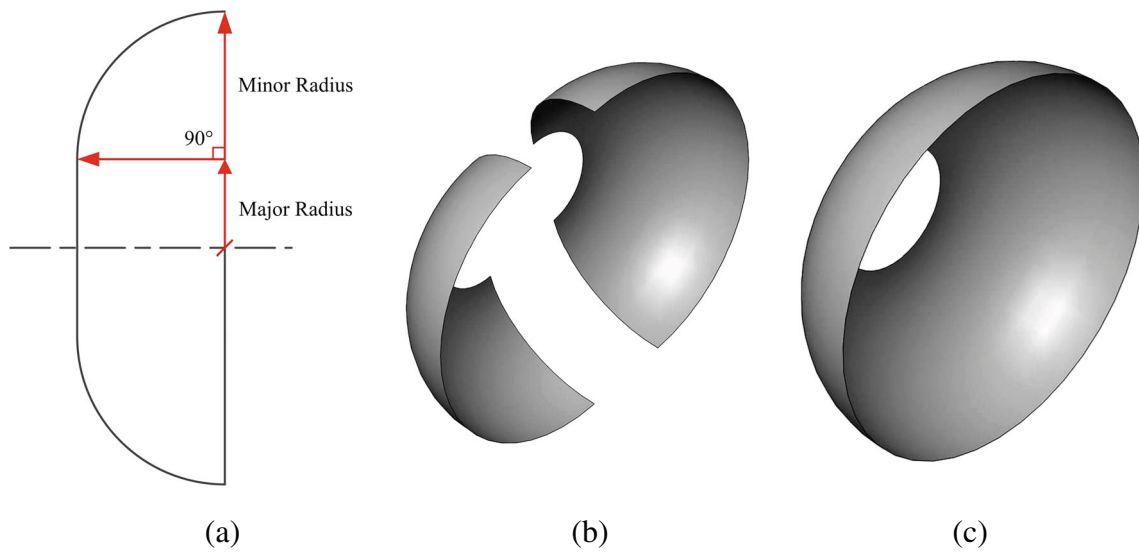


**Fig. 7** An algorithm for merging the information of two symmetrical cylindrical faces

case, because the toroidal face is still saved as two parts. Hence, another algorithm, which is shown in Fig. 10, has been developed for the purpose of merging the adjacent toroidal faces (MATF) information. After applying both the MSF and MATF algorithms, the toroidal face is saved as one (Fig. 9d), to be used later in the feature recognition system.

### 3.3.4 Sorting faces

A three-dimensional model is a part representation connecting a number of faces along various types of edges, such as lines and arcs. Regardless of the number and the type of a

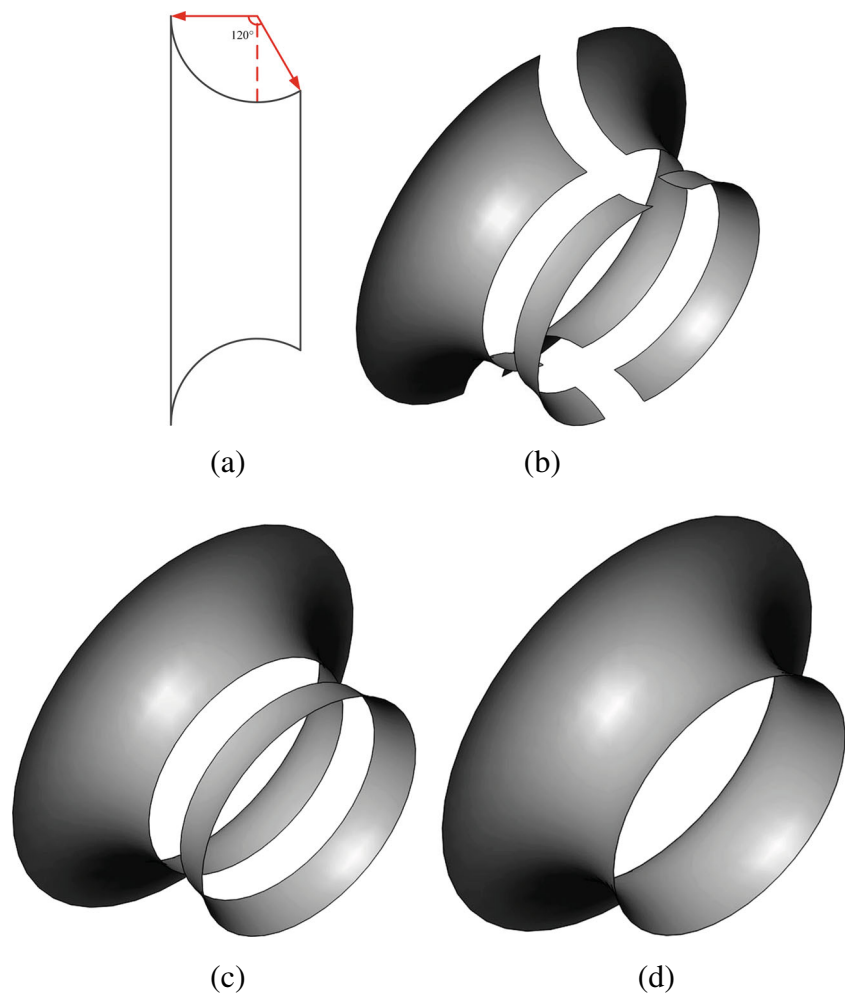


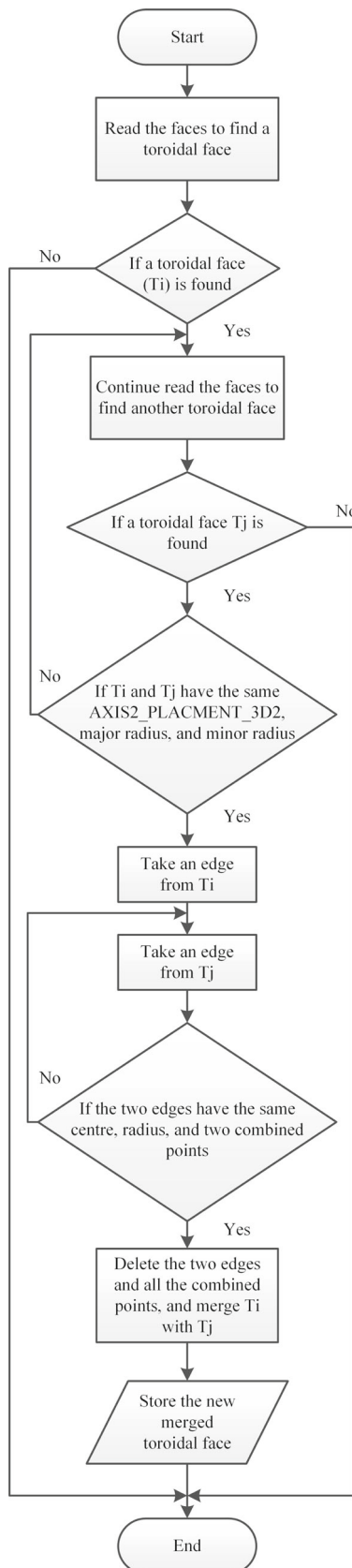
**Fig. 8** **a** 2D view of a toroidal face, **b** a declaration of the toroidal face in a STEP file, and **c** after merging the two symmetrical toroidal faces

3D model’s faces, the STEP file saves and transfers the geometrical and topological information of each face. However, these faces do not follow any type of order in the STEP file,

which makes them hard to be traced by a reader or a feature analyser. Thus, a faces sorting algorithm has been developed in order not only to facilitate the task of the feature

**Fig. 9** **a** 2D view of a toroidal face, **b** declaration of the toroidal face in a STEP file, **c** face description after merging each two symmetrical toroidal faces, and **d** face description after merging two adjacent ones





**Fig. 10** An algorithm for merging the information of two adjacent toroidal faces

recogniser but also to help in finding holes and internal features in the part, as explained in the next section.

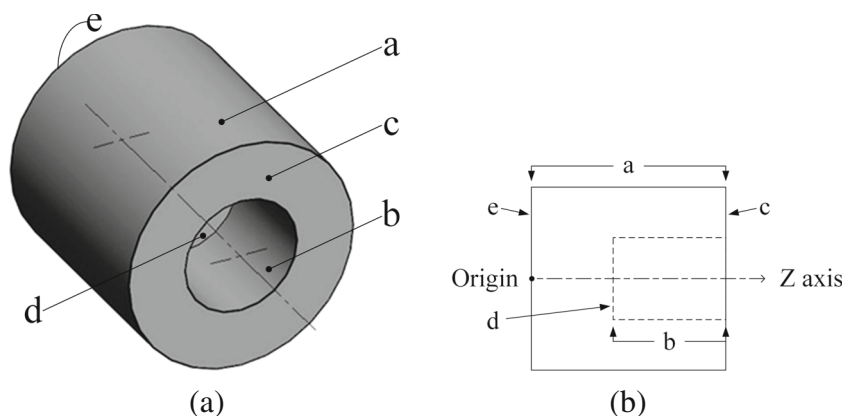
The coordinate system in turning machines is  $XZ$ , where  $X$  represents the diameter and  $Z$  represents the axis of rotation or length. This is the same system adopted to develop the sorting algorithm in this work since it is created for rotational parts. The sorting criterion is selected based on Cartesian  $Z$  values of each face. Each 3D face has maximum and minimum  $Z$  values, which are obtained from its edge start and end points, denoted  $MaxZ$  and  $MinZ$ , respectively. Regarding planes, which are 2D faces, their  $MaxZ$  and  $MinZ$  are equal. The sorting process of faces depends on their  $MaxZ$  values, which are taken in descending order. The  $MinZ$  of a face is used for comparison only if two or more faces have the same  $MaxZ$ , then the priority is given to the face that has the highest  $MinZ$ . Figure 11 shows a cylinder with a blind hole and an axis of rotation about the  $Z$  direction. After applying the MSF algorithm, five faces are declared in this cylinder, which are (a) outer cylindrical surface, (b) inner cylindrical surface of the hole, (c) plane ring surface, (d) circler plane at the bottom of the hole, and (e) circler plane at the cylinder base. By assuming that the base locates in the origin  $Z$  zero, three faces, a, b, and c, share the same  $MaxZ$  value. The face (c) takes the first place in the sorting node since it has the highest  $MinZ$  value when compared with the faces (a) and (b). By following the same arrangement, face (b) has priority over face (a), because its  $MinZ$  value is higher than that of face (a). Given face (d) has a higher  $MaxZ$  value than face (e), they take the fourth and the last places, respectively. The final sorting node for

1. Plane ring surface (c);
2. Inner cylindrical surface of the hole (b);
3. Outer cylindrical surface (a);
4. Circler plane at the bottom of the hole (d);
5. Circler plane at the cylinder base (e).

### 3.3.5 Split external features from internal features and holes

Although the STEP file is one of the most significant PDE formats in terms of saving and transferring the design information, this does not change the fact that the imparted information is of a low level. Designing models and consequently the STEP file, describes a part as geometrical and topological information, such as faces, surfaces, edges, lines and curves, and points. Thus, holes and general internal faces are not explicitly distinguished unless the provided information is analysed. Taking an example of the cylinder in Fig. 12, its original STEP file information does not clearly inform that this cylinder has holes. Hence, algorithms have been developed in order to separate the faces that form the external shell and the faces that belong to holes and

**Fig. 11** **a** A cylinder with a blind hole and **b** a side view of the cylinder



other internal features. This is achieved in three steps as follows:

**Step 1: finding the indicators of holes**

The first algorithm receives the sorted faces node of the previous step and scans all the faces to find which share the same MaxZ value. If there are three or more faces that have the same MaxZ, they are grouped together to

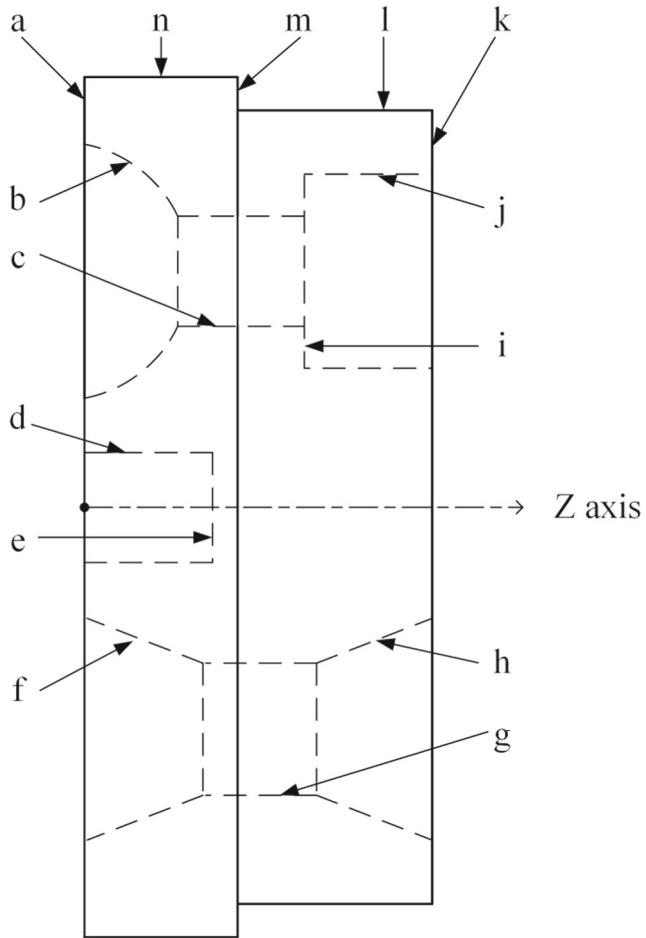
be analysed. In this example, one group of faces is found, which includes faces *k*, *j*, *h*, and *l*. Analysis of this group reveals the following: (1) there are three 3D faces, which are *j*, *h*, and *l*, and one plane face *k*, having the same MaxZ value; (2) faces *k* and *l* share the same “face outer bound”; (3) faces *j* and *h* share a “face outer bound” with a “face bound” of face *k*; and (4) based on face *k*, the three faces *j*, *h*, and *l* have the same direction. Such an analysis leads the system logically to select the cylindrical face *j* and the conical face *h*, flagging them as indicators of either holes or internal features.

**Step 2: find and isolate related faces**

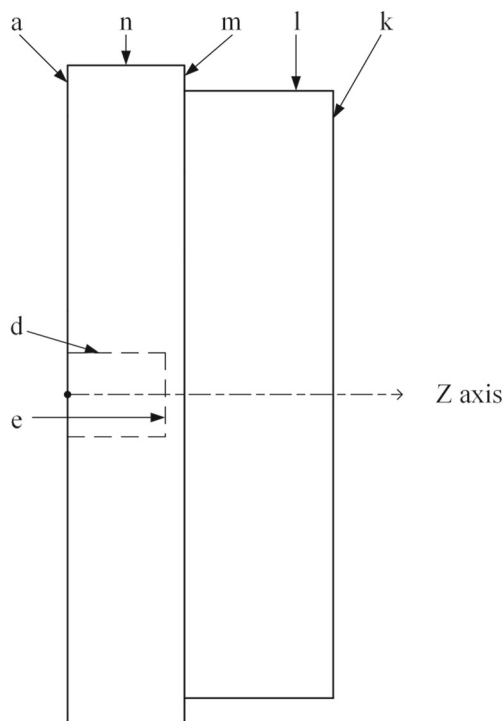
After flagging the indicators, each indicator is isolated and used for establishing an individual internal node of faces. The second algorithm examines the remaining faces to find which of them is connected to the indicator face. This is achieved by taking the edge data of the indicator face and scanning the edges of all the other faces. If a match is found, that face is evaluated as an internal face and moved to the same node of the indicator. The same action is repeated again by holding the edges of the new added face and matching them with the edges of the other faces. For example, the cylindrical face *j*, which is classified as a hole’s indicator, has only one combined edge with the plan face *i*. The face *i* is moved to join the indicator *j* in its node. Then, the action recurs, with this time, the matching being found between the “face bound” of the plane *i* and the “face outer bound” of the cylindrical face *c*. The scanning process is continued under specific matching conditions until all the related faces are found. In this example, the result of the second step is two nodes of internal faces (*j*, *i*, *c*, and *b*) and (*h*, *g*, and *f*), which are virtually subtracted from the original whole shape.

**Step 3: checking for other possibilities**

After the virtual subtracting of the previous internal nodes has been completed, the remaining shape appears as in Fig. 13. It takes a mere glance to distinguish that there is



**Fig. 12** A part design includes 14 external and internal faces



**Fig. 13** The remainder of the part after the virtually subtraction of two nodes of internal faces

another hole in the shape, which requires being recognised and isolated in an individual node, exactly like the preceding two. This is started by repeating step 1 with one adjustment, whereby instead of scanning the faces based on their MaxZ value, the algorithm uses the faces' MinZ. The faces that share the same MinZ are *n*, *a*, and *d*. Next, the system repeats the whole procedure of step 2, which results in a node of internal faces *e* and *d*. The final result of this example is as follows:

- External faces node (*k*, *l*, *m*, *n*, and *a*);
- Internal faces node (*j*, *i*, *c*, and *b*);
- Internal faces node (*h*, *g*, and *f*);
- Internal faces node (*e* and *d*).

The task of isolating external shell faces from internal faces and grouping the latter, should predate the feature recognition step. This is because the rules for recognising external features, holes, and internal features are different. Also, this method contributes to the elimination of the difficulties, as mentioned regarding the previous systems, in terms of recognising different types of holes and complex internal features. For example, in [9] the authors mentioned “it is difficult to recognize axial holes in rotational components” and in fact, their system is only able to identify simple blind holes. Also, in [27], the authors assumed that “external features are recognized from right to left and internal features are recognized from left to right” by default

in the STEP file. It has been proven by designing different parts with holes that this assumption is not always correct.

Although the proposed method is able to split external features from internal ones, it still has limitations. For example, if the design model in Fig. 12 has only one internal shape (*j*, *i*, *c*, and *b*), and face (*k*) is a sphere or conical surface, this means only two faces (*k* and *j*) share the same MaxZ. In such a case, the method is not valid since it requires at least three faces sharing the same MaxZ.

## 4 Smart interactive automatic feature recognition (SI-AFR)

The proposed system for AFR is divided into two sections: recognition of predefined features and interactive feature recognition. The first section is aimed at covering most of the prevalent external and internal turning features by defining and saving them in a database. These predefined features are called up to be matched with a part design for the purpose of recognising the part's features. However, if a new and unexpected feature is declared in the part design, the interactive feature recognition section is activated, which means that this feature is recognised and added to the original predefined feature database. Both sections are based on the evaluation of the extracted geometrical and topological information obtained from the parser and the subsequent algorithm application. A part's data, such as the centre point coordinates, radii, line lengths, and types of surfaces, are considered geometrical information, whereas topological information concerns edge loops, inner and outer bounds, edge curve construction, and so on. Also, both of the sections are developed to recognise external and internal features.

### 4.1 Recognition of predefined features

By using different algorithms to manipulate the results from the STEP file parser, the data describing a part are organised as external and internal faces nodes. Each of these nodes is intended to be used in the recognition phase of predefined features. The proposed AFR system adopts a rule-based approach, whereby it recognises a feature by scanning the faces in each node and matching them with the predefined features based on certain rules that are characteristic to that feature. The details about scanning faces and recognising features are as follows. The SI-AFR system's database contains varying types of features in terms of the number of faces that form one, and depending on the geometrical and topological description, a feature might include from one to five faces. Thus, the predefined feature set is categorised into five groups, with those in each group having the same number of faces. Regardless of the number of faces that

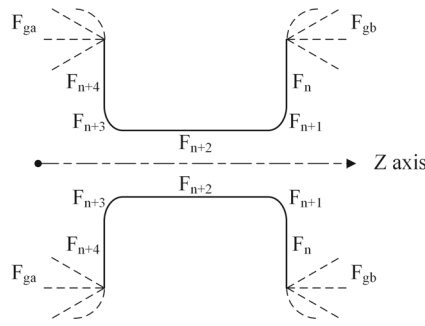


Fig. 14 A square groove with two round corners in the base

form the feature, each of these predefined ones has a unique description, which is saved in the database, with two extra faces denoted as  $F_{gb}$  and  $F_{ga}$  representing guides before and after a feature, respectively. For example, a square groove feature with two round corners in the base is shown in Fig. 14 and saved in the database, as in Table 1.

The SI-AFR system works as loops. Backing to the sorted external and internal face nodes, the first loop starts by holding a bundle of the first seven faces from a node and matches them with each predefined feature that includes *five related faces*. If a match is found:

- the loop is eliminated;

- five faces (from  $F_n$  to  $F_{n+4}$ ) of the bundle are declared as a feature;
- all the essential information is extracted from this feature’s entities;
- the next loop is started.

On the other hand, if no matching is confirmed, the first loop is continued, and the bundle is updated by keeping the first six faces and subtracting the last one. Then, the system compares the amended bundle with all the predefined features that include *four related faces*. This process continues, with the same decisions being taken, until all the faces in a node are covered. However, any loop is stopped, if the faces of a bundle reaches three items  $F_{gb}+F_n+F_{ga}$  and no matching is found. In this case, the face  $F_n$  is declared as undefined, and the second part of the system, which is the interactive feature recognition, is activated. Figure 15 shows a map of the predefined feature recognition loops.

One of the contributions with this methodology is the reduction of the stages required to recognise some multi-face features. This can be explained by taking different types of square groove features. The previous AFR systems, which were developed for rotational parts, can recognise blind square grooves, as in Fig. 16. This blind square groove consists of three faces, these being: a cylindrical

Table 1 Data of a square groove with two round corners in the base

Feature name	Square groove with two round corners in the base	
No. of related faces	Five faces= $F_n + F_{n+1} + F_{n+2} + F_{n+3} + F_{n+4}$	
No. of total faces	Seven faces= $F_{gb} + F_n + F_{n+1} + F_{n+2} + F_{n+3} + F_{n+4} + F_{ga}$	
Faces description	$F_{gb}$	Guide face before the feature Cylindrical, Conical, or Toroidal Surface.
	$F_n$	Plane surface Has face outer bound and face bound.
	$F_{n+1}$	Toroidal surface Concave
	$F_{n+2}$	Cylindrical surface
	$F_{n+3}$	Toroidal surface Concave
	$F_{n+4}$	Plane surface Has face outer bound and face bound.
	$F_{ga}$	Guide face after the feature Cylindrical, Conical, or Toroidal Surface.
Extracting data	$F_n$	Linear, start point, and end point.
	$F_{n+1}$	Circler, CW, toroidal centre, toroidal radius, start point, and end point.
	$F_{n+2}$	Linear, start point, and end point.
	$F_{n+3}$	Circler, CW, toroidal centre, toroidal radius, start point, and end point.
	$F_{n+4}$	Linear, start point, and end point.
	Max width	$F_n (\text{MaxZ}) - F_{n+4} (\text{MaxZ})$
	Max depth	$(F_n (\text{MaxX}) \text{ or } F_{n+4} (\text{MaxX})) - F_{n+2} (\text{MaxX})$

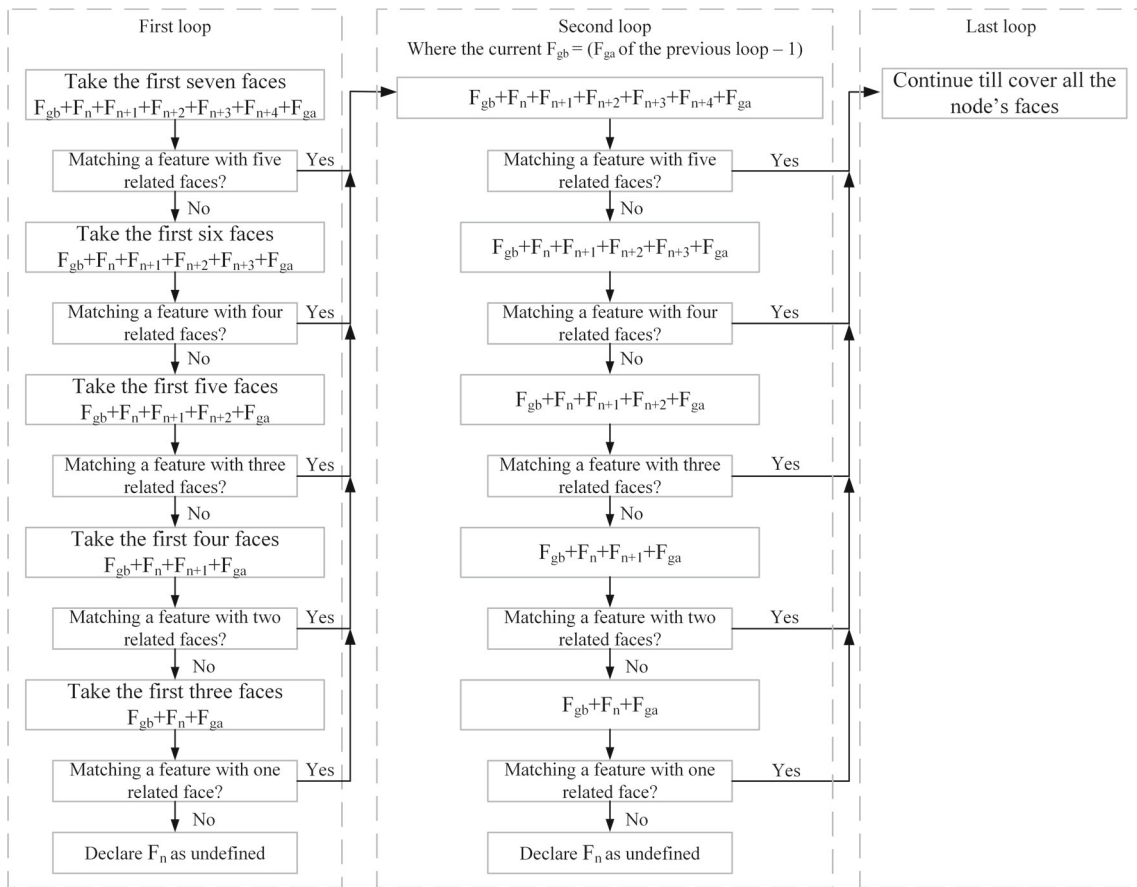


Fig. 15 A map of the predefined feature recognition loops

surface  $F_{n+1}$  in the middle and two bigger surfaces  $F_n$  and  $F_{n+2}$  on the sides, which might be cylindrical, conical, or toroidal. However, most of these AFR systems have not been developed to recognise a square groove with extra components and hence, the square groove with extra rounded corners in the base of Fig. 14 is declared as five

separated faces  $F_n+F_{n+1}+F_{n+2}+F_n+3+F_{n+4}$ . This is an unreliable solution and consequently, such systems require extra processing in order to recognise this type of feature. The proposed SI-AFR system has been developed to expect such features and thus, avoids the need for extra processing.

### 4.2 Interactive feature recognition

Regardless of the number of faces that form a feature, the following information of each predefined feature is saved in the system's database: (i) the number of faces, (ii) the type of each face, (iii) the relationship between faces, and (iv) the surface type of the two faces before and after the feature. When the loops of recognising predefined feature section are finished, one or more adjacent faces might be declared as an undefined feature. This has one meaning, *the system has diagnosed a new feature not included in the system's database*. The SI-AFR system has been prepared for such a situation, and the new feature can be added to the set of predefined ones through a smart and interactive procedure as is explained in the next sections.

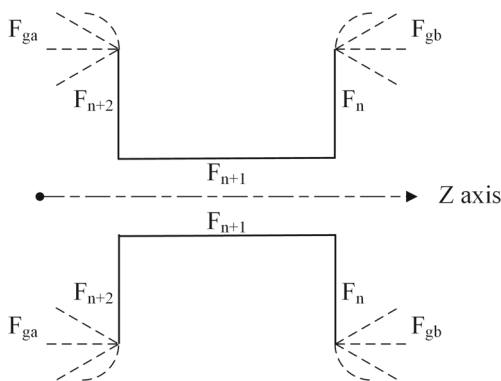


Fig. 16 A blind square groove



### 4.2.1 Extracting the new feature information

As mentioned, the interactive feature recognition section is activated if the SI-AFR system scans the model design of a part and cannot find a match between a feature’s information and any of those predefined in the database. Then, this is considered a new feature and analysed in order to extract the following data automatically:

- The number of faces that form the feature;
- The geometrical information of each face in the feature, such as the start and end point values of linear faces, as well as centre point and radius in the case of toroidal faces;
- The topological information of each face in the feature based on the face type. This information can include the type of the surface, the number and the type of bounds, the concavity and convexity in the case of a toroidal face, and a taper direction (left or right);
- The surface type of  $F_{gb}$  and  $F_{ga}$ .

The extracted data of the new feature is exactly the same as required for defining and adding each of the initial fifty-four features in the database. However, such data cannot be used directly in adding the new feature to the database; hence, it should be processed and saved in a way that allows the system to recognise similar features in the future. Figure 17 and Table 2 show an example of the upper profile of a new feature and how its data are extracted, respectively.

### 4.2.2 Processing the new feature information

As mentioned, the geometrical information of each face in the new feature is extracted as values. However, it is very important to note that this data should not be saved as actual numbers, but rather, as relationships between each face vertices, and between each two adjacent faces as well. The reason for this is because, if the new feature’s data

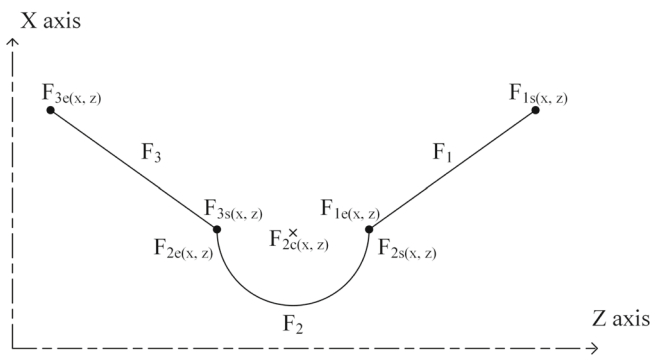


Fig. 17 An example of a new feature

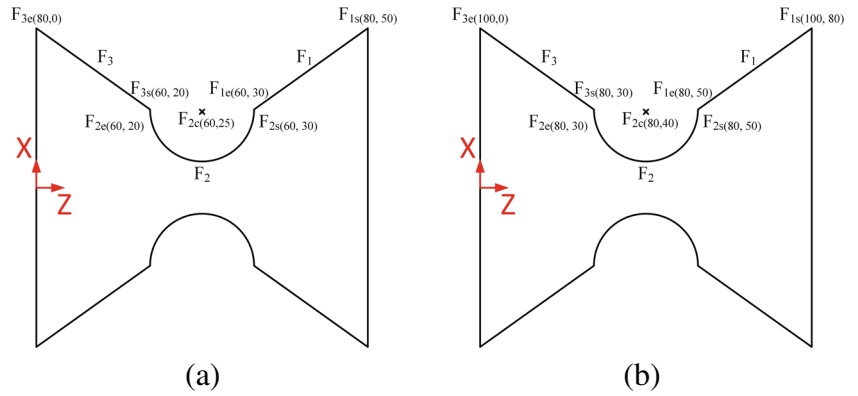
Table 2 Extracting a new feature’s data

Feature name:	“As the user suggested”
Number of related faces	Three
$F_1$	Type of surface: conical (left). $F_{1sz}$ = value. $F_{1sx}$ = value. $F_{1ez}$ = value. $F_{1ex}$ = value. $F_{1s}$ -to- $F_{1e}$ : linear.
$F_2$	Type of surface: toroidal (concave). $F_{2sz}$ = value. $F_{2sx}$ = value. $F_{2ez}$ = value. $F_{2ex}$ = value. $F_{2cz}$ = value. $F_{2cx}$ = value. Major radius. Manor radius. $F_{2s}$ -to- $F_{2e}$ : circler. Direction: CW.
$F_3$	Type of surface: conical (right). $F_{3sz}$ = value. $F_{3sx}$ = value. $F_{3ez}$ = value. $F_{3ex}$ = value. $F_{3s}$ -to- $F_{3e}$ : linear.
$F_{gb}$ and $F_{ga}$	Type of surface: “As they appear in the design”.

are saved as numbers, it will be restricted to these values, and thus, will not be recognised in new parts, if a change occurs in any dimension. For example, Fig. 18a shows a new feature with all the values of the start and end points of each face. Storing the geometrical information of this feature as numbers means the system will not recognise the feature in Fig. 18b because it has different values. Therefore, the feature’s information in this example (Fig. 18a) should be saved as follows:

1. Face1: left taper (linear);
2. Fcae2: concave surface (circler cw);
3. Face1-to-Face2:
  - $F_{1sz} > F_{2sz}$ ;
  - $F_{1sx} > F_{2sx}$ ;
4. Face3: right taper (linear);
5. Face2-to-Face3:
  - $F_{2sz} > F_{3sz}$ ;
  - $F_{2sx} = F_{3sx}$ ;

**Fig. 18** **a** A new feature with specific geometrical values and **b** a feature with same characteristics and different values



6. Face3 start-to-end:

- $F_{3sz} > F_{3ez}$ ;
- $F_{3sx} < F_{3ex}$ ;

7. Face<sub>gb</sub>: (As it appears in the design);

8. Face<sub>ga</sub>: (As it appears in the design).

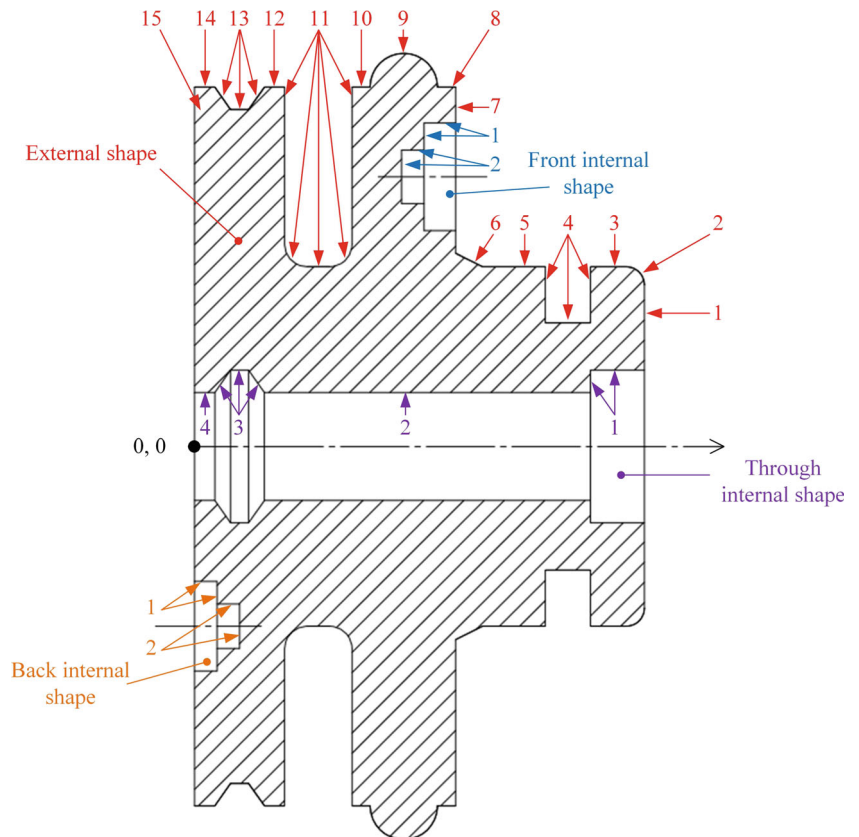
Using the relationships method to save a feature’s data makes the new feature more flexible to be recognised, if it is found in new parts.

#### 4.2.3 Saving the new feature information

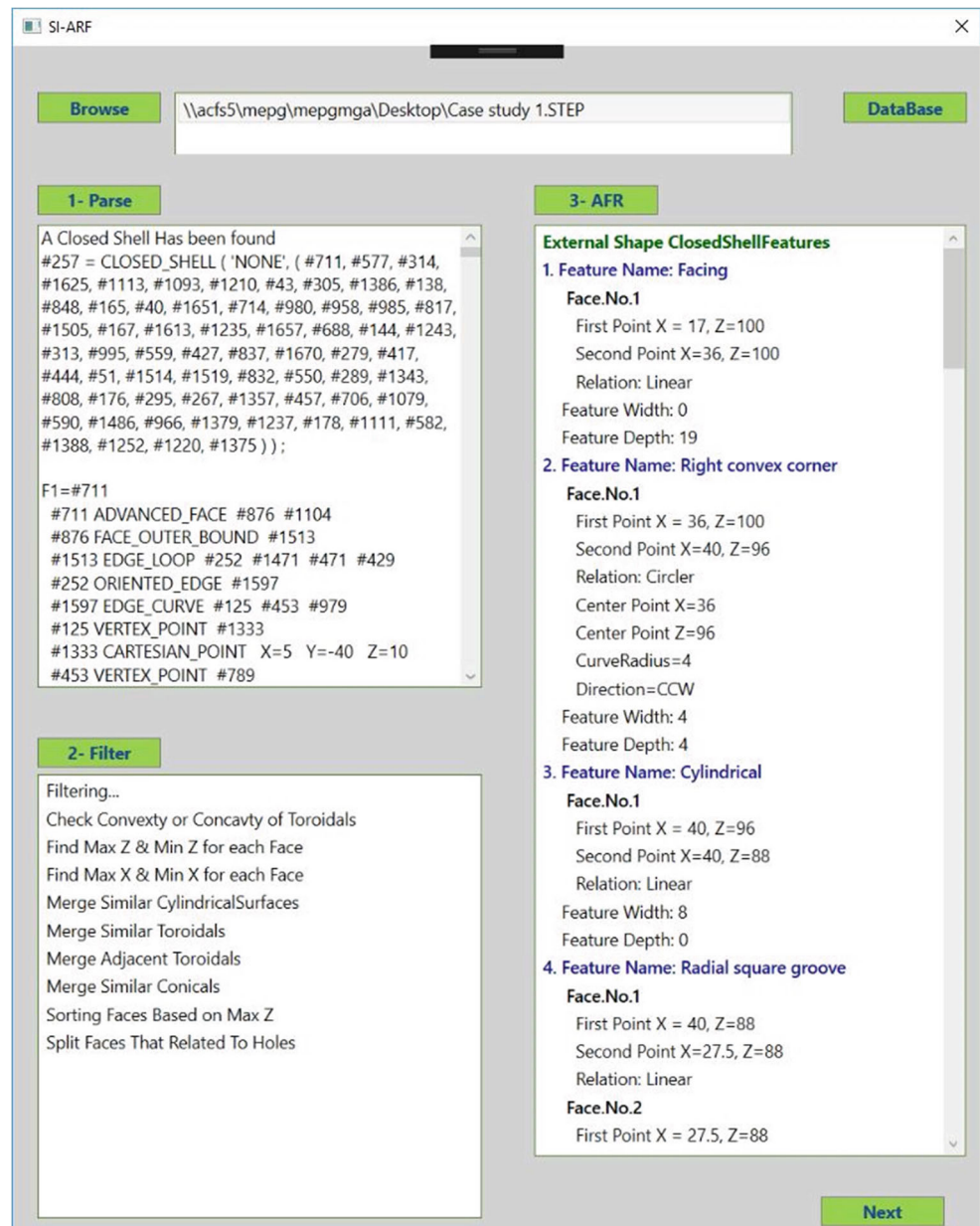
After extracting all the essential information automatically, the system asks the user to name the new feature. Next, the new feature with its suggested name and the extracted information are added to the database of predefined features. Since the database has five groups of predefined features, it will be added to a group based on its number of related faces.

The original number of predefined features in the SI-AFR system database is 54. As mentioned, this number can be

**Fig. 19** Top section of case study 1



**Fig. 20** The system's main window after applying the three steps for case study 1



increased using the interactive feature recognition section. However, the database is not just modifiable in terms of adding features, for it is also possible to delete or edit them. In fact, deleting and editing features can be undertaken via a friendly user interface without any need to log in and change the underlying code. The proposed interactive methodology contributes to eliminate the obstruction of recognising certain types of predefined features. This is achieved by analysing new features and adding them dynamically to the system's database via the user interface. Also, the added features are saved based on the entities' relationships method, which improves the flexibility of the system in recognising similar, "not necessary identical", features.

## 5 Case study

In order to evaluate the proposed system ability, two case studies have been tested. The two models examined do not exist in industry, but rather, have been designed for the purpose of validation and showing the high efficiency of the proposed system in recognising both predefined and new features.

### 5.1 Case study 1

A part has been designed using SOLIDWORKS and saved as a STEP AP 203. This model has only predefined features, which form the closed-shell comprising an external shape

**Table 3** The details of the recognized external features

External shape												
Feature No.	Feature name	Face No.	Xsp	Zsp	Xep	Zep	Start-to-end	Xcc	CR	Dir.	W	D
1	Facing	1	17	100	36	100	Linear				0	19
2	Left convex corner	1	36	100	40	96	Circler	36	4	CCW	4	4
3	Cylindrical	1	40	96	40	88	Linear				8	0
4	Radial square groove	1	40	88	27.5	88	Linear				10	12.5
		2	27.5	88	27.5	78	Linear					
		3	27.5	78	40	78	Linear					
5	Cylindrical	1	40	78	40	64	Linear				14	0
6	Right taper	1	40	64	43	58	Linear				6	3
7	Facing	1	43	58	80	58	Linear				0	37
8	Cylindrical	1	80	58	80	54	Linear				4	0
9	Convex	1	80	54	80	39	Circler	80	7.5	CCW	15	7.5
10	Cylindrical	1	80	39	80	35	Linear				4	0
11	Radial groove two rounded corners	1	80	35	45	35	Linear				15	40
		2	45	35	40	30	Circler	45	5	CW		
		3	40	30	40	25	Linear					
		4	40	25	45	20	Circler	45	5	CW		
		5	45	20	80	20	Linear					
12	Cylindrical	1	80	20	80	15.5	Linear				4.5	0
13	Two sides tapered groove	1	80	15.5	75	12	Linear				11	5
		2	75	12	75	8	Linear					
		3	75	8	80	4.5	Linear					
14	Cylindrical	1	80	4.5	80	0	Linear				4.5	0
15	Facing	1	80	0	12	0	Linear				0	68

**Table 4** The details of the three internal shapes

Internal shapes													
Feature No.	Feature name	Face No.	X <sub>c</sub>	Y <sub>c</sub>	Z <sub>cs</sub>	RZ <sub>cs</sub>	Z <sub>ce</sub>	RZ <sub>ce</sub>	Start-to-end	X <sub>cc</sub>	CR	Direction	
Through internal shape													
1	Right axial groove	1	0	0	100	17	88	17	Linear				
		2	0	0	88	17	88	12	Linear				
2	Cylindrical	1	0	0	88	12	15.5	12	Linear				
		3	Two sides tapered groove	1	0	0	15.5	12	17	Linear			
				2	0	0	12	17	8	17	Linear		
3	0	0	8	17	4.5	12	Linear						
4	Cylindrical	1	0	0	4.5	12	0	12	Linear				
Front internal shape													
1	Right axial groove	1	0	60	58	12	51	12	Linear				
		2	0	60	51	12	51	6	Linear				
2	Right axial groove	1	0	60	51	6	46	6	Linear				
		2	0	60	46	6	46	0	Linear				
Back internal shape													
1	Left axial groove	1	0	-40	10	5	10	0	Linear				
		2	0	-40	10	5	5	5	Linear				
2	Left axial groove	1	0	-40	5	10	5	5	Linear				
		2	0	-40	5	10	0	10	Linear				

and three internal ones. Figure 19 shows the top section of the part with its details, where the red numbers and arrows represent the external shape, and the blue, purple, and orange are the internal shapes. After importing the STEP file into the system, the developed parser reads it, and shows the details to the user as a text via the main window. In this example, the closed-shell has 64 faces, which will be manipulated in the next step, before moving on to the feature recognition. The manipulating processes include: check the convexity or concavity of toroidal faces, finding MaxZ and MinZ as well as MaxX and MinX for each. Then, merge similar cylindrical faces, similar toroidal faces, adjacent toroidal faces and similar conical faces. Also, sorting faces based on MaxZ and splitting ones that related to holes. The final step in this example is the feature recognition process, when the system compares the part's faces with the set of predefined features using the proposed methodology provided in Section 4.1. Moreover, it provides information to the user about the external and three internal shapes. Whilst both the external and internal shapes include the feature's name and information about the faces that form it, the specific details for them are different. That is, regarding the external shape, the system gives the following for each face:  $X$  and  $Z$  start point values ( $X_{sp}$ ,  $Z_{sp}$ ),  $X$  and  $Z$  end point values ( $X_{ep}$ ,  $Z_{ep}$ ) as well as the type of movement from the start to end points (linear or circular). Also, if the face is toroidal, additional data are provided:  $X$  centre value of the curve ( $X_{cc}$ ), curve radius (CR), and the direction of the curve (CW or CCW). Furthermore, the maximum depth ( $D$ ) and width ( $W$ ) of each feature are calculated and printed in the interface window. Whereas the internal shapes' information are detailed as follows:  $X$  centre value ( $X_c$ ),  $Y$  centre value ( $Y_c$ ),  $Z$  centre start ( $Z_{cs}$ ), radius of  $Z_{cs}$  ( $RZ_{cs}$ ),  $Z$  centre end ( $Z_{ce}$ ), radius of  $Z_{ce}$  ( $RZ_{ce}$ ), and the type of movement from the start to end points (linear or circular). Also, the  $X$  centre value of the curve ( $X_{cc}$ ), curve radius (CR), and the direction of the curve (CW or CCW) are provided in the case of toroidal faces. Figure 20 and Tables 3, and 4 show the main window after applying the three steps of parsing, manipulating, and recognition; the details of the recognised external features; and those of the three internal shapes, respectively.

All the details that the system provides can be used in a smart CAPP system, which is the objective of future work. This does not include just the point values, the types of movement from point to point and the direction, for the system also demonstrates in an efficient way the location and specification of the internal shapes. In this example, the through, front, and back internal shapes are provided, as in Fig. 21. Such information can be used whilst generating CAPP, because it helps to select the cutting tool path and whether this part requires being flipped or not. For instance, the CAPP system would select the required tools, processes, and sequence of operations to manufacture both the front and back holes. However, the CAPP system is provided with sufficient information so that when the front hole is manufactured, the system gives instruction to flip the part before machining the back hole.

## 5.2 Case study 2

This example explains how the system identifies a new feature and adds it to the database. For this purpose, the previous design has been used with an alteration, whereby the square groove feature (4) is replaced with an undefined feature, as shown in Fig. 22. To avoid repetition, the steps of the parser, modification, and predefined feature recognition will not be described again here. During the feature recognition process, the new feature is declared in a pop-up window, as in Fig. 23, which has only one meaning: “the smart interactive feature recognition has been activated”. By clicking on the “OK” button, another pop-up window is shown, which includes two text fields. The first is automatically filled with an explanation about the new feature and provides information to the user, such as the number of faces that form the feature as well as the type and specification of each of these faces. In addition, it provides details about the faces that come before and after the feature. Such information allows for understanding regarding the geometrical and topological nature of the new feature, thus helping the user to suggest a name in the second text field, as shown in Fig. 24. In this example, the user has suggested a feature name “V groove concave base”, which is the closest to the feature description. Whilst it is possible to avoid this

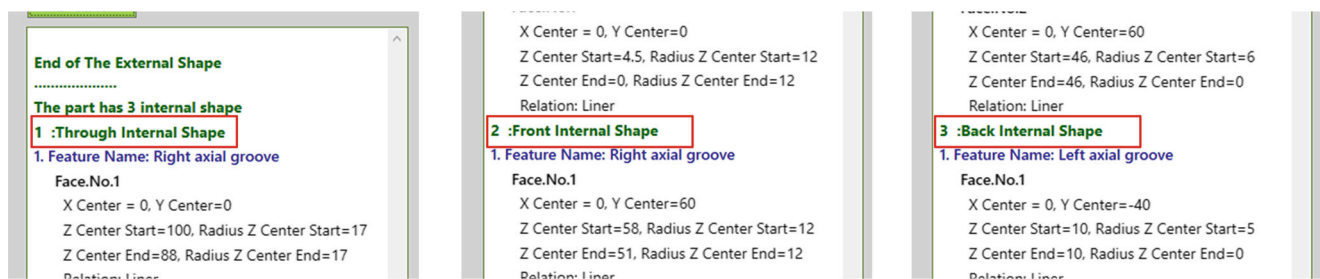
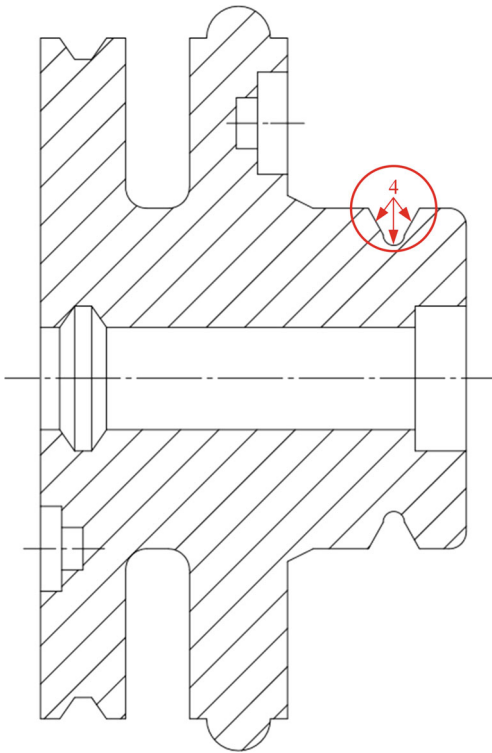


Fig. 21 Three internal shapes with different locations and specifications

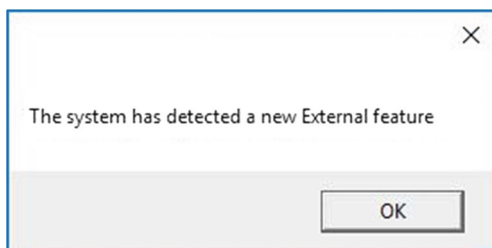


**Fig. 22** A part design with a new feature

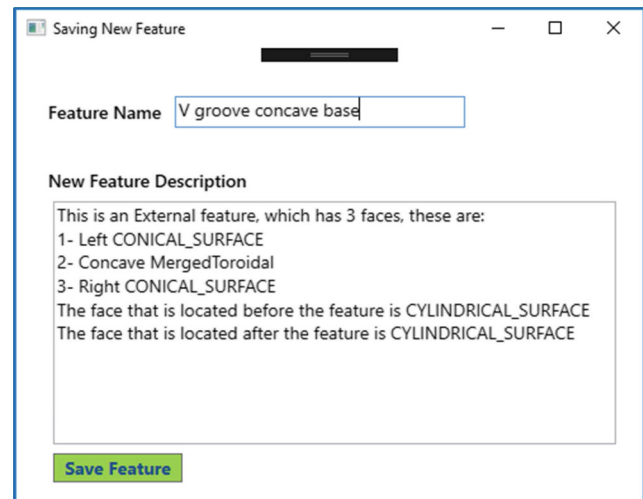
simple interaction part and give the new feature a name automatically, it makes more sense to leave this option for the user, because giving a random name to the new feature might not be satisfying to the user since this name does not describe the nature of the new feature.

By clicking on the “Save Feature” button, the feature is saved in the database with its name and description. Hence, the total number now of the predefined features in the database is 55, instead of the initial number of 54.

The proposed SI-AFR system provides high efficiency regarding the required time to accomplish the feature recognition task. For example, the total time that the system takes to recognise the external and internal features of the model design in Fig. 22 is 5 seconds. This includes one second for each of the parsing process, achieving all the manipulating algorithms, and recognising all the predefined features, as well as two seconds to diagnose, process, and add the new feature to the system’s database.



**Fig. 23** A declaration window about a new feature



**Fig. 24** A description and naming window

## 6 Conclusion

Computer-aided design (CAD) and computer-aided process planning (CAPP) are two essential concepts in manufacturing industry. However, they are isolated from each other due to their different natures since the former provides geometrical and topological information, whereas the latter requires manufacturing information, and hence, a link is required to bridge this gap. Different automatic feature recognition (AFR) systems have been developed and used for the purpose of connecting CAD with CAPP via one of the PDE standards. Whilst these AFR systems have different abilities, they share the same limitation, which is the restriction in recognising a specific set of predefined manufacturing features.

In this research, a smart interactive automatic feature recognition (SI-AFR) system is proposed to recognise the features of rotational parts. As an input, a STEP AP203 file is used given its high capability in transferring design information to downstream applications, such as CAPP. A parser has been developed to extract the geometrical and topological information of a part design from the STEP file and to send it to the next steps. Then, the system manipulates the extracted information to facilitate the feature recognition process. During this progression, the system contributes to solving issues considered drawbacks in previous works, such as identifying the convexity and concavity of toroidal surfaces and efficiently isolating faces that belong to holes and internal shapes. Finally, the feature recognition process has been divided into two parts: recognition of predefined features and smart interactive feature recognition. Whilst the preceding systems have divided a rotational feature with extra details into several ones, such as the square groove with two rounding corners in Fig. 14, the first part of the proposed system allows for the recognition of such

features in just one step. This can simplify and reduce the recognition processes by solving feature's intersecting issues. However, the main contribution of this paper is concentrated in the second part of the system, through new features can be detected, identified, and added to the predefined feature set. This is achieved by extracting the type and specification of each face, the geometrical and topological relation between each two adjacent faces, and the number of the faces that form the new feature. Due to its ability in identifying predefined and new features, it is believed that the system can be considered as representing a new generation of feature recognition systems.

Whilst the SI-AFR system is able to recognise new features, add them to the set of predefined ones, and solve simple feature's intersecting issues, it still has limitations. For example, it is restricted to recognise features that include flat and circular surfaces; whereas, those with freeform surfaces are not included. Also, the method of splitting internal features from external ones is not valid if the internal shape is created through a sphere or conical face as mentioned in Section 3.3.5.

Achieving a smart feature recognition system is the first step in a big project aims to convert design into manufacturing information. Then, the manufacturing information is provided to a smart CAPP system that is the final goal of the project, to help in automatically generating all expected CAPP outputs. Despite the SI-AFR system being able to minimise feature's intersecting issues, extra work and programming are still required for this matter to be completely resolved. This is to optimise the manufacturing process by achieving high production quality with minimum cost and time. Future work includes solving feature's intersecting issues along with the automatic extraction of Product Manufacturing Information (PMI), including tolerance of the shape, tolerance of feature positioning, and roughness.

**Acknowledgements** The authors of the paper would like to sincerely thank the Republic of Iraq Ministry of Higher Education & Scientific Research and the University of Technology, Baghdad for funding the project.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

- Ponomarev D (2015) Enhancing the design transfer process within ENOVIA V6/TVC. PhD thesis
- Oussama J, Abdelilah E, Ahmed R (2014) Manufacturing computer aided process planning for rotational parts. Part 1 : automatic feature recognition from STEP AP203. *Int J Eng Res Appl* 4(5):14–25
- Borkar BR, Puri YM (2015) Automatic extraction of machining features from prismatic parts using STEP for downstream applications. *J Inst Eng Ser C* 96(3):231–243. <https://doi.org/10.1007/s40032-015-0171-3>
- Sunil VB, Pande SS (2009) Automatic recognition of machining features using artificial neural networks. *Int J Adv Manuf Technol* 41(9-10):932–947. <https://doi.org/10.1007/s00170-008-1536-z>
- Babic B, Nestic N, Miljkovic Z (2008) A review of automated feature recognition with rule-based pattern recognition. *Comput Ind* 59(4):321–337. <https://doi.org/10.1016/j.compind.2007.09.001>
- Zhang X, Nassehi A, Newman ST (2014) Feature recognition from CNC part programs for milling operations. *Int J Adv Manuf Technol* 70(1-4):397–412. <https://doi.org/10.1007/s00170-013-5275-4>
- Marjudi S, Amran M, Abdullah KA, Widarto S, Majid NAA, Sulaiman R (2010) A review and comparison of IGES and STEP. In: World Academy of Science, Engineering and Technology, vol 62, pp 1013–1017
- Ismail N, Abu Bakar N, Juri AH (2005) Recognition of cylindrical and conical features using edge boundary classification. *Int J Mach Tools Manuf* 45(6):649–655. <https://doi.org/10.1016/j.ijmactools.2004.10.008>
- Sreeramulu D, Rao CSP (2011) A new methodology for recognizing features in rotational parts using STEP data exchange standard. *Int J Eng Sci Technol* 3(6):102–115. <https://doi.org/10.4314/ijest.v3i6.9>
- Ketan HS, Yaqoub ZH (2010) Built automatic feature recognition system based on sweeping primitive rule. In: Seventh Jordanian International Mechanical Engineering Conference, p 15
- Meeran S, Pratt MJ (1993) Automated feature recognition from 2D drawings. *Comput Des* 25(1):7–17. [https://doi.org/10.1016/0010-4485\(93\)90061-R](https://doi.org/10.1016/0010-4485(93)90061-R)
- Yip-Hoi D, Dutta D, Huang Z (2003) A customizable machining feature extraction methodology for turned components. *J Manuf Syst* 22(2):82–98. [https://doi.org/10.1016/S0278-6125\(03\)90007-0](https://doi.org/10.1016/S0278-6125(03)90007-0)
- Sivakumar S, Dhanalakshmi V (2013) A feature-based system for CAD/CAM integration through STEP file for cylindrical parts. *Indian J Eng Mater Sci* 20(1):21–26
- Zhu J, Kato M, Tanaka T, Yoshioka H, Saito Y (2015) Graph based automatic process planning system for multi-tasking machine. *J Adv Mech Des Syst Manuf* 9(3):1–15. <https://doi.org/10.1299/jamdsm.2015jamdsm00>
- Kang M, Han J, Moon JG (2003) An approach for interlinking design and process planning. *J Mater Process Technol* 139(1-3 SPEC):589–595. [https://doi.org/10.1016/S0924-0136\(03\)00516-8](https://doi.org/10.1016/S0924-0136(03)00516-8)
- Rameshbabu V, Shunmugam MS (2009) Hybrid feature recognition method for setup planning from STEP AP-203. *Robot Comput Integr Manuf* 25(2):393–408. <https://doi.org/10.1016/j.rcim.2007.09.014>
- Li WD, Ong SK, Nee AY (2003) A hybrid method for recognizing interacting machining features. *Int J Prod Res* 41(9):1887–1908. <https://doi.org/10.1080/0020754031000123868>
- Tseng YJ, Joshi SB (1998) Recognition of interacting rotational and prismatic machining features from 3-D mill-turn parts. *Int J Prod Res* 36(11):3147–3165. <https://doi.org/10.1080/002075498192346>
- Liu L, Huang Z, Liu W, Wu W (2018) Extracting the turning volume and features for a mill/turn part with multiple extreme faces. *Int J Adv Manuf Technol* 94(1-4):257–280. <https://doi.org/10.1007/s00170-017-0862-4>
- Shi Y, Zhang Y, Baek S, De Backer W, Harik R (2018) Manufacturability analysis for additive manufacturing using a novel feature

- recognition technique. *Comput Aided Des Appl* 15(6):941–952. <https://doi.org/10.1080/16864360.2018.1462574>
21. Al-wswasi M, Ivanov A, Makatsoris H (2018) A survey on smart automated computer-aided process planning (ACAPP) techniques. *Int J Adv Manuf Technol* 97(1-4):809–832. <https://doi.org/10.1007/s00170-018-1966-1>
  22. Qin Y, Lu W, Qi Q, Liu X, Zhong Y, Scott PJ, Jiang X (2016) Status, comparison, and issues of computer-aided design model data exchange methods based on standardized neutral files and web ontology language file. *J Comput Inf Sci Eng* 17(1):010801. <https://doi.org/10.1115/1.4034325>
  23. (2006) STEP Application Handbook - ISO 10303 - Version 3. June, Scra DOI Free download from <http://isg-scrs.org/STEP/STEPHandbook.html>, [http://www.engen.org.au/index\\_html\\_files/STEP\\_application\\_hdbk\\_63006\\_BF.pdf](http://www.engen.org.au/index_html_files/STEP_application_hdbk_63006_BF.pdf)
  24. (1994) ISO 10303-203:1994. Industrial automation systems and integration\_product data representation and exchange—part 203: application protocol: configuration controlled 3D designs of Mechanical parts and assemblies. International Organization for Standardization (ISO TC 184/SC 4), Geneva, Switzerland
  25. Oussama J, Abdelilah E, Ahmed R (2014) Manufacturing computer aided process planning for rotational parts. Part 2: a new approach for optimizing multiple interpretations of interacting features based on manufacturing rules and metal removal principals. *Int J Innov Appl Stud* 8(1):307–319
  26. Bhandarkar MP, Nagi R (2000) STEP-Based feature extraction from STEP geometry for Agile Manufacturing. *Comput Ind* 41(1):3–24. [https://doi.org/10.1016/S0166-3615\(99\)00040-8](https://doi.org/10.1016/S0166-3615(99)00040-8)
  27. Vangipurapu NM (2013) Automatic feature recognition for rotational components from step files <http://shodhganga.inflibnet.ac.in/handle/10603/14116>
  28. Reddy S, Manidhar DV, Manupati VK, Raj H, Ali A (2016) An intelligent feature based process planning for rotational parts. In: Proceedings of the 2016 international conference on industrial engineering operations management, vol 8-10, pp 2675–2685

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.