

# **Smart automated computer-aided process planning (ACAPP) for rotational parts based on feature technology and STEP file**



by

**Mazin Ghazi Al-wswasi**

Supervisor: **Dr. Atanas Ivanov**

Department of Mechanical and Aerospace Engineering

College of Engineering, Design and Physical Sciences

Brunel University London

United Kingdom

This thesis is submitted for the degree of

*Doctor of Philosophy (PhD) in Smart Manufacturing*

June 2019

## **Abstract**

The concept of smart manufacturing comprises high levels of adaptability with rapid design changes, digital information technology, and more data training. This differs from traditional manufacturing, which depends on constant inputs for the generation of process planning to manufacture a part or requires human intervention if any of the input changes. Smart manufacturing has become a vital issue in the manufacturing industry since the start of the twenty-first century, in terms of time and production cost. One of the most effective concepts for achieving a smart manufacturing industry is the use of Computer-Aided Process Planning (CAPP) which is the key technology that connects Computer-Aided Design (CAD) and the Computer-Aided Manufacturing (CAM) processes. A lot of effort has been spent taking CAPP systems to the next upgraded level that is Automated Computer-Aided Process Planning (ACAPP) in order to provide complete information about the product, in a way that is automated, fast, and accurate. One of the most important aspects in creating an ACAPP system is the use of feature technology, as it is the first step in converting the design to manufacturing features. This includes in particular the development of efficient Automatic Feature Recognition (AFR) systems and solving features intersecting issues.

The implementation of AFR techniques is an indispensable concept for transferring product data between CAD and ACAPP systems. Different international Product Data Exchange (PDE) standards, such as Drawing Exchange Format (DXF), Initial Graphic Exchange Specification (IGES), and Standard for the Exchange of Product (STEP) files are used to accomplish this purpose. Although many AFR techniques and systems have been developed to serve this aim, each of them has limitations. For example, each system is restricted to recognise a specific set of predefined manufacturing features; hence, if new features are included in the model design, they will not be recognised. In this work, a novel and smart interactive AFR (SI-AFR) system has been proposed for recognising features of



rotational parts. A parser has been developed to extract the geometrical and topological information of a part design from a 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. This has been written using C# coding to extract the features' geometrical and topological information from the STEP file. Whilst the first part of the proposed system has the ability of recognising 54 predefined features, the main contribution of this research is concentrated in the second part of the system which allows new features to 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 represents a new generation of feature recognition systems.

Also, a “features subtraction” system has been created as an optimal solution for complex features intersecting cases. It takes the final manufacturing features from the SI-AFR system as an input. The system has seven steps for analysing, processing, and calculating intermediate features. The intermediate features represent layers of material to be removed, in an optimal sequence. These are recognised by scanning in all directions of the part, to determine the intersecting areas between the final manufacturing features. Such a system provides a whole vision of transferring a blank into the desired shape via step-by-step rough turning, drilling, and boring processes.

The results from the SI-AFR and features subtraction systems depend on the geometrical and topological information of the pre-defined and new features. These are analysed for the purpose of automatically generating CAPP outputs, such as the process selection, cutting tools, sequence of operations, and generating G-code. This is to reduce the time and production cost, as well as human intervention, and hence significantly contributes to an organisations efforts in sustainability. The proposed ACAPP system has been practically validated, clearly demonstrating how it surpasses the capabilities of traditional CAM software, since all the outputs are achieved automatically, which CAM software are currently not capable of. The final manufacturing features of the part have been

produced accurately, compared to the design features, in terms of specified design dimensions and tolerances. The current version of the system covers rotational symmetrical parts, however this work can be extended to include rotational non-symmetrical and prismatic parts.

**Keywords:** Automatic CAPP, Smart manufacturing, Automatic feature recognition, Features subtractions

*To,*

*This thesis is dedicated to the memory of my father "Ghazi Al-wswasi" and my wife "Wasan Tareq". I miss you, and hope you are proud of me.*

## **Declaration**

I declare that this thesis is my own work and is submitted for the first time to the Post-Graduate Research Office. The study was originated, composed and reviewed by myself and my supervisors in the College of Engineering, Department of Mechanical and Aerospace Engineering, Brunel University London, UK. All the information derived from other works has been properly referenced and acknowledged.

SIGNED: ..... DATE: .....

Mazin Al-wswasi

SIGNED: ..... DATE: .....

Atanas Ivanov

## **Acknowledgements**

I would like to express my sincere gratitude to the people who have helped and supported me in working to realize this research project.

First and foremost, I would like to sincerely thank the Republic of Iraq Ministry of Higher Education & Scientific Research, Iraqi Cultural Attaché, and the University of Technology, Baghdad for granting me a fully funded scholarship during my study.

I would like to thank my supervisor, Dr. Atanas Ivanov, for his patience, assistance, and motivation. His guidance helped me throughout my research, during the publishing of journal papers, and the writing of this thesis.

I would also like to show endless appreciation for my wife Zainab and three lovely children Daniah, Yousif, and Meriam, who always surrounded me with love and emotional support. Moreover, I am forever indebted to my mother and sister for encouraging me and believing that I could vanquish any challenges in my life.

I am also grateful to my other family members and friends who have supported me along the way. In particular, I am thankful to my brother-in-law Rafid Al-rubaii who taught me programming from scratch and never stopped answering my questions in this matter. I would also like to express my deepest gratitude to my dearest friend Dr. Angela Mumith for not only her advice in writing up the thesis, but also due to the fact that she kept me feeling positive when I had difficulties during this journey of discovery. A very special thanks goes out to Ali Khaghani, who is a brother more than a friend, for giving valuable information in our field of research, and all the fun we have had in the last three years.

I would like to thank the staff of Brunel University, and particular Patricia Correia from the PGR office for dealing patiently with my queries and giving advice that helped to submit the thesis with

minimum stress. I would also like to show great appreciation for Philip Mackenzie for helping me accomplish the practical side of my work, at a very critical time.

Finally I would like to thank all my colleagues at the office who have shared with me difficult and fun times during my PhD. They have filled the place with an environment that helped me to work without additional stress.

# Table of contents

<b>Abstract</b>	<b>ii</b>
<b>Dedication</b>	<b>v</b>
<b>Declaration</b>	<b>vi</b>
<b>Acknowledgement</b>	<b>vii</b>
<b>Table of contents</b>	<b>ix</b>
<b>List of figures</b>	<b>xiv</b>
<b>List of tables</b>	<b>xviii</b>
<b>List of Abbreviation</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background of research . . . . .	1
1.2 The concept of CAPP . . . . .	3
1.3 Justification and motivation . . . . .	4
1.4 Research aims and objectives . . . . .	5
1.5 Key contributions and novelty . . . . .	6
1.6 Outline of thesis . . . . .	8
<b>2 Literature Review</b>	<b>9</b>
2.1 Introduction . . . . .	9

---

2.2	Required inputs of the CAPP . . . . .	9
2.2.1	Feature technology . . . . .	10
2.2.1.1	Syntactic Pattern Recognition . . . . .	11
2.2.1.2	Graph Based . . . . .	14
2.2.1.3	Hint Based . . . . .	17
2.2.1.4	Logic Rule Based . . . . .	19
2.2.1.5	Artificial Neural Network . . . . .	20
2.2.1.6	Hybrid AFR systems . . . . .	21
2.2.2	Geometric dimensions and tolerances . . . . .	23
2.2.3	Materials . . . . .	25
2.2.4	Surface finishes . . . . .	26
2.2.5	Machining process and process capabilities MP&PC . . . . .	27
2.2.6	Summary . . . . .	28
2.3	Output activities of CAPP . . . . .	29
2.3.1	Process selection . . . . .	29
2.3.2	Sequence of operations . . . . .	30
2.3.3	Cutting tools . . . . .	31
2.3.4	Cutting conditions . . . . .	33
2.3.5	Selection of jigs and fixtures . . . . .	35
2.3.6	Identifying the tool path for both of rough and finish cycles . . . . .	37
2.3.7	Estimate the time and the cost to manufacture the part . . . . .	39
2.3.8	Summary . . . . .	42
2.4	Discussion and summary of literature . . . . .	42
<b>3</b>	<b>STEP file structure and characteristics</b>	<b>44</b>
3.1	Introduction . . . . .	44
3.2	Background . . . . .	45
3.3	Structure of a STEP (AP-203) file . . . . .	45
3.3.1	Bound . . . . .	48



---

3.3.2	Surface . . . . .	50
3.4	Discussion and summary . . . . .	50
<b>4</b>	<b>A smart interactive AFR system for rotational parts</b>	<b>52</b>
4.1	Introduction . . . . .	52
4.2	Developing a parser for a STEP file . . . . .	55
4.3	Algorithms to manipulate the STEP before moving on to feature recognition . . . . .	57
4.3.1	Find convexity or concavity of a toroidal face . . . . .	57
4.3.2	Merging symmetrical faces . . . . .	59
4.3.3	Merging adjacent toroidal faces . . . . .	60
4.3.4	Sorting faces . . . . .	64
4.3.5	Split external features from internal features and holes . . . . .	65
4.4	Smart interactive automatic feature recognition (SI-AFR) . . . . .	68
4.4.1	Recognition of predefined features . . . . .	69
4.4.2	Interactive feature recognition . . . . .	73
4.4.2.1	Extracting the new feature information . . . . .	73
4.4.2.2	Processing the new feature information . . . . .	74
4.4.2.3	Saving the new feature information . . . . .	76
4.5	Case study . . . . .	77
4.5.1	Case study 1 . . . . .	77
4.5.2	Case study 2 . . . . .	82
4.6	Discussion and summary . . . . .	84
<b>5</b>	<b>Intersection of manufacturing features</b>	<b>86</b>
5.1	Introduction . . . . .	86
5.2	Defining the problem . . . . .	87
5.3	A methodology of subtracting . . . . .	89
5.3.1	Features subtraction of external shapes . . . . .	90
5.3.2	Features subtraction of internal shapes . . . . .	102
5.4	Case study . . . . .	103

---

5.4.1	External shape . . . . .	103
5.4.2	Internal shape . . . . .	113
5.5	Discussion and summary . . . . .	121
<b>6</b>	<b>Geometric dimensioning and tolerancing</b>	<b>123</b>
6.1	Introduction . . . . .	123
6.2	Problem definition: Semantic representation of GD&T data . . . . .	125
6.3	Investigation of the solution . . . . .	126
6.4	Discussion and summary . . . . .	127
<b>7</b>	<b>Results and conclusion</b>	<b>128</b>
7.1	Introduction . . . . .	128
7.2	Launching the ACAPP system . . . . .	128
7.3	Process selection . . . . .	129
7.4	Cutting tools . . . . .	130
7.4.1	Roughing and finishing . . . . .	131
7.4.2	Feature location . . . . .	132
7.4.3	Feature geometry . . . . .	132
7.5	Sequence of operations . . . . .	135
7.6	Generating G-code . . . . .	136
7.7	Case study . . . . .	140
7.8	Methodology evaluation . . . . .	145
7.8.1	Comparison to CAM software . . . . .	145
7.8.2	Practical validation of the ACAPP system . . . . .	147
7.9	Discussion and summary . . . . .	155
<b>8</b>	<b>Recommendations for future work</b>	<b>157</b>
8.1	Covering CAPP inputs . . . . .	157
8.2	Optimising and considering the uncovered CAPP outputs . . . . .	158
8.3	Extending functionality to prismatic parts . . . . .	159

---

8.4 How AI will effect manufacturing in future . . . . .	159
<b>References</b>	<b>160</b>
<b>Appendix A</b>	<b>171</b>
<b>Appendix B</b>	<b>183</b>
<b>Appendix C</b>	<b>189</b>
<b>Appendix D</b>	<b>196</b>

# List of figures

1.1	The architecture of a smart manufacturing system . . . . .	3
1.2	CAPP system with its inputs and outputs . . . . .	5
2.1	Block diagram of a SPR system [44] . . . . .	12
2.2	An example of a 3D prismatic part (a) 14 faces of the prismatic part, and (b) explanation of AAG for the part in (a) [51] . . . . .	15
2.3	Three different parts with identical AAG representation [51] . . . . .	16
2.4	An example of intersecting features [53] . . . . .	16
2.5	Architecture of the Hint-Based approach [57] . . . . .	18
2.6	Architecture of the FEATURES system [60] . . . . .	19
3.1	The Structure of a STEP AP 203 file . . . . .	46
3.2	A cylindrical part (a) how a cylinder is represented during the design phase, and (b) four faces of the cylinder with some details of the first face . . . . .	47
3.3	Part of the STEP AP 203 file of the cylinder in fig.3.2 . . . . .	47
4.1	Examples of external and internal predefined features . . . . .	53
4.2	The structure of the proposed SI-AFR system . . . . .	54
4.3	Part of the resultant STEP file after applying the parser . . . . .	56
4.4	Toroidal surfaces (a) concave, and (b) convex . . . . .	58
4.5	The effect of the start and end point on forming toroidal shapes (a) concave shape, and (b) convex shape . . . . .	58
4.6	An algorithm for merging the information of two symmetrical cylindrical faces . . . .	60

---

4.7	Merging adjacent toroidal faces (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	61
4.8	A special case for merging adjacent toroidal faces (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	62
4.9	An algorithm for merging the information of two adjacent toroidal faces	63
4.10	A cylinder with a blind hole (a) schematic representation and (b) a side view of the cylinder	65
4.11	A part design includes 14 external and internal faces	66
4.12	The remainder of the part after the virtually subtraction of two nodes of internal faces	68
4.13	A square groove with two round corners in the base	70
4.14	A map of the predefined features recognition loops	72
4.15	A blind square groove	73
4.16	An example of a new feature	74
4.17	Saving a new feature based on relationships between its geometrical values (a) a new feature with specific geometrical values and (b) a feature with same characteristics and different values	76
4.18	Top section of case study 1	78
4.19	The system's main window after applying the three steps for case study 1	79
4.20	Three internal shapes with different locations and specifications	82
4.21	A part design with a new feature	83
4.22	A declaration window about a new feature	83
4.23	A description and naming window	84
5.1	Different strategies for considering features intersecting (a) a design model, (b) original blank, (c), (d), and (e) three different strategies of dividing the machining area	88
5.2	An example of which points are taking as maximum X and Z	90
5.3	Three features have the same X value of the blank	91

---

5.4	Two groups of features have the same X value of the blank . . . . .	92
5.5	Two different examples where the second highest X value can be . . . . .	94
5.6	An example of an intermediate feature with its three faces . . . . .	97
5.7	A special case of an intermediate feature . . . . .	98
5.8	A part and its blank after removing the first intermediate feature . . . . .	98
5.9	Steps of machining intermediate and final dependent features (a) second, (b) third, and (c) fourth, and (d) final shape . . . . .	99
5.10	A part includes a duplication case of an intermediate feature . . . . .	100
5.11	The sixteen possibilities of intermediate features . . . . .	101
5.12	The sequence of forming an internal shape . . . . .	102
5.13	Removing the first intermediate feature . . . . .	105
5.14	Removing the second intermediate feature . . . . .	107
5.15	Removing the third intermediate feature . . . . .	108
5.16	The machining process (a) machining of the right convex corner, and (b) machining of the radial square groove . . . . .	109
5.17	Removing the fourth intermediate feature . . . . .	111
5.18	The machining process (a) machining of the radial groove with two rounded corners, and (b) machining of the two sides tapered groove . . . . .	111
5.19	Removing the first intermediate feature from the through internal shape . . . . .	115
5.20	The machining process (a) machining of the right axial groove, and (b) machining of the two sides tapered groove . . . . .	115
5.21	The machining process (a) machining of the first intermediate feature, and (b) ma- chining of the right axial groove (F1) . . . . .	117
5.22	The machining process (a) machining of the first intermediate feature , and (b) machining of the left axial groove (F2) . . . . .	118
5.23	The main window . . . . .	120
6.1	A CAD system with its inner translator . . . . .	124

---

6.2	A cylindrical part with three different diameters and lengths (a) the representation of the cylindrical part in the design phase, (b) dimensions as understood by the STEP file, (c) dimensions specified by the designer . . . . .	126
7.1	Examples of insert shapes that are encoded based on ISO standards . . . . .	130
7.2	Examples of holder styles that are encoded based on ISO standards . . . . .	130
7.3	The effect of the feature position in selecting cutting tools (a) a right step feature on the right side of the part, and (b) a radial square groove feature in the middle of the part	132
7.4	The effect of the internal shape location in selecting the cutting process (a) an internal shape is located at the axis of rotation, and (b) an internal shape is located with an offset from the axis of rotation . . . . .	134
7.5	Two different sequence of operations strategies based on the features of the part and the clamping zone . . . . .	136
7.6	Setting the Z-zero point(a) the Z-zero at the first set-up, and (b) the Z-zero at the second set-up after flipping the workpiece . . . . .	138
7.7	Calculating the start point based on the tool geometry (a) a tool path start point based on the tool width, and (b) a tool path start point is equal to that of the feature . . . . .	140
7.8	Examples of insert shapes that are encoded based on ISO standards . . . . .	143
7.9	The final ACAPP system window . . . . .	145
7.10	Dividing the design into machining areas in CAM software (a) a part profile as a connected sequence of segments in CAM software, and (b) divide the machining area into sub-areas based on the user's experience . . . . .	147
7.11	The left section of the model design . . . . .	149
7.12	The instruction window generated of the model design used in the practical validation	154
7.13	The initial blank of the model design used in the practical validation . . . . .	154
7.14	The final machined part of the model design used in the practical validation . . . . .	155

# List of tables

4.1	Data of A square groove with two round corners in the base . . . . .	71
4.2	Extracting a new feature's data . . . . .	75
4.3	The details of the recognized external features . . . . .	80
4.4	The details of the three internal shapes . . . . .	81
5.1	Intermediate and independent features of the external shape . . . . .	112
5.2	The details of the recognized internal features . . . . .	119
7.1	Examples of different grooves geometry and the criteria of choosing appropriate tool parameters . . . . .	133
7.2	The machining process for each external and internal feature . . . . .	142
7.3	The total number of tools required to manufacture the part design with their details .	142
7.4	The sequence of operations for all the external and internal feature . . . . .	144
7.5	The recognized external features of the model design used in the practical validation	150
7.6	The recognized internal features of the model design used in the practical validation .	151
7.7	External intermediate and independent features of the model design used in the practical validation . . . . .	152
7.8	Internal intermediate and independent features of the model design used in the practical validation . . . . .	153



# List of Abbreviation

<b>2D</b>	Two Dimensions
<b>3D</b>	Three Dimensions
<b>AAG</b>	Attributed Adjacency Graph
<b>ACAPP</b>	Automated Computer Aided Process Planning
<b>AFR</b>	Automatic Feature Recognition
<b>AM</b>	Additive Manufacturing
<b>AMPs</b>	Advanced Machining Processes
<b>ANNs</b>	Artificial Neural Networks
<b>Aps</b>	Application Protocols
<b>B-rep</b>	Boundary Representation
<b>CAD</b>	Computer Aided Design
<b>CAFDs</b>	Computer-Aided Fixture Designs
<b>CAM</b>	Computer Aided Manufacturing
<b>CAPP</b>	Computer Aided Process Planning
<b>CC</b>	Cutter Contact
<b>CCW</b>	Counter-ClockWise
<b>CE</b>	Cost Estimation
<b>CESS</b>	Cost Estimation and Supplier Selection
<b>CIM</b>	Computer Integrated Manufacturing
<b>CR</b>	Curve Radius
<b>CSG</b>	Constructive Solid Geometry
<b>CSH</b>	Maximum Scallop Height

---

<b>CW</b>	ClockWise
<b>DBF</b>	Design by Features
<b>DOF</b>	Degree of Freedom
<b>DTCE</b>	Detailed Time-Cost Estimation
<b>DXF</b>	Drawing Exchange Format
<b>EAAG</b>	Enhanced Attributed Adjacency Graph
<b>EXCATS</b>	Expert Computer Aided Tool Selection System
<b>FEM</b>	Finite Element Method
<b>FLGs</b>	Face Loop Graphs
<b>FRV</b>	feature Representation Vector
<b>GAs</b>	Genetic Algorithms
<b>GB</b>	Graph Based
<b>GD&amp;T</b>	Geometric dimensioning and tolerancing
<b>GEPPACE</b>	Generative Process Planning and Cost Estimation
<b>HKS</b>	Heat Kernel Signature
<b>IF2</b>	Integrated Incremental Feature Finder
<b>IGES</b>	Initial Graphic Exchange Specification
<b>IPPM</b>	Integrated Process Planning Model
<b>ITSS</b>	Incorporated Tool Selection Systems
<b>MATF</b>	Merging Adjacent Toroidal Faces
<b>MC</b>	Machining Complexity
<b>MP&amp;PC</b>	Machining process and process capabilities
<b>MSF</b>	Merging Symmetrical Faces
<b>MSH</b>	Minimum Scallop Height
<b>MTS</b>	Maximum Turnable State
<b>MTV</b>	Maximum Turn-able Volume
<b>NC</b>	Numerical Control
<b>NLPPs</b>	Non-Linear Process Plans
<b>NN</b>	Neural Networks
<b>OOFF</b>	Object-Oriented Feature Finder system

---

<b>OSRI</b>	Operation Sequencing Rating Index
<b>PD</b>	Process Detailing
<b>PDE</b>	Product Data Exchange
<b>PNs</b>	Petri Nets
<b>RFQ</b>	Request-For-Quotation
<b>RTCE</b>	Rough-Cut Time-Cost Estimation
<b>RZce</b>	Radius Z-Centre Start
<b>RZcs</b>	Radius Z-Centre Start
<b>SA</b>	Simulated Annealing
<b>SAT</b>	Standard ACIS Text
<b>SCTF</b>	Super Constraint Tolerance Feature
<b>SI-AFR</b>	Smart Interactive Automatic Feature Recognition
<b>SP</b>	Sequence Planning
<b>SPR</b>	Syntactic pattern recognition
<b>STEP</b>	Standard for the Exchange of Product
<b>TE</b>	Time Estimation
<b>TTRS</b>	Technologically and Topologically Related Surface
<b>UML</b>	Unified Modelling Language
<b>Xc</b>	X-Centre
<b>Xcc</b>	X-Curve Centre
<b>Xep</b>	X-End Point
<b>Xsp</b>	X-Start Point
<b>Yc</b>	Y-Centre
<b>Zcc</b>	Z-Curve Centre
<b>Zce</b>	Z-Centre End
<b>Zcs</b>	Z-Centre Start
<b>Zep</b>	Z-End Point
<b>Zsp</b>	Z-Start Point

# Chapter 1

## Introduction

### 1.1 Background of research

Process planning is an essential link between design and manufacturing. It determines the manufacturing processes, their sequences and the conditions to convert a design into a physical component economically and competitively [1–6]. With the traditional approach, manufacturing experts used their experience and knowledge to solve process planning issues and give instructions about how to manufacture products [7, 8]. Since the early 1960's, computers have been used to assist design, process planning and manufacturing activities due to their ability to perform complex functions in a fast and accurate way [1, 9]. Thus, the three traditional concepts (design/process planning/manufacturing) have now become known as Computer Aided Design (CAD), Computer Aided Process Planning (CAPP), and Computer Aided Manufacturing (CAM) in the industrial world [9].

The term CAD refers to the use of computer systems to assist in generating surfaces and 3D solids of products [9, 10]. Design information needs to be shared accurately among designers, process planners and other participants in the production process [2, 11]. A part's information is represented internally in CAD systems and thus, it is logical to let computers analyse that database of information directly to create a process plan. This is instead of converting the design from the computer into a drawing on paper and back again, to produce the manufacturing report [12]. However, sharing design information is challenging because different enterprises, which are involved in designing and manufacturing, use different CAD systems [13, 14]. As an ideal solution, many companies

and organisations have designed and presented formats of international *Product Data Exchange* (PDE) standards to solve this issue: *Drawing Exchange Format* (DXF), *Initial Graphic Exchange Specification* (IGES), and *Standard for the Exchange of Product* (STEP) [13–15]. The benefits of these standards are not limited to handling and saving the product data, for it is also possible for them to be used in identifying the manufacturing features of the product [15, 16]. In fact, the standard formats are essential requests, since manufacturing features are used for almost all CAPP systems for product description [2].

Computer Aided Process Planning (CAPP) is the use of computers to aid process planners in a systematic determination of convenient methods in the manufacturing of a part or product [17, 18]. Niebel in 1965 was the first who discussed the use of computers to assist in the selection of basic processes in order to plan manufacturing processes for a given functional design [1, 6, 19]. To date, much research and many publications have been devoted to develop CAPP systems. However, in comparison with CAD and CAM systems, few CAPP systems have reached the stage where they could provide significant solutions to the manufacturing industry [20]. Part of this slow progress is due to the enormous complexity of the nature, and the dynamic aspects of the process planning task [12, 19, 20]. Furthermore, it might be discouraging to know that since Sutton mentioned in 1989 that, “*there are no fully automated CAPP system and almost all of them still require extensive manual operations*” [21], this statement still holds true, in spite of the significant efforts that have been made to take the traditional CAPP systems to the next automated level [22, 23]. Hence, it is very important to pinpoint the barriers that obstruct this goal. The following research question will be examined to achieve a smart Automated Computer Aided Process Planning (ACAPP) system, which is an extension of the functionalities of CAPP:

1. What are the required inputs and expected outputs to generate ACAPP?
2. What optimisation functions can be implemented (cost, productivity, etc.)?
3. What factors and strategies are needed, and what further knowledge could be added?

Figure 1.1 shows a smart manufacturing system as a sequence of a product life cycle, which is supported by computers throughout the three main stages (design process/CAPP based on automatic feature recognition (AFR)/manufacturing process).

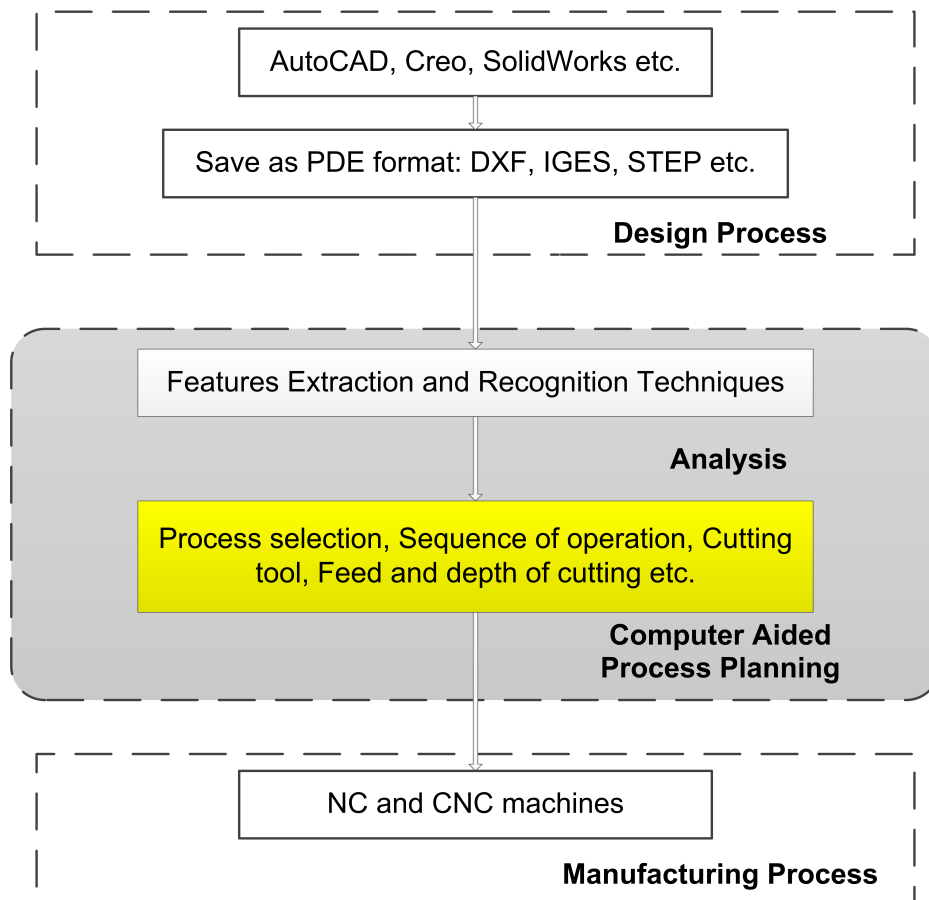


Figure 1.1 The architecture of a smart manufacturing system

Where the analysis includes features extraction and recognition in addition to all other information, such as dimensions and tolerances, surface finish, and the part datum.

## 1.2 The concept of CAPP

As mentioned in section 1.1, it is clear that the decisive role of process planning is as a bridge between the design and manufacturing processes and that an efficient CAPP can reduce the lead time [21, 24]. In addition, essential issues such as cost of components, company competitiveness, production planning, and production efficiency could be determined with the associated CAPP [25]. Owing to the reduction in the number of experienced process planners in industry, the field of CAPP research is growing and receiving greater attention than ever before [24].

Whilst there has been much research to automate CAPP, the results are still considered to be lagging behind where they should be due to the multidisciplinary nature of process planning [5, 26].

The various aspects of process planning are classified according to the required inputs and operational details (outputs). The required inputs of a CAPP include:

1. Feature technology;
2. Geometric dimensioning and tolerancing (GD&T);
3. Materials;
4. Surface finishes;
5. Machining process and process capabilities (MP&PC).

These inputs should be accurately analysed and evaluated in order to generate process planning based on available machinery and processes [4, 17].

The final output of the process planning activities includes:

1. Processes selection;
2. Sequence of operation;
3. Cutting tools;
4. Cutting conditions;
5. Selection of jigs and fixtures;
6. Identifying the tool path for both the rough and finish cycles;
7. Estimating the time and the cost to manufacture the part[3, 17, 26].

Figure 1.2 shows the general structure of CAPP systems with their inputs and outputs.

### **1.3 Justification and motivation**

There are many methods and technologies of CAPP, with those that are widely used including feature-based technologies, knowledge-based systems, artificial neural networks, genetic algorithms (GAs), fuzzy set theory and fuzzy logic, Petri nets (PNs), agent-based technologies, internet-based

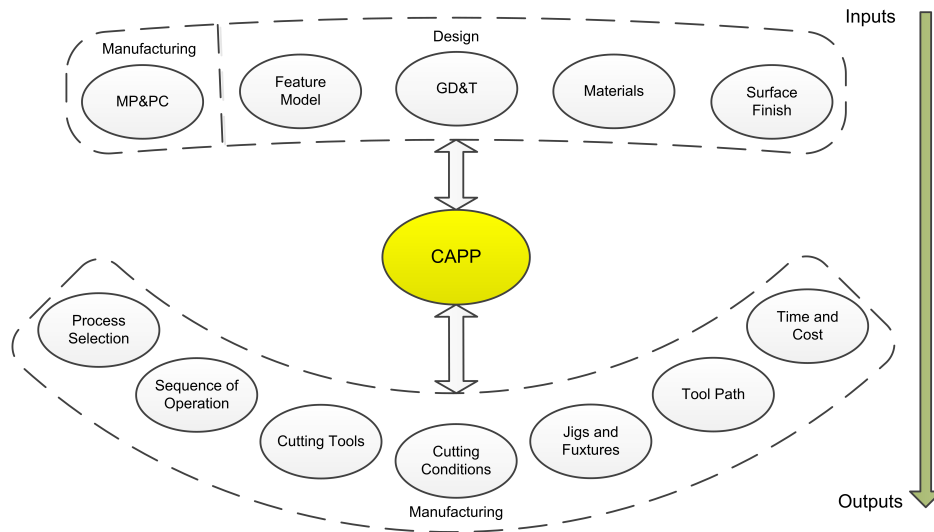


Figure 1.2 CAPP system with its inputs and outputs

technologies, and the standard for the exchange of product data (STEP)-compliant method [1, 6]. It is important to re-examine and analyse the current methodologies and approaches to determine whether they will lead to convincing solutions of ACAPP.

There are numerous reasons why developing a smart ACAPP system would be massively beneficial to the manufacturing industry. Firstly, it can reduce human intervention in regenerating the required manufacturing information of a CAD model by automating the process. Secondly, the output manufacturing information of the system can be saved and presented to the machine operator as documentary files, including complete process planning in terms of instructions. Thirdly, based on the points mentioned above, a smart ACAPP system would significantly minimise the time and production cost, and increase flexibility due to the increased intelligence of the system. Finally, all these advances enhance the sustainable capabilities of the manufacturing industry, by reducing waste of resources. This research intends to elevate current capabilities of traditional CAPP to the next level of intelligence and automation.

## 1.4 Research aims and objectives

The aim of this research is to take CAPP systems a step forward to the next level in the form of smart ACAPP. This is by solving some AFR drawbacks such as recognising new manufacturing features



that are not included with the predefined ones and features intersections issues. Whilst this research attempts to cover as much as possible of CAPP inputs and outputs, more emphasis has been given to the AFR since it is considered an indispensable concept for transferring product data between CAD and CAPP systems; hence, it helps in generating CAPP outputs. Finally, the instructions of how to manufacture a part is automatically generated and presented as text files to the machine operator for delivering a complete picture of the process planning. Such a system would reduce the expected time to produce instructions setting up a blank on a machine and manufacturing the part, hence reducing the overall lead time. Also, another value that smart ACAPP adds is that it does not require human experience to carry out these steps, which is traditionally the case. In order to achieve these aims the following objectives has been set:

1. Reviewing previous technologies and methods developed in CAPP and AFR, and presenting a critical assessment of the work achieved in the last decades.
2. Gaining a solid knowledge about the structure of the STEP file, and diagnosing its benefits and limitations.
3. Investigating, assessing, and defining the main AFR problems in terms of recognising new manufacturing features that might be included in a CAD model as well as features intersecting issues.
4. Investigating, assessing, and defining the existing issues in recognising some features of rotational parts based on a STEP file information, such as identifying whether a toroidal face in a part is a concave or convex surface, and if a feature is external or internal.
5. Developing a smart interactive AFR system using C#.
6. Solving feature intersecting issues.

## **1.5 Key contributions and novelty**

As a result of this work, a step towards a smart ACAPP system for manufacturing rotational parts has been accomplished. This is the first advancement in developing a completely smart ACAPP, which

covers all the required inputs and expected outputs, for rotational and prismatic parts. Such a system can be used as a commercial industrial application to reduce human intervention in process planning, as well as the time and production cost. The development of this system is associated by the following contributions to knowledge in the CAPP field:

1. A critical assessment about the degree of automation of each required input and expected output of CAPP systems, which illustrates the missing aspects in smart ACAPP generation. Also, this critical assessment demonstrates the benefits and limitations of different automatic feature recognition systems and techniques, currently found in the literature. This research has been published as a journal paper (M. Al-Wswasi, A. Ivanov, and H. Makatsoris, "A survey on smart automated computer-aided process planning (ACAPP) techniques," *The International Journal of Advanced Manufacturing Technology*, 2018) [23].
2. The development of a novel and Smart Interactive AFR (SI-AFR) system for recognising manufacturing features of rotational parts has many contributions that solve issues considered drawbacks in previous works:
  - (a) geometrical and topological information of new features can be detected, identified, and added automatically to the database of predefined features;
  - (b) identifying the convexity and concavity of toroidal surfaces;
  - (c) isolating external features from internal ones that form holes and internal shapes;
  - (d) the proposed system accepts any of the three application protocols AP203, AP214, or AP242, in contrast with the previous AFR systems when each of them has been developed based on only one application protocol (AP).

This work has been published as a journal paper (M. Al-wswasi and A. Ivanov, "A novel and smart interactive feature recognition system for rotational parts using a STEP file," *The International Journal of Advanced Manufacturing Technology*, 2019) [27].

3. A new methodology of features subtraction has been developed to solve features intersecting issues in rotational parts. This is to help planning the rough cutting cycle when most of the material is removed from the original stock. A journal paper of this methodology has been

submitted to The International Journal of Advanced Manufacturing Technology as continuation of the two previous publications and is currently under review.

4. The result from the SI-AFR and features subtraction systems has been used to automatically generate certain CAPP outputs, such as the selection of machining processes, cutting tools, sequence of operations, as well as generating G-code.

## 1.6 Outline of thesis

The thesis is structured as follows:

**Chapter 1 (Introduction)** gives a brief background of the research and explains the major motivations, objectives, aims, and key contributions of this work.

**Chapter 2 (Literature review)** presents a literature review about principles, functions, requirements, and different methodologies and approaches to CAPP regarding inputs and outputs. Also, it gives a special emphasis about the fundamental concepts and methods of AFR techniques.

**Chapter 3 (STEP file structure and characteristics)** explains the structure of the STEP file taking an example of a simple 3D shape to illustrate its outline.

**Chapter 4 (A smart interactive AFR system for rotational parts)** presents the novel and smart interactive AFR system for rotational parts using a STEP file. The way the problem is defined and the system is built in terms of recognising new manufacturing features and adding them to the system's database is outlined. The results from the preliminary experiment is analysed and discussed.

**Chapter 5 (Intersection of manufacturing features)** clarifies the applied methodology in solving features intersecting issues. An example with its results is discussed.

**Chapter 6 (Geometric dimensioning and tolerancing)** describes an attempt of automatically extracting GD&T data from a CAD model as a semantic representation.

**Chapter 7 (Results and conclusion)** includes the implementation of the results from Chapters 4 and 5 for the purpose of generating smart CAPP outputs. Also, it outlines the assessment and conclusion of the whole research.

**Chapter 8 (Recommendations for future work)** presents recommendations for possible future directions of research.

## **Chapter 2**

# **Literature Review**

### **2.1 Introduction**

This literature review chapter describes different methods and technologies of CAPP in terms of required inputs and expected outputs. However, it gives special attention to the current AFR methodologies and approaches because this aspect is considered the key to success in any CAPP system. This is to reconsider and analyse the previous work, to determine whether convincing solutions has been reached to obtain ACAPP. Also, a vision of how it is possible to optimise the covered inputs and outputs will be mentioned at the end of this chapter.

### **2.2 Required inputs of the CAPP**

In order to generate instructions to manufacture a part, CAPP has to recognise and analyse the product model data provided by a CAD system. The product data includes manufacturing features, which are identified based on geometric features including dimensions, tolerances and roughness as well as relevant non-geometric information, such as materials and hardness [28]. However, CAD and CAPP systems have different product data descriptions, for example, the former is geometry-based, whereas the latter is manufacturing feature-based [29]. In addition, it is difficult to embed non-geometric information as real attributes in most, if not all, current CAD models. As a matter of fact, even if the non-geometric data are incorporated in these models, they are represented as text in the same way as

technical notes in a drawing. Hence, human intervention is needed to regenerate the manufacturing information when a CAD model is to be transferred to downstream users, such as the process planner [30]. This section explains the concepts of the required CAPP inputs, and is aimed at diagnosing the shortcomings that prevent auto interlinking between CAD/CAPP/CAM systems.

### 2.2.1 Feature technology

Computer aided design systems have been used since the 1960s to assist draftsmen with tedious drawings and redrawing tasks, as well as to improve drafting efficiency [17, 31]. All CAD packages include geometric or modelling features which describe a part using low-level entities, such as points, lines, and curves. Whilst the CAD packages store the information in their own databases, each package has a different database structure, for there is no one standard that has been developed to be used by all [30]. Furthermore, the geometrical features of a part are not suitable to be used directly in CAPP systems; hence, they should be converted to high-level manufacturing features, such as holes, grooves, and pockets. However, feature technology is expected, in the near future, to achieve better integration between CAD and CAPP. Feature technology represents an essential tool for computer integrated manufacturing (CIM), which is considered a focal research area in manufacturing industry. The main two approaches in this field are design by features (DBF) and automated feature recognition [32–36].

In the DBF methodology, a part is designed using a set of predefined features in the modelling system libraries, such as slots, holes and pockets, etc. [33, 35]. This approach facilitates modelling design data by storing high level information in the CAD module's data structure [37]. In addition, this method can speed up the design process, since it remarkably reduces the amount of work required to recognise the features [35]. Despite the advantages of DBF, it has not yet reached expectations. The main drawback is the unlimited set of features in design [32]. Also, as this approach stores high-level feature information in CAD packages, it is difficult to transfer and exchange this level of information among different systems [38]. Furthermore, despite DBF reducing the work of feature recognition, the need for it still exists [32, 39]. Hence, all the systems that include feature modelling require feature recognition techniques [35].

In AFR techniques, low-level geometric entities of part design models, which are created in CAD systems, are converted to manufacturing features. This is to be used in various engineering tasks, such as CAPP, CAM and inspection [32, 40]. Different AFR approaches have been developed, since there are various classifications of features, and they are used in different application areas [40]. Whilst few AFR systems use DXF and IGES files for the purpose of extracting geometrical data of the part, more emphasis has been given to STEP AP203, AP214, and AP242 files. These APs were developed by different groups, and each covers a specific requirement. For example, STEP AP203 defines the geometry, topology, and configuration management data of solid models for general mechanical parts and assemblies; whereas, STEP AP214 is more focused on automotive I industry. The recent STEP AP242 is a merging of AP203 and AP214, and it exceeds their abilities by not only extending the covered industry areas, for it also exports extensive information about the part. However, these APs share the same hierarchical structure as will be demonstrated in chapter three. The main five categories of the AFR are:

1. Syntactic Pattern Recognition;
2. Graph Based;
3. Hint Based;
4. Logic Rule Based;
5. Artificial Neural Network [41, 42].

Whilst AFR methods have been used widely in the intelligent manufacturing environment, each method still has drawbacks, such as limited ability of learning, specific range of features to be recognized, low speed, etc. [35].

### 2.2.1.1 Syntactic Pattern Recognition

*Syntactic pattern recognition* (SPR) is a formalised technique to represent a pattern as a structure of the form of string, a tree or a graph. In a recognition process, a pattern is analysed and assigned to a predefined class of features. If the pattern is complex, it is defined in a hierarchical way, whereby

primitives are used at the bottom of the hierarchy in order to build simple substructures with symbols [43]. There are three major components in this approach: Pre-processing, Pattern description or representation and Syntax analysis. In the pre-processing, a pattern is coded using a convenient form to be prepared for further processing. The coded pattern could be represented as a language-like code, such as a string of alphabets. During the pattern representation process, the pattern is segmented into simpler sub-patterns called primitives and, the segmentation process helps to identify relations between these primitives. The decision on whether or not the primitive belongs to the class that is described by an existing grammar will be achieved by the Syntax Analyser [44]. Figure 2.1 shows the outlines of this approach, and how an analogy is drawn between the structure of patterns and the syntax of languages.

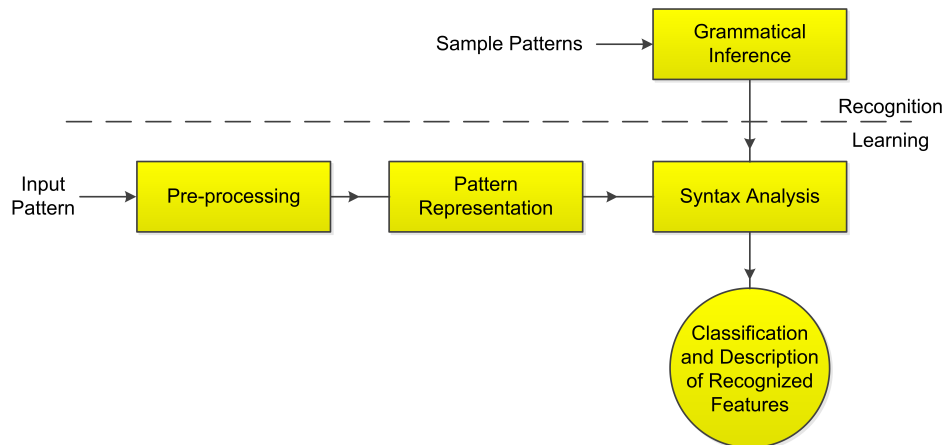


Figure 2.1 Block diagram of a SPR system [44]

Yip-Hoi et al. [45] utilized 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 gouge 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 MTVS decomposition and each predefined feature pattern. After finishing the features recognition, three steps of precedence generation are included: determining the features 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.

Ketan and Yaqoub [46] 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 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 Autodesk 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.

The SPR technique has been also used for recognising features of prismatic parts. Perng et al. [47] proposed a SPR method to extract manufacturing features from 3D constructive solid geometry (CSG). Whilst the system is able to recognise 18 prismatic features, it cannot handle intersecting ones. However, the output feature information from this algorithm can be passed to CAPP systems for the rough-cut machining. Meeran and Pratt [48] 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. Also, Arivazhagan et al. [49] developed a feature recognition methodology that uses STEP files to extract geometrical information directly from a B-rep model. The proposed approach implements SPR to recognise five classes of prismatic parts features: interacting, tapering, interacting-tapering, curved base features, and tapering cross-sections. Each edge loop



of a prismatic part includes details of edges, vertices, coordinate points, and directions. With all the implicit information of an edge loop, a basic feature can be described by comparing it with pattern strings, which have been developed for every class of features. The final shape of a feature is recognised by checking: the presence of similar edge loops on parallel faces and the connectivity of faces between these parallel edge loops. The system is able to recognise 195 types of tapered features. Furthermore, the output provides: the tool path direction, details about edge loops, and the dimensions of features, including the value of taper angles.

Whilst the Syntactic Pattern Recognition method has succeeded in identifying classes of features, it has a narrow area of application. That is, the method is limited to turning features of rotational parts and 2D prismatic parts. The implementation of this method lacks success, if it is implemented with rotational parts that have non-turning features or 3D parts with non-axis symmetry. Because of this weakness, it has been replaced by newer techniques, such as logic rule based and artificial neural network that have overcome these limitations [43, 50].

### 2.2.1.2 Graph Based

The first *Graph Based* (GB) feature recognition formal approach was developed by Joshi and Chang in 1988. It is highly regarded as a successful features recognition method, since it has been implemented in many commercial FR systems. Moreover, many techniques have been incorporated based on this approach and used in different applications, such as process planning software [50]. Joshi and Chang [51] developed the GB approach to recognise machined features from a B-rep solid model. Their approach uses an *Attributed Adjacency Graph* (AAG), which can be defined as a graph  $G = (N, A, T)$ , where:

$N$  = a set of nodes,

$A$  = a set of arcs (or angles);

$T$  = a set of attributes to arcs in  $A$ .

The AAG method assumes that a unique node  $N$  exists at each face of a part. Also, for each two adjacent faces, there is a unique arc  $A$  that connects their nodes. Finally, each arc is assigned an attribute  $T$ , which is usually a concave angle ( $T = 0$ ), or a convex angle ( $T = 1$ ). Figure 2.2 (a & b) shows an example of the AAG for a part. The part has 14 faces, and each face is represented by a

unique node in the AAG net. After that, a line is drawn between each two adjacent faces and indicated either by "0" or "1" to refer to concave and convex angles, respectively.

Figure 2.3 shows three different parts that have identical topological information and AAG representation. Hence, the features need to be accurately defined, and that involves identifying the minimal set of essential conditions that classify a feature uniquely. In this example, the value of the two angles formed by the slot faces specifies the difference between the definitions of the three slots [51].

At that level, the use of AAGs to recognise features was limited to polyhedral parts, which have flat faces, straight edges and sharp corners as isolated features. However, the concept could be extended to identify features formed by both planer and cylindrical faces [51]. Whilst the number of primitive features is countable, the configurations when two or more features intersected are unlimited [52, 53].

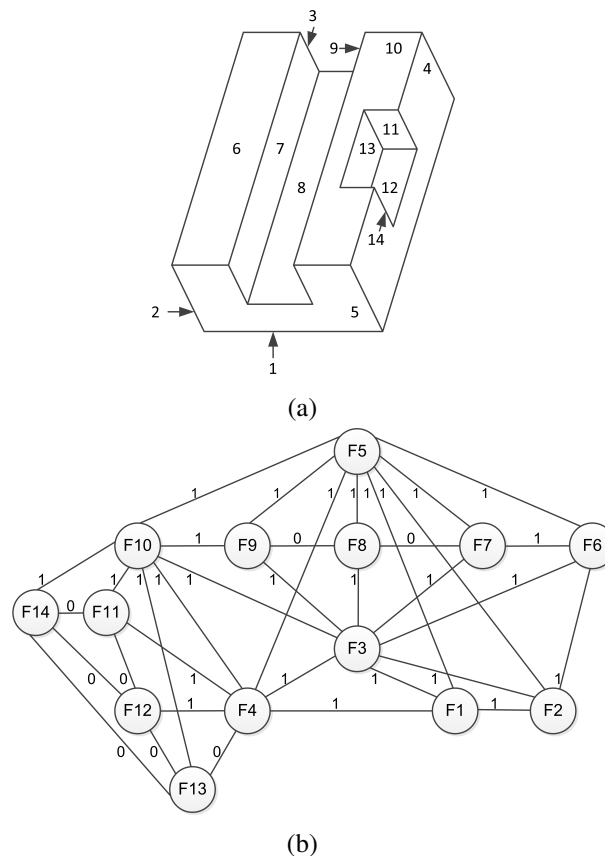


Figure 2.2 An example of a 3D prismatic part (a) 14 faces of the prismatic part, and (b) explanation of AAG for the part in (a) [51]

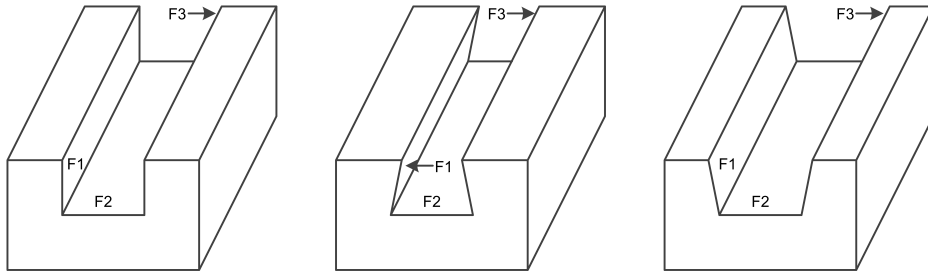


Figure 2.3 Three different parts with identical AAG representation [51]

The difficulty of recognising interacted features is one of the major issues in a GB features recognition system. Consider the example part in figure 2.4, theoretically, two slots are recognised by applying the initial concepts of GB systems, and these are  $(f1, f2, f3)$  and  $(f5, f6, f7)$ . But practically, there is a third slot  $(f1, f4, f7)$ , which intersects with the other two. This example reveals the shortcoming of the unmodified graph-based systems [53]. To overcome this issue, Marefat and Kashyap [52] presented a novel solution that includes simplifying and restoring the AAG representation by creating a conceptual face, which is the a result of unifying two or more unifiable faces. For example, in figure 2.4, a slot with  $f1, f4$  and  $f7$  can be recognised after restoring  $f2, f3, f5$  and  $f6$ . However, the system cannot work correctly, if the AAG representation of a primitive feature consists of several disconnected components [52, 53].

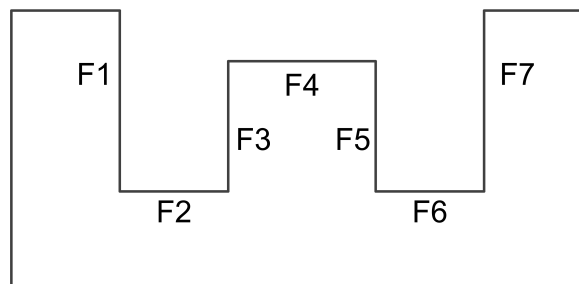


Figure 2.4 An example of intersecting features [53]

In a multi-tasking machine, both turning and milling machining can be performed. ZHU et al. [54] 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 features 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.

According to Woolridge [55], an agent is “a computer system that is situated in some environment and that is capable of autonomous action in this environment in order to meet its design objectives”. Fougères and Ostrosi [56] have used the intelligent agents concept to enhance the graph based method in order to recognise a set of features in prismatic parts. Topological and geometrical information of features is represented by adopting two formalisms, namely, graphs and grammar. A feature is transformed into an agent, with sub-features forming a network of agents that communicate and share knowledge with each other. The feature recognition approach performs three stages that are carried out by the multi-agent system and assisted by the designer: (1) identifying possible areas where features may be created, called regioning; (2) building links and virtual faces, called virtual extension; and finally, (3) feature identification includes identifying features in these zones. Currently the approach is limited to manufacturing features. However, there is potential to extend the approach of the multi-agent system to update knowledge continuously through new rules that are learnt and hence, the set of recognised features could be extended.

### 2.2.1.3 Hint Based

Initially, each proposed feature has a frame, which could be represented as a *hint*. A *hint* is a proof about the existence of a specific machining feature in a part. For example, primitive linear slots or grooves are produced by a pair of parallel opposing faces, whereas hole hints are formed by either cylindrical or conical face and one can also be triggered by a thread attribute [57, 58].

Vandenbrande and Requicha were the first to present the hint-based approach in 1993 [53, 59], proposing an Object-Oriented Feature Finder system (OOFF). It starts by generating hints for the existence of features, and posting them on a blackboard. Then, the hints are analysed by a hint classifier and the results placed in one of three categories: Promising, Unpromising and Rejected hints. If the hints seem promising, they are processed by a feature completer to confirm them as actual features and if they are evaluated as rejected, then they are neglected. Otherwise, if the features are

unpromising, they are temporarily stored, because they may be reactivated later to be identified as part of a composite feature.

The algorithm also includes the implementation of another set of rules in an attempt to combine the stored processed features and check the possibility of getting more complex features. Finally, verification is undertaken, since it is important to check the machinability of the proposed features.

Figure 2.5 shows the architecture of the hint-based approach [57].

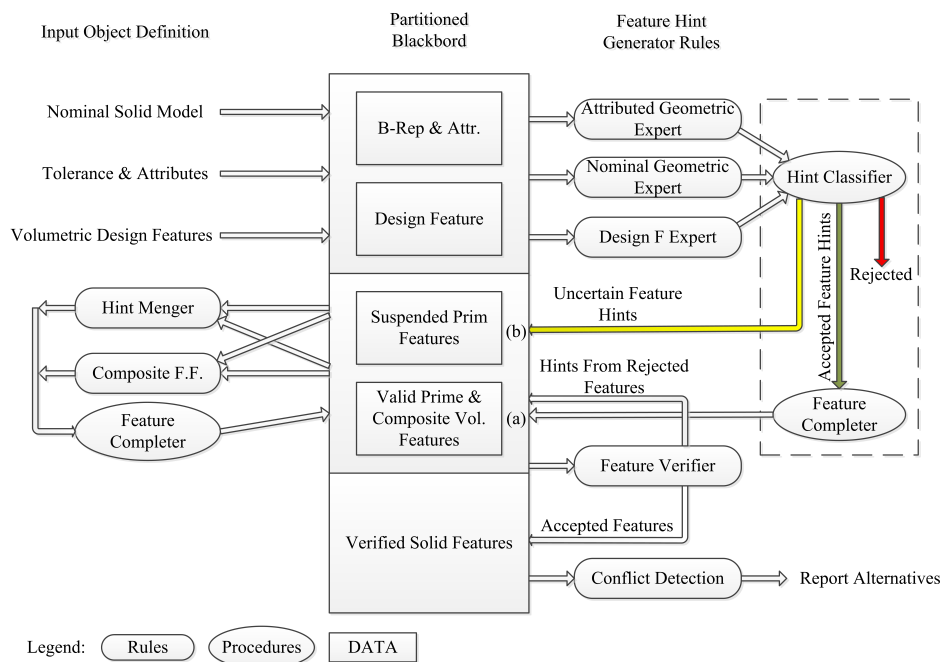


Figure 2.5 Architecture of the Hint-Based approach [57]

A feature hint may be generated from different sources, for example: nominal geometry, design feature, tolerance or surface attributes to infer parts machining features and process planning aspects.

Kang et al. [28] 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 file 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.

#### 2.2.1.4 Logic Rule Based

In 1984, the logic rule based approach for feature recognition was introduced by Henderson and Anderson, being presented as a FEATURES system. The system replaces the human role with logic programming in the part interpretation to extract high-level knowledge from a stored part description. The FEATURES system consists of three main components: feature recogniser, feature extractor, and feature organiser as shown in figure 2.6. The feature recogniser can define cavities made up of various types of features, such as pockets, holes, and slots, whereas, the feature extractor can separate these features based on their respective entities. Finally, the feature organiser arranges the separated features as a graph structure, which is defined as volumes to be removed. The structure of the graph consists of nodes for the features and links for their relationships. Moreover, the feature graph can be used in subsequent manufacturing processes, since it provides information about each feature as well as the stock material [60].

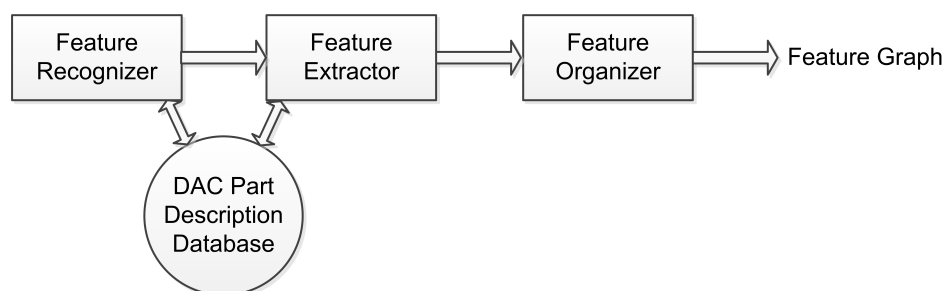


Figure 2.6 Architecture of the FEATURES system [60]

Sivakumar and Dhanalakshmi [61] 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.

Oussama et al. [62] developed a new methodology for recognising both interacting and isolated features for rotational parts. The system consists of three main modules: geometrical and topological data extraction, feature recognition, and feature generator. The first module includes the extraction of dimensional, topological and geometric of the part features from a STEP file using C++ programming. The extracted information is reorganised and stored in a database. The geometrical and topological data is analysed in the second module using a rule-based approach. Also, a library of thirty predefined turning manufacturing features is associated in this module in order to recognise high-level features. In the third module, there is a feature generator analysis of the recognised features to generate all possible combinations of interacting features. However, multiple combinations of interacting features are given to machine the same workpiece, which can lead to complex computational processes in the feature recognition stage and is time consuming during the tool selection stage.

#### **2.2.1.5 Artificial Neural Network**

Artificial neural networks (ANNs) are human-like computational models inspired by the biological neural networks (NN) of the brain. An ANN is a network of nodes and links, which is specified by: net topology, the characteristics of each node, and learning rules. The net topology assigns the inputs of each node, whereas the node characteristics determine its output. A node is defined by three factors: inputs, an arithmetic operation, and a weighting. Each node can receive several inputs and perform arithmetic operations. Then, this node sends only one output via link(s) to another node(s) which perform operations in turn. Both inputs and weights of a node are allowed to change over time, whilst the learning rules train the NN how to react to an unknown input. An important characteristic of ANN systems is their ability to adapt and learn by collecting examples. Thus, they can be trained to solve feature recognition tasks by repeatedly presenting input patterns to the net. The use of ANNs to recognise manufacturing features from 3D solid models has been demonstrated since 1990 [50, 63].

Prabhakar and Henderson [64] presented a new algorithm for feature recognition based on an NN technique. The methodology involves coding of a B-rep solid model into an adjacency matrix, which is a 2D array of integer vectors, and contains eight-digit code for all faces in a part. An entry of the adjacency matrix is a vector of integers. Each vector gives information about a face and defines its relationship to another face. Then, the developed algorithm performs pattern recognition on each row of the matrix. A feature is recognised if the recogniser detects the presence of a predefined set of rules belonging to a specific feature. The developed NN has five-layers, each performing one of the steps in the algorithm. Whilst the approach recognized simple features such as holes, slots, and pockets, many limitations were found, which are discussed in [64].

Sunil and Pande [65] 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 standard ACIS text (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.

An important constraint must be kept in mind, which is that an ANN performs only simple arithmetic operations and is not expected to perform any logical operations explicitly. This limitation goes against the development of conventional feature recogniser systems, which check to see whether a potential feature satisfies a predefined set of rules [64].

#### **2.2.1.6 Hybrid AFR systems**

A hybrid AFR method was developed by Rameshbabu and Shunmugam [66] 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. [67] 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 features 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 [68] 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 latter. Finally, two approaches are used to recognize 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. [69] utilized the concept of AAG to analyze 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 features recognition techniques that serve different manufacturing processes in addition to those of machining. For example, Shi et al. [70] presented a features 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. While 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 [23].

### **2.2.2 Geometric dimensions and tolerances**

The initial design is a critical stage of the product life cycle. That is, decisions at this stage provide a rich quantity of essential information for the remainder of the design and manufacturing cycle. Identifying feature dimensions and tolerances of a part design is one of these essential decisions [71]. In an engineering drawing, a dimension is a natural descriptor or a numerical value of the geometry [71, 72], whilst a dimensioning scheme is a set of dimensions, which is chosen by a designer and shows the nominal geometry of a part [73]. However, the dimensions of a real part cannot be produced exactly as designed under normal manufacturing conditions. Thus, tolerances

are needed to accommodate variations in dimensions, which are the permitted range of deviation in part geometry, according to the nominal size and shape [73, 74]. Traditionally, the designers specify dimensions and tolerances just before releasing the drawings. According to the reviewed literatures, these tolerances are based on standards that have emerged from engineering experience, best guess or through anticipated manufacturing capability [73, 75].

Although most current CAD systems have the facility of adding geometric dimensioning and tolerancing (GD&T) on 3D models at the presentation level, their internal translators do not yet convert such information into a semantic representation. This means CAD systems do not take advantage of STEP AP242 characteristic that is saving and translating GD&T data. Therefore, researchers have been working on developing different solutions in which GD&T data is interpreted from CAD systems and provided to CAPP ones in an efficient and automatic way. Well-known examples of interpretation models are: offsetting, parametric, vector equation, variational surface, kinematic, generalized interval, degree of freedom (DOF), and tolerance-map models [76]. This section illustrates some of these models.

Computer-aided analysis of dimensions and tolerances appeared for the first time in the 1980s [72]. Hoffmann [77] suggested the use of linear programming in order to solve basic problems of tolerance allocation in mechanical parts. A method was presented to check inaccuracies of operations by finding the tolerance between two elements for an individual part, where an element can be a point or a line. Later, Weill and Bourdet [74] developed a computer program that calculates dimensioning and tolerancing, thereby introducing the aid of computers in process planning. A two dimensional drawing of a part, in three different directions, was used as an input to the system. The output provides the following: tolerances of positions, machining tolerances, clear distinctions between dependent and independent variables and minimum dimensions of the raw material. Britton and Thimm [78] presented a new matrix method based on the datum hierarchy tree technique, to calculate functional dimensions and offsets for tolerance charts. The calculations in the matrix method are performed either manually or are aided by a computer. It has been implemented in a prototype AI program for process planning.

Shen et al. [79] proposed a semantic GD&T model which can be used with many tolerance analysis methods, such as min/max chart, Monte Carlo simulation, and multivariate regions. This

model is a super constraint tolerance feature (SCTF) graph, and it can represent all the tolerance types in the standards. Also, it includes geometric, constraint, assembling, degrees of freedom, and tolerance information that is required for the representation of designed parts in a CAD system. Later, Venkiteswaran [80] presented a system that extracts the semantic GD&T information from the SCTF file and implements it to create a STEP AP 242 file. This is by following three phases: (i) attach GD&T information of a design model to ACIS SAT file in order to generate a SCTF file, (ii) save the design as a STEP AP203, which includes geometric entities, and then merge it with the GD&T information from the first phase, and (iii) convert the resultant file into a STEP AP242 using STEP NC Libraries. According to the authors, the output STEP file was evaluated using available AP 242 parsers and showed full conformance with the STEP standard.

As mentioned, almost all CAD packages have the ability to generate and locate dimensions automatically. However, the way that these packages specify the dimensions does not reflect the functionality of the part and does not satisfy the designer and manufacturer's point of view. At present, there are no CAD packages that can calculate and assign tolerances of different dimensions in a satisfactory way, which means alternative methods (solutions) are required.

### 2.2.3 Materials

From a designer viewpoint, material selection is a process that identifies the material for a design that after appropriate manufacturing meets its required function [81]. The decision of material selection is made during the preliminary stage of design calculations, and it has a great effect on overall product cost [82, 83]. From a process planner perspective, knowing the product material is one of the important aspects that gives early direction regarding the product manufacturing process [84]. This is because the product material affects the selection of cutting tool material and geometry, cutting conditions (feed, speed, depth of cut) and the type of coolant [85]. There are recommended cutting conditions for each material, which are traditionally found in the tool manufacturing catalogues or cutting data handbooks [86]. It takes merely a glance at these machining handbooks to realise they contain complex information about hundreds of thousands of different materials. This makes it very difficult and time consuming for a process planner to identify the ideal cutting tool, cutting conditions

and coolant fluid for each process. Hence, a type of upgradable computerised database is needed to overcome the aforementioned drawbacks.

Al-Shebeeb and Gopalakrishnan [87] used generative CAPP software tools to analyse the effects in process plans with respect to changing material properties. Precisely, the system explored the change that occurs in the cost and production rate of process plans. The research showed how a design can be evaluated from the manufacturing perspective in an attempt to decrease production cost, increase production rate and improve the quality.

#### **2.2.4 Surface finishes**

Surface finish is one of the essential factors that affect the planning of manufacturing, inspection and assembly processes of a mechanical component [88]. Its notations are used to define the surface quality of a designed part, which is usually specified by maximum allowable surface roughness in micrometres, as Ra or Rz numbers [89]. As an important consideration, surface finish must be supplied by the designers as it reflects intended functions of the part, and must be known by process planners, for this helps to predict the machining performance of any machining operation [88, 90]. Some researchers have focused on the control of cutting conditions to obtain a specific surface finish [91, 92]. However, other groups of researchers have claimed that in addition to the cutting conditions, other factors such as the tool geometry, tool material, process capabilities and workpiece material properties could affect the surface finish value [93]. As a result, it is very important to ensure that from among all the alternative machines and tools in the shop, the machine, the tool design and the tool material assigned to perform an operation are the best choices [4, 94].

(Ben) Wang and Wysk [95] and Chang and Wysk [96] presented the Turbo-CAPP and TIPPS systems, respectively and whilst both of these systems consider the surface finish value in their processing, the input method needs human intervention. Furthermore, the surface finish was isolated from the other inputs. However, systems like the ones mentioned above have paved the way for more automated process planning systems. Chang et al. [97] proposed a variant CAPP system. In this method, a workpiece is represented by geometry, material, and precision indexing. The surface finish is categorised into three groups: fine, intermediate, and rough. If two parts have the same geometry and their surface finishes are located in the same group, it means the cutting processes and machine

tools are the same for both. According to the authors, the system is able to find out the most relevant case rapidly and is convenient for manufacturing products that vary little in their process planning.

Grzesik and Wanat [98] presented experimental work aimed at finding alternative process planning that produces the required surface finish. A workpiece of hardened low-chromium alloy steel was used and instead of deploying grinding, as the final process in a traditional process plan, mixed Alumina-Titanium Carbide (TiC) ceramic cutting tools were utilised in turning machine to get the required value of surface finish. This alternative process plan reduced the manufacturing costs, decreased the production time and improved overall product quality.

Özel et al. [94] examined the use of neural network (NN) models to predict the surface finish and tool wear in finish hard turning processes. The study involved investigating the finish turning of AISI D2 steels (60 HRC) with the use of ceramic wiper (multi-radii) tools. Three factors were utilised in order to train the NN algorithm: measured forces, power, and specific forces. According to the authors, the NN models are able to predict the surface finish of a part for a range of cutting conditions. Furthermore, it can enhance the creation of intelligent process planning (IPP).

### **2.2.5 Machining process and process capabilities MP&PC**

In order to make a decision on which process(es) can be used and in which sequence to manufacture a part, the characteristics of available manufacturing processes should be known. The basic characteristics of a manufacturing process include features that the process can produce, limitations of the dimensional and geometrical tolerances (process tolerances), and the achievable surface finish. Such characteristics are called process capability, which must be matched with the required geometrical and topological specifications of a part to identify the necessary manufacturing processes [96, 99]. For example, depending on the required surface finish of a hole, a process planner should know that a reaming operation can produce a surface finish within the range 2.5-0.4  $\mu\text{m}$ , whereas a drilling operation can guarantee accuracy between 80 to 12.5  $\mu\text{m}$ . Also, a drilling operation is required before a boring operation due to the limitation of the cutting tool accessibility. Thus, it is possible and necessary to utilise the knowledge about the process capabilities as a functional model of process planning [100].

The knowledge about process capability can be divided into three levels: universal, shop and machine level. The universal level ignores the knowledge of a specific machine shop, and its information is available in handbooks. This is to be used when the details of a specific machine shop are not available. In the shop level, a specific set of machines and/or cutters is considered in order to predict the required accuracy. The machine level provides the most accurate information since it takes in consideration only certain machine capabilities. Šormaz et al. [101] proposed a rule-based intelligent process planning system, which includes the process selection based on the universal level information of process capabilities. The process selection (IMPlanner) algorithm starts by comparing the feature requirements with the stored process capabilities. The process is selected based on inheriting relations from the feature type. The algorithm also includes an evaluation process, which has one of three possible results: complete, partial, and no matching. Depending on the feature requirements, process capabilities, and the evaluation result, the system shows one of these outputs:

1. "complete" means the process is selected as a full compatible process for that feature;
2. "partial" means the process is accepted to produce an intermediate feature with processes in the next stage of precedence;
3. "no matching" means the process is rejected and the system searches for an alternative process;
4. If the result is "no matching" and there are no more processes to consider, then the system reports dead-end to the user.

### 2.2.6 Summary

While CAPP systems have been developed over the years to consider all the necessary inputs, some have been given greater emphasis, whereas others have been somewhat neglected. Older CAPP systems would deal with inputs individually, which impacted negatively on the time, cost, and quality of products. Whilst CAPP systems nowadays attempt to deal with the inputs simultaneously, there is still a lack of smartness in the way these systems update inputs. For example, AFR systems are limited to recognise a set of predefined features. Also, CAPP systems are not able to determine cutting conditions for new materials. These fundamental limitations are currently shaping a vital area of research within smart CAPP that requires further development.

## **2.3 Output activities of CAPP**

Whilst the automation of inputs in CAPP has significant value, process planners are continually looking for what such a system can give as an output. As was mentioned previously, the output of a CAPP system should include selection of operations, sequence of operations and associated processes to convert a raw material into a product. Also, it has to specify each dimension and geometry requirement through which the operation is achieved [102]. Removing human intervention to achieve these tasks would improve CAPP performance. In order to create a convenient CAPP, these requests of production should be considered: reducing cost, quality control, and shortening time taken [103]. This section studies what has been achieved in each output parameter of CAPP so far.

### **2.3.1 Process selection**

To manufacture a product with the required shape, size and properties, this depends not only on the design, but also on the selection of an appropriate manufacturing process(es) [104]. Process selection in CAPP is making the decision of which to use to manufacture a part, whilst taking into consideration the existence of alternative processes that are able to achieve the same task. Consequently, a part that is planned to be manufactured on a specific class of machines, could be also manufactured on a different class of machines. However, using a different class of machines means significantly different manufacturing procedures and therefore, different process planning [17].

Jain and Jain [104] presented a process selection methodology for advanced machining processes (AMPs) as well as basic types of manufacturing process. A combination of elimination and ranking strategy is used in this methodology, whilst the AMPs were reclassified to facilitate the process selection. Software known as APSPOAMPS is used to implement this approach. According to the authors, the users of AMPs are able to save time, effort and money by using this system. However, AMPs were only reclassified based on their material application capabilities, feature recognition generating capabilities, and operational capabilities, in addition to economic and environmental aspects. Hence, the authors confessed that this is not an exhaustive list, due to unavailability of other quantitative information about the manufacturing process. In consequence, the described methodology is considered preliminary and not comprehensive or complete.



Lau et al. [105] applied a rule-based method to extract the data of a part from a CAD model saved as a STEP file. The system is aimed at recognising the features of a part and creating a process plan to manufacture it. By applying user-defined rules, the system provides rapid process selection. Even though the system saves time in process planning, it is limited to prismatic components and the common processes that are used to produce them. Zhang and Merchan [106] proposed an integrated process planning model (IPPM), with three levels: preplanning, pairing planning, and final planning. The process selection is a step in the preplanning level. The machining process is selected based on characteristics of a part's features and the equipment on the shop-floor. The main concern of the process selection with this system is the availability of machines on the shop-floor.

### 2.3.2 Sequence of operations

The sequence of operations is a primary objective of most CAPP systems [17]. Many factors can affect this task such as types and numbers of available machines, available tools and fixtures, part features, tolerances and required heat treatment [107]. Each of these factors imposes limitations or constraints on the sequence of operations, which are called manufacturing constraints [107, 108]. Moreover, some of the constraints may cause confusion as they conflict with each other. Hence, some processing steps are executed in the same setup, whereas others are executed consecutively [108].

Kruth and Detand [109] presented non-linear process plans (NLPPs) for a CAPP system. The NLPPs provide rescheduling and re-planning functions which are missing in conventional process plans. The objective of NLPPs is to offer manufacturing alternatives. In this method, feature based input is required to generate a sequence of operations. The data are provided to the system via two ways, namely, as a neutral file or a graphical editor. GU and ZHANG [107] used an object-oriented CAPP system in order to generate a sequence of operations. The method includes three phases of planning. The first phase is initial planning, which includes the selection of the operations and the machine cells. Then, a set-up planning identifies machines and fixtures, clamping surfaces and feature accessibility. Based on the information from the previous phases, the final planning determines all the detailed sequences of operations.

Pandey et al. [110] developed an operation sequencing rating index (OSRI) system by associating four factors: setup changeover, tool changeover, motion continuity, and loose precedence. The next

step includes the use of the simulated annealing (SA) algorithm to determine the best sequence of operation by maximising OSRI. According to the authors, the system is able to reduce both the computational time and the search space to reach the best solution. However, it does not take into consideration the stuck up tolerances.

### 2.3.3 Cutting tools

A machining process includes the material removing from a workpiece as swarf or chips using single or multi-point cutting tools with a specified geometry[93]. Despite the cutting tools selection being just a sub-function of process planning, it is a complex task that requires considerable experience and knowledge [111, 112]. Traditionally, the best set of cutting tools is selected by individuals based on their previous experience, but this manual approach is slow as well as leading to errors and inconsistencies [112]. Thus simply, it is inconvenient to select the cutting tools based on the familiarity, experience and the memory of operators. To overcome this issue, a system is required to analyse part geometry and specifications, subsequently identifying the appropriate tools automatically. Unfortunately, despite the development of CAPP systems, automatic cutting tools selection for machining operations is still in its infancy [111].

The tool selection affects many other parameters in manufacturing such as the selection of fixtures, production rate, machining accuracy and the final cost of the product [113]. The correct selection of cutting tools and its associated cutting conditions could significantly reduce the production cost. Hence, any CAPP system that does not consider these parameters will have severe limitations [114]. Even though most modules use "minimum cost" criteria in the tools selection task, this criteria does not necessarily represent the ideal solution as it does not always consider other technological constraints. In such situations, it is difficult to include machinists' experience of decision making, as these systems are built up based on "low cost" criteria. So far, there is no general acceptable solution for tools selection that could be used in all workshops [111].

Ribeiro and Coppini [114] presented a new algorithm to improve the computer aided technical assistance (CATA) system. Previously, this was used to determine operational costs of machining processes. By adding the improvement, the system is able to choose cutting tools and cutting conditions. This is based on a database to determine maximum production. Arezoo et al. [111]

developed an Expert Computer Aided Tool Selection System (EXCATS) by using the Prolog language. The objective is to select tool-holder, insert and cutting conditions in turning operations. The system has many features, for example, it allows the user to modify the result by feeding his or her own shop floor experience. In addition, it has an interface that enables tools manufacturers to add their tooling system to the package. The required inputs to the system are the part representation and tools file. Later, Zhao et al. also [115] presented a novel method to integrate a CAD system with EXCATS in turning operations. The CADEXCATS system starts with the use of an IGES neutral format to save product data. Thereafter, a feature recognition approach is applied to generate a component representation file for EXCATS.

Fernandes and Raja [113] proposed Incorporated Tool Selection Systems (ITSS) in a CIM environment. There are five steps carried out in ITSS to select a set of tools for each feature in the product, as processed on a specific machine. The first step is to define possible alternative tools for each feature. Then, the system excludes some of the tools from step one that are not compatible with the selected machines. The next step includes further eliminating the tools that are not compatible with the job, part material, or feature attributes. In step four, the determination of tooling parameters for each tool type is achieved by using the available information about the part. Finally, a tooling match is determined when the system searches through an object-oriented database. According to the authors, the system is fast and helps to keep the user up-to-date with developments in the tooling industry.

Based on the use of mathematical modules and heuristic data, Edalew et al. [116] developed a dynamic programming-based system to select cutting tools and calculate total component cost. This is to give users suggestions and solutions to reduce cost and time of machining. The system was developed using Kappa-PC software with five modules: knowledge acquisition, knowledge base, inference engine, user interface, and database. The system was designed to cover different machining processes, whereby it is able to analyse cylindrical and prismatic products. All the calculations are based on the product material, features attributes, machining time and cost, tool life and material removal rate.

Orala and Cakir [112] presented a tool selection method used in a generative CAPP system for rotational parts (GPPS-RotP). In this system, the automatic tools selection task is based on several

factors: part machinability, part features, machine tool data, part holding device, and set-up number. There are two criteria that have been implemented to achieve tool sequence that minimise both the number of tool changes and the tool travel time.

#### **2.3.4 Cutting conditions**

Cutting conditions or cutting parameters include cutting depth, feed rate and cutting speed. The selection of convenient cutting conditions is an essential task for every machining process and CAPP system. Usually, experiences and handbooks are used to determine the desired cutting conditions. However, this does not necessarily mean that the selected parameters will achieve the desired surface quality features [117–119]. In order to select proper cutting conditions, reliable mathematical models, which are usually based on neural computing or statistical regression techniques, have been developed to study the relationship between cutting conditions and cutting performance. The next step includes the defining of an objective function with constraints to find the optimal cutting conditions [117, 120].

Based on the design and analysis of machining experiments, Chua et al. [120] developed mathematical models for Röchling T4 medium carbon steel workpiece material using TiN-coated carbide as a cutting tool in the turning operation. The experiment was planned using three levels (low, medium, high) for each cutting parameter. The results of 27 experiments were recorded, each of which was carried out with a combination of different levels of parameters. The tool life, cutting forces and power consumption were measured and related to the cutting conditions. Vele [121] presented a mathematical model to select optimal cutting conditions for TiN-coated carbide cutting tools and a T4 medium carbon steel workpiece, in order to achieve economic objectives. The system was developed based on detailed planning and proper analysis of machining experiments for multi pass turning operations. The research was also aimed at studying the effects of cutting parameters on the tool life, cutting forces and power consumption. Despite the mathematical models approach helping to determine desired cutting conditions, it requires considerable knowledge and experience. Furthermore, to build mathematical models it is necessary to carry out and analyse a large number of cutting experiments, which consumes material and time. Hence, a more cogent approach with less consumption is needed.

Subramanyam and Rao [122] analysed the values of three cutting parameters (speed, feed and depth of cut) of 14 turning operation samples with their respective surface roughness. The values obtained by varying the parameters, are used in the design of expect V-8 software in order to obtain an equation. There are two phases in this study, which include the use of a genetic algorithm (GA) and Particle Swarm Optimisation (PSO). In the first phase, the experimental values were used instead of mathematical models to create a neural network. Then, through MATLAB, a GA is used to optimise the cutting conditions. In the second phase, the program calculates the cutting parameters by using the obtained equation from the design of the Expect software and written PSO in c language. Finally, the results from the GA and PSO are compared.

In 2006, Sardinas et al. [118] presented a multi-objective method to optimise the cutting parameters in turning processes based on posteriori techniques and using a GA. In this system, two conflicting objectives were considered as optimisation criteria and simultaneously optimised: tool life and operation time. The system starts with the use of a micro-GA in order to create and preserve an elitist population of the fittest cutting parameters: speed, feed rate, and depth of cutting. After applying a series of evolutionary periods, non-dominated points are plotted to build the Pareto front in order to make the analysis and decision-making process easier. Later, Sardinas et al. [123] applied the same earlier techniques (posteriori and GA) and procedure to optimise the cutting parameters for drilling laminate composite materials. In this work, two conflicting objectives were considered and optimised: material removal rate and delamination factor. The authors confirmed that the use of posteriori and GA in different machining processes increases the flexibility in selecting the optimal cutting parameters.

Addona and Teti [124] proposed an optimisation system based on a GA to determine the cutting parameters in turning operations. The main objective of this system is to minimise the production time without affecting the cutting constraints, which in this research include: tool-life, surface finish, cutting force, chip-tool interface temperature, power, roughing and finishing parameter relations, stable cutting region, and the number of passes. According to the authors and based on the simulation model, the system presents a fast and suitable solution for automatic selection of the machining parameters.

Agrawal and Verma [125] used a new evolutionary GA approach in CAPP to optimise machining parameters for multi-tool milling operations. The system was developed based on the Tolouei-Rad and Bidhendi mathematical model, whilst the maximum profit rate was utilised as the objective function. The depth of cut was not included in the problem of determining the machining parameters, but instead, the value of the combination between maximum allowable depth for a given workpiece and cutting tool was taken. Hence, the calculations for selecting the speed and the feed rate of cutting were reduced. In this research, the following were used as constraints: surface finish requirement, maximum machine power, available feed rate and spindle speed on the machine tool, and maximum cutting force permitted by the rigidity of the tool. According to a comparison of the results, the methodology delivers an improvement of 419% in profit rate over the handbook recommendation.

Tsai et al. [126] proposed a methodology to define the cutting condition and tool path for pocket milling operations. In this research, the cutting conditions include axial and radial depth of cut and feed rate, with a procedure being followed to determine these parameters. Firstly, the axial depth of cut is selected based on the geometry of the product. Next, the maximum allowable radial of cut or the maximum allowable engage angle is calculated. Finally, the feed rate is selected for both the rough and finish cutting cycles. In the rough cycle, the cutting torque is considered to calculate the feed rate, whereas this calculation in the finish cycle is based on the tool deflection. The main purpose of determining these cutting conditions is to avoid abnormal cutting states such as excess cutting torque, excess tool deflection and chatter vibration.

### **2.3.5 Selection of jigs and fixtures**

Fixtures and jigs are mechanical devices and tools frequently used with different types of machining, assembling and inspection operations, to maintain accuracy and facilitate production [127, 128]. A fixture is a holding or support device that securely locates and supports the workpiece with respect to a cutting tool or measuring device in manufacturing industry [129, 130]. A fixture system is a set of clamps and locators aimed at removing the degrees of freedom, thereby restricting the part to that particular position [131, 132]. The position and orientation of a workpiece is usually determined by locators, whereas clamps exert a clamping force in order to press the workpiece firmly against the locators [132, 133]. Whilst jigs are used to locate and hold components as well as fixtures, they also

guide the cutting tools and provide repeatability in the manufacturing of products [134]. Traditionally, an appropriate designing of fixtures and jigs scheme is determined by relying on the experience of a tool designer or by using trial-and-error methods. These methods are costly, time-consuming, and not accurate. Thus, many computer-aided fixture designs (CAFDs) have been developed over the past decades to overcome the aforementioned drawbacks [135–137].

Artificial intelligence techniques is one of the methods that has been used in the design and selection of fixtures. Bhattacharyya et al. [135] presented an expert system in order to select the appropriate fixture, according to a particular workpiece specification. In this research, the fixtures are classified into three main categories of elements, which represent part's datums: clamping, positioning and guiding, supporting and base. The system contains many typical search routines built based on the experiences of fixture designers, each routine being linked to a specific group of elements. Then, the parameters of the chosen elements are passed to an interface program calling standard commands from a drafting package to construct the drawing of these elements parametrically. Finally, the fixture is formed by assembling all of the elements together. According to the authors, it is possible to use this procedure not just with the design of jigs and fixtures, for it could be extended to include almost any products where a large number of variations exist.

Wang and Pelinescu [131] proposed an approach to optimise fixture layout for arbitrary complex 3-D workpieces and the fixture elements are restricted to being in discrete locations. The authors claimed that two major issues were addressed in this research: the development of an efficient algorithm to define fixture synthesis and the selection of optimal fixture design based on practical requirements. An interchange algorithm was employed in the system in order to improve the locator locations, which were selected randomly, in a subsequent set of interchange processes. The objective functions include: maximise the accuracy of the workpiece localisation as well as minimising the norm and dispersion of the locator contact forces. Also, the interrelationship between locators and clamps were presented in two different design strategies, which was to evaluate the trade-offs between different performance objectives.

Qin et al. [133] developed a new methodology to analyse and optimise the clamping sequence by taking into account the varying of both contact and friction forces during clamping, which affects the degrees of freedom. A nonlinear mathematical programming problem was solved in order to evaluate

the errors of varying contact forces and workpiece position in each clamping step. To achieve this task, the total complementary energy of the workpiece–fixture system was minimised. Furthermore, a study to reduce the effect of the clamping sequence on low-stiffness workpiece machining accuracy was presented by means of the finite element method (FEM). By taking three examples, the results showed good agreements between the predicted ones and experimental data.

In mass volume production, a large number of similar parts are produced once jigs and fixtures are designed. In this case, these are designed as special-purpose tools for machining and assembling, and they may or may not be utilised in another mission. The main two purposes are: elimination of the need for an additional setup for every workpiece and ensuring that each workpiece is manufactured within the allowable tolerances.

### **2.3.6 Identifying the tool path for both of rough and finish cycles**

The responsibility of preparing the technical machining information (e.g. numerical control NC program, tool sets, design of jigs and fixtures) is laid on the process planner. The NC program consists of tool path (cutter location) and machine tool operating commands, such as cutting conditions [86]. The tool path is a coded instruction, which is represented by a specific command and numerical value, making specific trajectories on the workpiece being processed [138]. Once the features of a designed part are recognised and defined with all the geometrical information, it is possible to determine the cutter tool path. The manual generation of a tool path is considered a bottleneck in a production system since it requires extensive calculations, which makes it time consuming and error prone. Thus, automatic tool path generation is an essential task of an automated manufacturing system [138, 139].

In processes such as turning and milling, the operations are divided into stages: rough, single pass, multi passes and fine cycle. In the rough cycle, the workpiece is machined in incremental layers in order to avoid damage of the tool and/or machine. This is to remove most of the material from the original stock of a particular workpiece to the desired shape and size, which greatly affects the total machining time and partially, the accuracy of the finished product [139, 140]. Whereas in the finish cycle, the workpiece surface is machined smoothly, using line segments to get the finished product with its required shape and accuracy. The finish cycle directly affects the product accuracy in terms of



shape, dimensions and surface roughness [139]. However, to generate an efficient automatic tool path for rough and finish cycles, these requirements must be considered:

- It is applicable for general feature types as canned cycles;
- It is applicable for general surfaces;
- While cutting a specific part of the workpiece, the cutter does not adversely affect any other part (gauge avoidance);
- Cutting efficiency, which refers to saving time while the cutter repeats traversals in the rough cycle, and achieves the required accuracy in the finish cycle [141]. This is accomplished by reducing the number of moves in which no cutting occurs.

In 2013, Sadílek et al. [142] proposed a new methodology for a rough cycle tool path in turning operations. In contrast to the conventional roughing cycles, when the tool machines a constant depth, a variable depth of cut is applied, in order to increase the tool life. Three paths of roughing cycles are included in this research: gradually decreasing the depth of cut, creating conical paths of cut, and using nonlinear methods. By implementing the developed method on a flange using a sintered-carbide cutting tool, the results showed: increasing the durability of the cutting edge by 44%, reducing the total cost for the cutting tools, and decreasing the load in the spindle by 10%. According to the authors, complex programming is required to generate tool paths for the roughing cycle with variable depths of cut. Later, in 2015, Sadílek et al. [143] used the same previous methodology for a rough cycle with variable depths of cut this time and with the cutting forces as an object function. In addition to the aforementioned results, the new experiment showed: the cutting forces are reduced by up to 10.8%, and the tool life can be affected owing to the relation of cutting force components being changed.

Francis et al. [144] presented automated tool path generation for finish machining of freeform surfaces. The free form surfaces contain scallops, which are the amounts of material intentionally left behind. Two scallop height strategies were used in this research in order to compare the optimal tool path method. The first approach is the minimum scallop height (MSH), which includes extra tool passes and results in smoother surfaces with lower machining efficiency. The second approach is maximum scallop height (CSH), when the cutter sweeps in fewer passes compared with the MSH

method, hence less smoothness of the final surface will be gained. However, the cutter contact (CC) points and the tool orientation at those points are known, thus it is possible to determine whether a collision will occur between the tool and the neighbouring surface.

### **2.3.7 Estimate the time and the cost to manufacture the part**

A time-cost estimation process is greatly required for any new product before manufacturing since these factors affect the success of the product [145]. One of the essential tasks in process planning is to send feedback information to assist the designer at an early stage in evaluation of the design features. This does not just include the functional aspects, but also the manufacturability, assimilability, and estimate processing time and cost. Hence, if the designer realises that the initial design's features require expensive tools and/or complex manufacturing processes, it would be logical to try an alternative one [146, 147]. Underestimating will certainly cause a financial loss to the company, whilst overestimating might mean losing the contract or customer goodwill [145]. Consequently, the estimates should be as accurate as possible.

The cost estimation of non-linear process planning is one of the problems under consideration. This includes, for example, taking into account processing alternatives. In 1998, Xirouchakis et al. [148] presented a PP-net (Process Planning net) model, which is an extension to the Petri net model, in order to determine four types of cost in process planning:

- The pure machining cost, which depends on the machining time for a particular machining operation;
- The cost to transmit a part from one machine to another;
- The cost when the setup is changed in one machine;
- The cost of changing tools in one machine.

In this system, a PP-net construct is created, firstly, by including the costs of operations, machine changes, setup changes, and tool changes. Next, Dijkstra's algorithm is used to compute the shortest path of the previous PP-net. This algorithm has an incremental nature and calculates the optimum process planning cost directly, which means there is no need to develop all possible solutions first.

A year later, Xirouchakis et al. [149] applied Petri net in non-linear process planning to estimate the delivery time and cost of a batch of manufactured mechanical parts. In comparison with the previous work, the interleaving of transitions is allowed in the new method. This is to obtain optimal schedules and realistic delivery time estimations. Also, the simple structure of the Petri net was extended to include a two level hierarchy of nets. The top level of the hierarchy represents the system net modelling, which is the job shop layout, including the machines. Each node of the system net can hold one or more token nets, which is the second level of the hierarchy and includes the jobs and setups. The new model construction allows the job shop layout, jobs and setups to be considered as separate objects with unique identity and behaviour.

Gomaa [145] developed a system aimed at minimising the gap between the estimated cost and the actual cost of products. Three models are presented in this work: detailed time-cost estimation (DTCE), machining complexity (MC), and rough-cut time-cost estimation (RTCE). The DTCE module is divided into five phases:

- Feature recognition (FR), using a feature-based description system;
- Sequence planning (SP), to determine a sequence of machining processes and operation;
- Process detailing (PD), which gives details about each machining process and operations, i.e. tools selection and cutting parameters;
- Time estimation (TE), determining the total standard machining time;
- Cost estimation (CE), which determines the total standard machining cost.

All these phases are essential details for process planning. Ten experimental factors in the MC model have been presented and evaluated: workpiece weight, mean outer diameter, machining yield, number of machining surfaces, surface finish, tolerances, metal type, machining features, process type, and quantity required. These factors are important in the group technology and the rough-cut estimation system. Whereas the RTCE module is needed for a rapid request-for-quotation (RFQ) process, design stage, new parts processing, and production schedule. According to the author, the maximum deviation of the estimated time is 12%, whereas that of the estimated cost is 15%.

Ben-Arieh and Li [150] developed a web-based system that links design stations with manufacturing shops in order to provide accurate and fast machining time and cost estimation. Furthermore, the system provides process-planning capabilities and a supplier selection facility. A cost estimation and supplier selection (CESS) module has been developed and connected via the web-based system with the design departments and the manufacturing shops. The designer client needs to: firstly, register with the CESS server and then, a request for quotes (RFQ) form must be filled. Finally, browsing and the evaluation of cost estimation information from different manufacturing shops is received from the CESS module. The CESS module includes three major activities: account management, RFQ management, and process plan/time estimation. The developed CESS system offers several benefits, including finding the right manufacturers and getting the quotations faster. It also means that the suppliers can get the right work depending on their shop capabilities.

Chi [151] developed an application program “Quotation Calculator” in order to calculate the material and manufacturing costs of a new product. The system was developed in Microsoft Excel with partial programming in the Visual Basic editor. In this methodology, product data is extracted from a CAD model and transformed to Microsoft Excel. Whilst the sequence of operation and process planning are done manually, the cost calculation can be automatically implemented based on standard operations and tables for machine data and material costs.

Elgh and Sunnersjö [152] presented a generative process planning and cost estimation (GEPPACE) system. In this method, the information about topology, features, and parameters is extracted from a CAD model and exported to generic process plans for a given class of products. The DBF approach was used to identify features in a CAD model for CAPP. The core of GEPPACE includes renaming of assemblies, parts, features and parameters using strings with predefined positions and nomenclature for classification of objects. The nomenclature system depends on the product nature and company needs. An application program was developed in order to extract and transfer CAD model information. The application program matches the CAD model information with the generic standard CAPP information and worksheets for cost estimation. This allows the designer to view cost effective solutions in accordance with manufacturing restrictions.

Germani et al. G[153] used a knowledge-based system to link automatically the design features with manufacturing operations in order to obtain the estimation of manufacturing cost. To achieve a

robust combination between the manufacturing operations and the design features, two clusters of data were defined: simple modelling features and advanced manufacturing features. A CAD data structure is analysed using the knowledge-based tool to extract the design information that the system needs. Finally, the cost estimation is generated after mapping design and manufacturing features.

### **2.3.8 Summary**

From the literature survey, it has emerged that not all CAPP systems can provide all seven outputs, thus delivering incomplete information about production, and therefore, requiring human input for completion, which introduces variability and subjectivism. Also, the outputs are determined with specified fixed inputs and a CAPP system should be smart enough to update its output calculations based on new changed inputs. For example, if a new feature is detected, the CAPP system should be smart enough to calculate cutter location data, and other cutting conditions. One way of possibly solving this problem is to use neural networks, training the CAPP system to simplify complex features into their initial entities.

## **2.4 Discussion and summary of literature**

Process planning is an essential multi-task concept in the production industry, which links the design process with the manufacturing process and is aimed at providing a whole plan to transfer the idea of a designer into a final physical product. Process planning functions can be divided into two main categories: required inputs and expected outputs. Computers have been used to assist process planning activities. That is, many CAPP systems and technologies have been developed in the hope that they can contribute significantly towards time and cost reduction. However, the use of computers in process planning does not necessarily mean that these activities are being achieved automatically. A new version of CAPP, which is ACAPP, is needed in order to obtain a rapid, accurate, and robust system with minimum human intervention.

Based on a comprehensive survey, the shortcomings and barriers in current CAPP systems have been detailed, which prevent the creation of effective smart ACAPP systems. Outlined below some of the main facts that have been drawn out:

- Process planning is multidisciplinary. This means, each function of input and output can affect and be affected by the other functions. However, most of the current CAPP systems isolate these functions from each other. Thus, the final decision is far from being optimal.
- Part features recognition is the first and most important input to any CAPP system. Each current AFR and DBF system is limited to recognise a specific set of predefined features. To overcome this drawback, a new feature recognition system is needed, which can automatically recognise new types of features, or at least be able to learn how to recognise a new feature. Also, the system should be able to solve intelligently feature recognition issues, such as features intersection.
- Some CAPP systems include databases that are necessary in the decision making for some CAPP functions. This type of database needs to be updated, otherwise, a lack of information could cause a failure in the CAPP system. However, this issue can be minimised by replacing the database with an intelligent knowledgebase.

All the topics mentioned above could be considered essential issues in terms of creating an effective smart ACAPP system. However, many other considerations must be borne in mind to achieve this task in a satisfactory way. One of the remarkable points in this survey is the use of STEP file in developing AFR systems and consequently CAPP ones. The STEP file is a very wide standard and includes many parts and protocols. Although it should be a well-known format, many researchers do not have enough information about its details; hence, they make mistakes in developing systems based on misconception. This leads to the demand of chapter three, which explains the STEP file structure and outlines its characteristics.

The above conclusions emphasise the need for the creation of a smart AFR system. This will be a major building block for a smart ACAPP that is able to recognise and construct new features using self-learning mechanisms. Such a system takes advantage of the distinctive features of smart process planning that will allow greater flexibility, integration, and automation. Further developments in this area will strongly benefit from the recent developments in artificial intelligence and smart manufacturing systems. The significant by-product of ACAPP is that the total cost of manufacturing decreases, and less resources will be consumed, consequently this supports efforts in sustainability.

## **Chapter 3**

# **STEP file structure and characteristics**

### **3.1 Introduction**

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 [154, 40]. For this purpose, a part CAD representation is saved and exported using one of the international 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 [155]. 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 lifecycle. 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 [155]. However, many researchers have

made mistakes in developing AFR systems based on misconception about details of STEP files. This chapter briefly explains the structure of a STEP file as well as few essential details.

## 3.2 Background

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 [13]. 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 [156], with AP203 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 [157]. Most of these APs use the EXPRESS language to describe the representation of the data, which is transformed into a physical text file [13].

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

The STEP AP203 file is a language-based text file, containing strings and entities describing a product's representation [158], being divided into two main sections: "HEADER" and "DATA". The HEADER section provides general product information, such as file name, creation date, company name, programmer name, etc. 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 figure 3.1. 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 figure 3.2a is considered a "CLOSED\_SHELL". Figure 3.3 shows a part of the STEP AP203 file of this cylinder, which contains 253 entity lines in its DATA section (appendix A). 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. For instance, by scanning the STEP file of the cylinder in figure 3.2a, The “CLOSED\_SHELL” is found in line #13:

#13 = CLOSED\_SHELL ( 'NONE', ( #47, #23, #123, #198 ) );

where #47, #23, #123, and #198 represent the faces that form the cylinder.

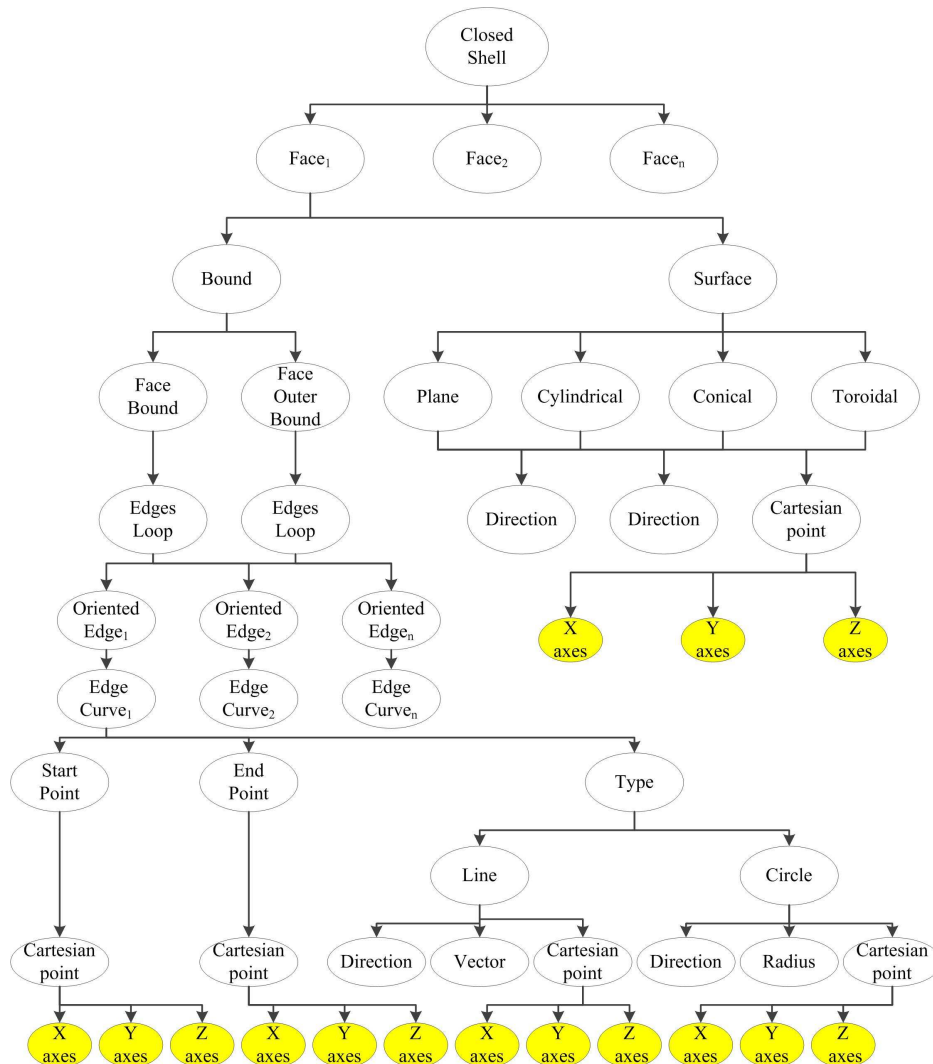


Figure 3.1 The Structure of a STEP AP 203 file

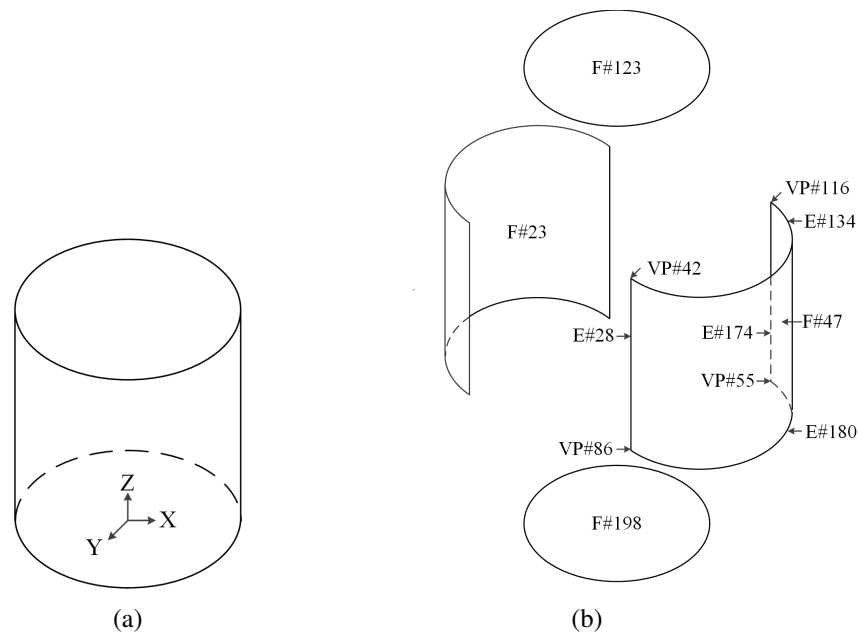


Figure 3.2 A cylindrical part (a) how a cylinder is represented during the design phase, and (b) four faces of the cylinder with some details of the first face

```

DATA;
#1 = SHAPE_DEFINITION_REPRESENTATION ( #82, #221 );
#2 = PLANE ( 'NONE', #102 );
#3 = PLANE ( 'NONE', #247 );
#4 = CARTESIAN_POINT ( 'NONE', ( 30.000000000000000, 3.673940397442058400E-015, 45.000000000000000 ) );
#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', ( 0.00000000000000000, 0.00000000000000000, 1.00000000000000000 ) );
#13 = CLOSED_SHELL ( 'NONE', ( #47, #23, #123, #198 ) );
#14 = LOCAL_TIME ( 22, 52, 35.000000000000000, #72 );
#15 = MANIFOLD_SOLID_BREP ( 'Boss-Extrude1', #13 );
#16 = DIRECTION ( 'NONE', ( 1.00000000000000000, 0.00000000000000000, 0.00000000000000000 ) );
#17 = APPROVAL_STATUS ( 'not_yet_approved' );
#18 = PRODUCT ( 'Fig2', 'Fig2', '', ( #54 ) );
#19 = COORDINATED_UNIVERSAL_TIME_OFFSET ( 0, 0, .AHEAD. );
#20 = APPROVAL ( #228, 'UNSPECIFIED' );
#21 = CYLINDRICAL_SURFACE ( 'NONE', #207, 30.000000000000000 );
#22 = CC_DESIGN_DATE_AND_TIME_ASSIGNMENT ( #131, #150, ( #165 ) );
#23 = ADVANCED_FACE ( 'NONE', ( #6 ), #39, .T. );
#24 = EDGE_CURVE ( 'NONE', #55, #86, #121, .T. );
#25 = APPROVAL_PERSON_ORGANIZATION ( #80, #52, #173 );
#26 = CALENDAR_DATE ( 2017, 9, 11 );
#27 = DATE_AND_TIME ( #245, #49 );
#28 = EDGE_CURVE ( 'NONE', #42, #96, #8, .T. );
#29 = CARTESIAN_POINT ( 'NONE', ( 0.00000000000000000, 0.00000000000000000, 45.000000000000000 ) );
#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', #98, 30.000000000000000 );
#40 = PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE ( 'ANY', '', #18, .NOT_KNOWN. );
#41 = DIRECTION ( 'NONE', ( 0.00000000000000000, 0.00000000000000000, 1.00000000000000000 ) );
.
.
#253 = CIRCLE ( 'NONE', #83, 30.000000000000000 );
ENDSEC;
END-ISO-10303-21;

```

Figure 3.3 Part of the STEP AP 203 file of the cylinder in fig.3.2

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 3.2b 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. This can be noticed by taking the first face as an example, which is represented by line #47 as follows:

```
#47 = ADVANCED_FACE ( 'NONE', ( #94 ), #21, .T. );
```

The first pointer #94 refers to the bound, whereas the second pointer #21 denotes the surface of the face #47. The ADVANCED\_FACE entity ends with a Boolean flag, which indicates whether the loop direction is oriented in accordance or opposed to the surface normal [159].

### 3.3.1 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. The bound line #94 provides the following records:

```
#94 = FACE_OUTER_BOUND ( 'NONE', #251, .T. );
```

Since the bound #94 includes “FACE\_OUTER\_BOUND” string, this means the face #47 has an outer bound. Regardless the type of the bound, it always contains one “EDGE\_LOOP” indicator. The “EDGE\_LOOP” is given as:

```
#251 = EDGE_LOOP ( 'NONE', ( #190, #66, #71, #84 ) );
```

where #190, #66, #71, and #84 represent the oriented-edges. 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. The lines of these oriented-edges are given as following:

```
#190 = ORIENTED_EDGE ( 'NONE', *, *, #134, .F. );
```

```
#66 = ORIENTED_EDGE ( 'NONE', *, *, #28, .T. );
```

```
#71 = ORIENTED_EDGE ( 'NONE', *, *, #180, .T. );
```

```
#84 = ORIENTED_EDGE ( 'NONE', *, *, #174, .F. );
```

where #134, #28, #180, and #174 are pointers to the edge-curves. An edge-curve is a topological entity bounded by a start and an end point. The edge-curves lines are given as:

```
#134 = EDGE_CURVE ( 'NONE', #42, #116, #70, .T. );
#28 = EDGE_CURVE ( 'NONE', #42, #86, #8, .T. );
#180 = EDGE_CURVE ( 'NONE', #86, #55, #124, .T. );
#174 = EDGE_CURVE ( 'NONE', #116, #55, #176, .T. );
```

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. This can be noticed through the EDGE\_CURVE #28, since #42 and #86 are indicators of the start point vertex and end point vertex respectively, whereas #8 refers to the edge-curve type.

```
#42 = VERTEX_POINT ( 'NONE', #110 );
#86 = VERTEX_POINT ( 'NONE', #51 );
#8 = LINE ( 'NONE', #4, #189 );
```

where #110, #51, and #4 are pointers of Cartesian points, and #189 is an indicator of a vector. 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. For example, the Cartesian point #110 is given as:

```
#110 = CARTESIAN_POINT ( 'NONE', ( 30, 0, 45 ) );
```

where 30, 0, and 45 is the values of x, y, and z axes, respectively. The vector is a geometric entity that describes the direction of a line, and it is denoted as:

```
#189 = VECTOR ( 'NONE', #218, 1000 );
```

where #218 is a pointer of a direction, which is given as:

```
#218 = DIRECTION ( 'NONE', ( -0, -0, -1 ) );
```

Meanwhile, a circle is defined by a start point, end point, and centre point vertices as well as a radius and two directions. For example, the edge-curve #134 has three pointers: #42, #116, and #70. Although these three pointers have the same explanation of a line three pointers, the last pointer #70 includes a different data type since it represents a circle.

```
#70 = CIRCLE ( 'NONE', #69, 30 );
```

where #69 refers to “AXIS2\_PLACEMENT\_3D”, and 30 represents the radius of the circle. The AXIS2\_PLACEMENT\_3D provides the necessary information to describe a circle centre and direction.

#69 = AXIS2\_PLACEMENT\_3D ( ‘NONE’, #215, #12, #152 ) ; where #215 refers to a Cartesian point which represents the circle’s centre, and both of #12 and #152 refers to directions.

### 3.3.2 Surface

Back to the surface #21 of the first face #47, a surface is an indicator of the face type, which can take only one prospect such as a plane, cylindrical, conical, toroidal, etc. The surface #21 is given as:

#21 = CYLINDRICAL\_SURFACE ( ‘NONE’, #207, 30 ) ;

where #207 is a pointer of AXIS2\_PLACEMENT\_3D, which has the same explanation as well as of the circle. The last digit 30 represents the radius of the “CYLINDRICAL\_SURFACE”. So far, the first face #47 has been covered in terms of bound and surface. The same procedure can be followed with the other three faces #23, #123, and #198. For more details about the STEP AP203 see [157].

## 3.4 Discussion and summary

STEP is one of the international PDE standards, and the most utilised in developing AFR and CAPP systems because it has the ability in transferring rich information of a product through its lifecycle. Although the STEP standard has been categorised and published as a series of nearly 2,000 parts, many researchers have wrong notions about a few of its symbols and characteristics. This can lead, for example, to a fault in specifying the convexity or concavity of a toroidal face, and difficulty in distinguishing internal and external features of a part. Cases of such mistakes have been mentioned with details of corrections and solutions in the next chapter. Different STEP APs have been developed for transferring product data between industrial applications, with these AP203, AP214, and AP242 being widely used in developing AFR systems. Despite the fact that each AP is used for a specific industry area, they have the same hierarchical structure. It depends on the AFR developer to consider only one of the APs as an input to the system and if necessary makes a small adjustment to the syntax so that it is accepted. This is to rectify any errors/issues that can occur during the recognition process.

Whilst the use of STEP files in AFR systems has numerous benefits, it 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. These issues have been considered in the developed smart interactive AFR system. This has been explained with examples in chapter four.

## **Chapter 4**

# **A smart interactive AFR system for rotational parts**

### **4.1 Introduction**

Sharing product design information with other downstream applications, such as process planning, is a major barrier to developing 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 AFR techniques is considered an indispensable concept for transferring product data between CAD and ACAPP. This is accomplished using one of the international 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 recognise 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 chapter presents a novel and smart interactive AFR (SI-AFR) system that has been developed for rotational parts features 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. 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 rules based technique. Figure 4.1 shows examples of some external and internal predefined features.

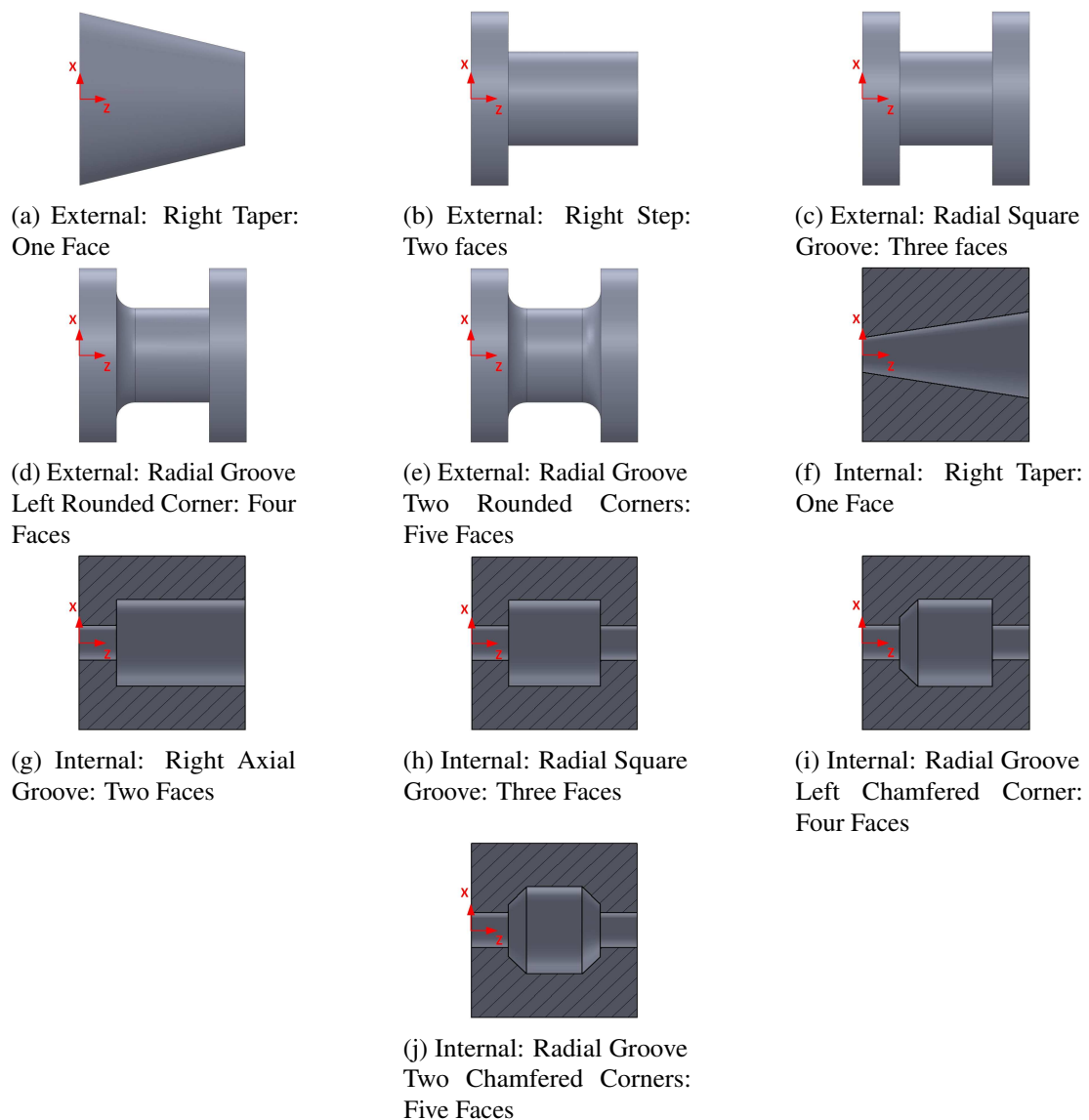


Figure 4.1 Examples of external and internal predefined features



This predefined set of features is saved in the system's database, and represents the third module. 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 features 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 features 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. Figure ?? shows the structure of the proposed SI-AFR system.

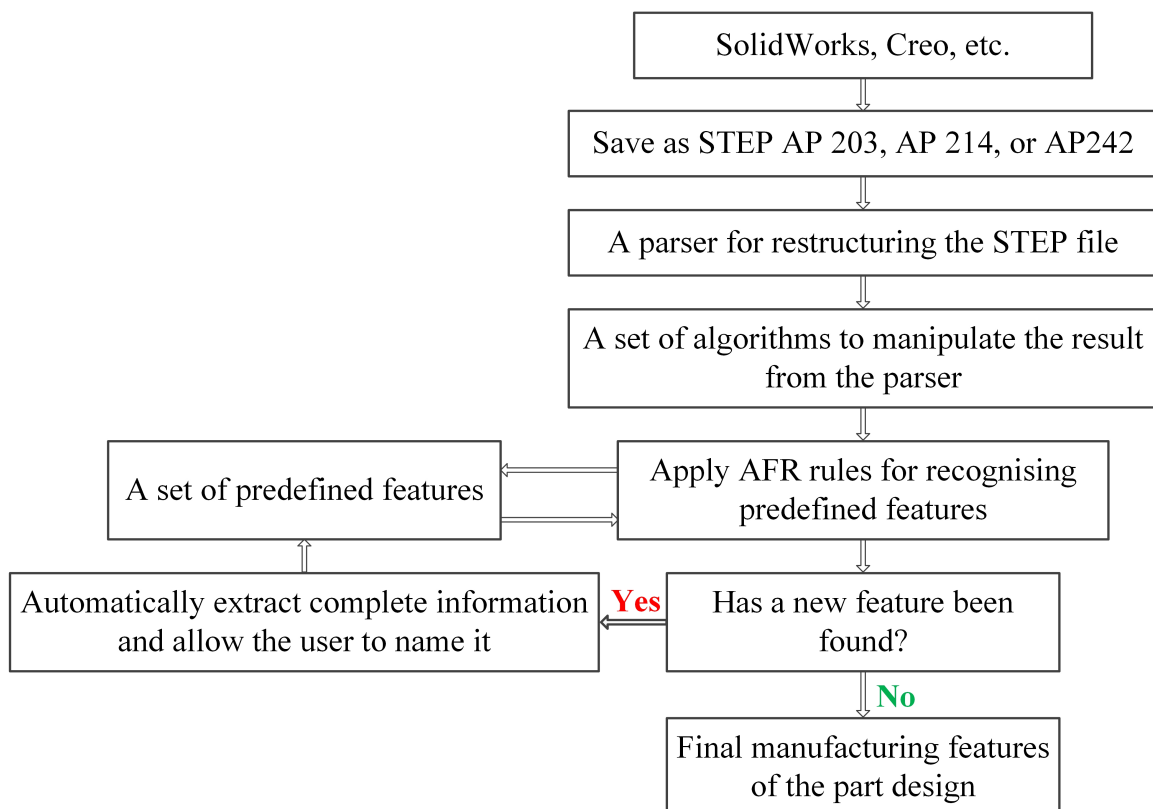


Figure 4.2 The structure of the proposed SI-AFR system

## 4.2 Developing a parser for a STEP file

As mentioned in chapter 3, 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 necessary to develop a parser so as to re-arrange the data in a comprehensible way. Although many commercial parsers have been developed to extract information from the STEP files, each of these parsers serves a special need. For example, some of them extract one level of information and that could be the part's surfaces or the end points. Therefore, developing a parser that provides the proposed AFR system with all the required information was an indispensable task.

The parser that was developed in this work imports and reads any of the STEP files AP 203, AP 214, or AP242, 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 4.3 and appendix B shows part and full details of the resultant STEP file AP 203 after applying the parser, respectively. The parser is developed 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.

```

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_LOOP #136 #197 #161 #236

```

Figure 4.3 Part of the resultant STEP file after applying the parser

### 4.3 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.

#### 4.3.1 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 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 convex. 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 figures 4.4a 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 to the fact 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 rotation from the start to the end point is always *counter clockwise*.

Figures 4.5 a and b show the effect of swapping positions between the start and 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 or 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.

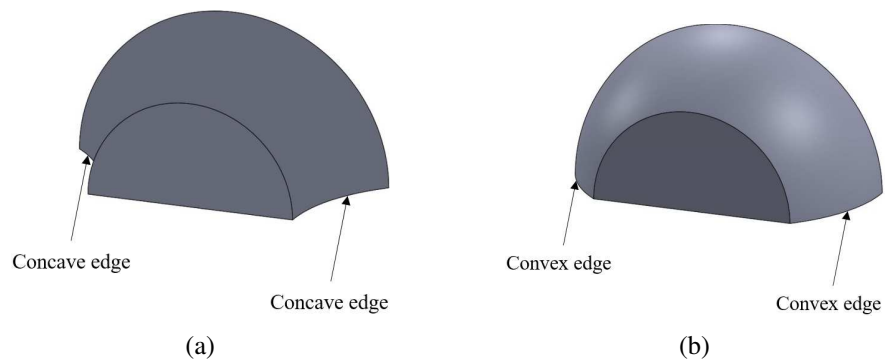


Figure 4.4 Toroidal surfaces (a) concave, and (b) convex

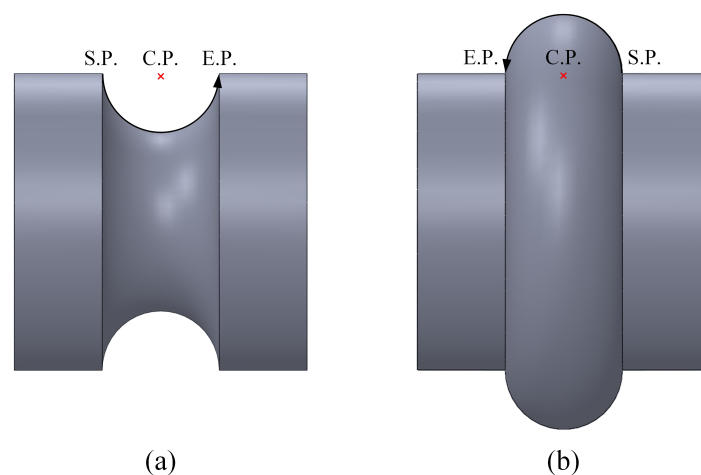


Figure 4.5 The effect of the start and end point on forming toroidal shapes (a) concave shape, and (b) convex shape

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 [160] the authors assumed that “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 [161], 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.

#### 4.3.2 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 figure 3.2a 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 4.6 shows one that merges the information of two symmetrical cylindrical faces, and saves them as single one.

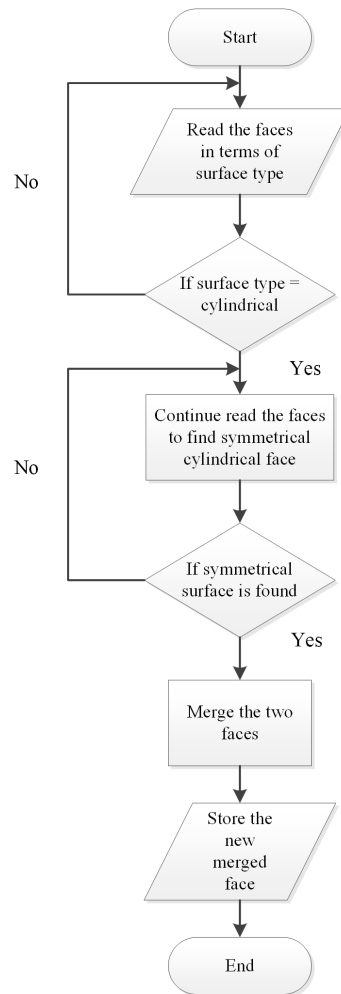


Figure 4.6 An algorithm for merging the information of two symmetrical cylindrical faces

### 4.3.3 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. Figures 4.7a, 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 figure 4.8a is located in two angular partitions and hence, it is saved in the STEP file as four separated faces, as shown in figure 4.8b. The first step includes the use of the MSF algorithm, in order to merge the information of each two symmetrical toroidal faces, as in figure 4.8c. Despite the implementation of the MSF algorithm helping to combine the information of symmetrical faces, the result is still insufficient in this case, because the toroidal face is still saved as two parts. Hence, another algorithm, which is shown in figure 4.9, 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 (figure 4.8d), to be used later in the feature recognition system.

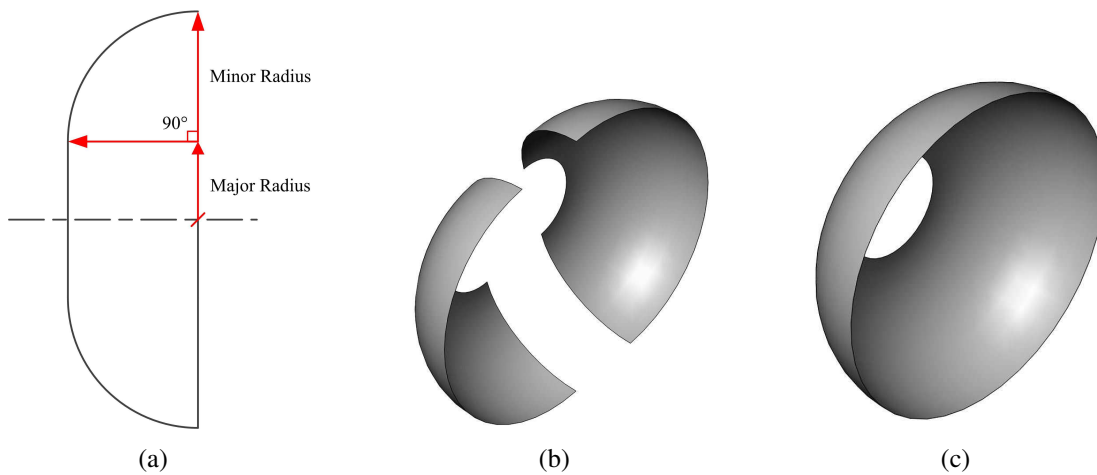


Figure 4.7 Merging adjacent toroidal faces (a) 2D view of a toroidal face, (b) a declaration of the toroidal face in a STEP file, and (c) after merging the two symmetrical toroidal faces



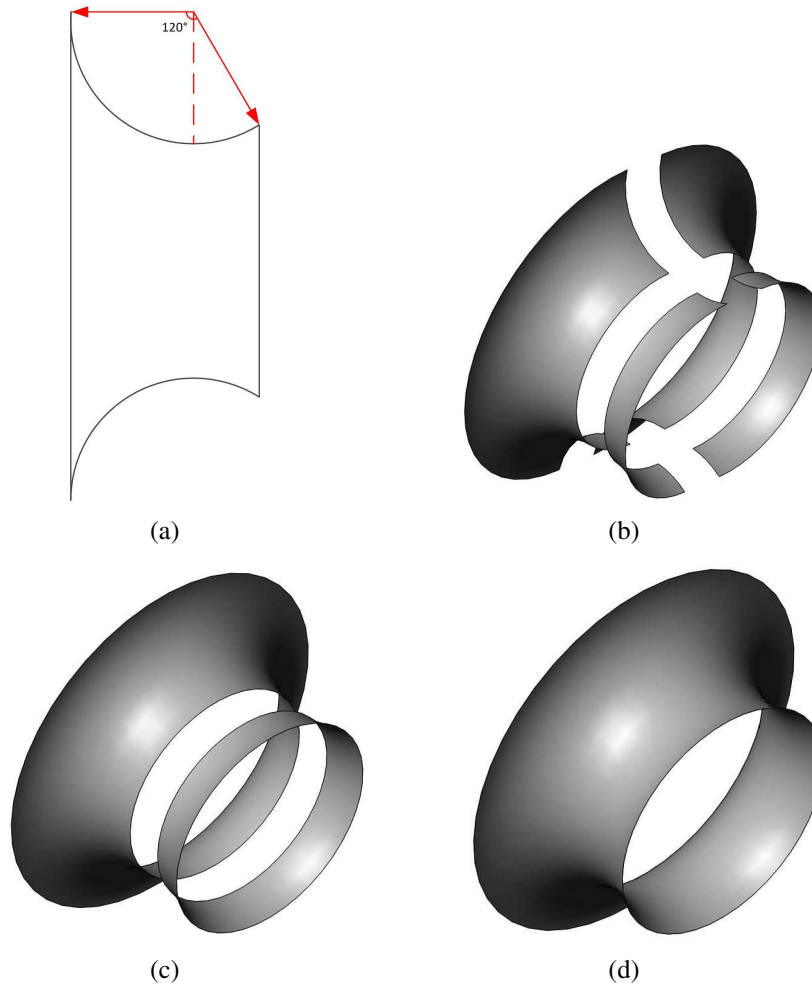


Figure 4.8 A special case for merging adjacent toroidal faces (a) 2D view of a toroidal face, (b) declaration of the toroidal face in a STEP file, (c) face description after merging each two symmetrical toroidal faces, and (d) face description after merging two adjacent ones

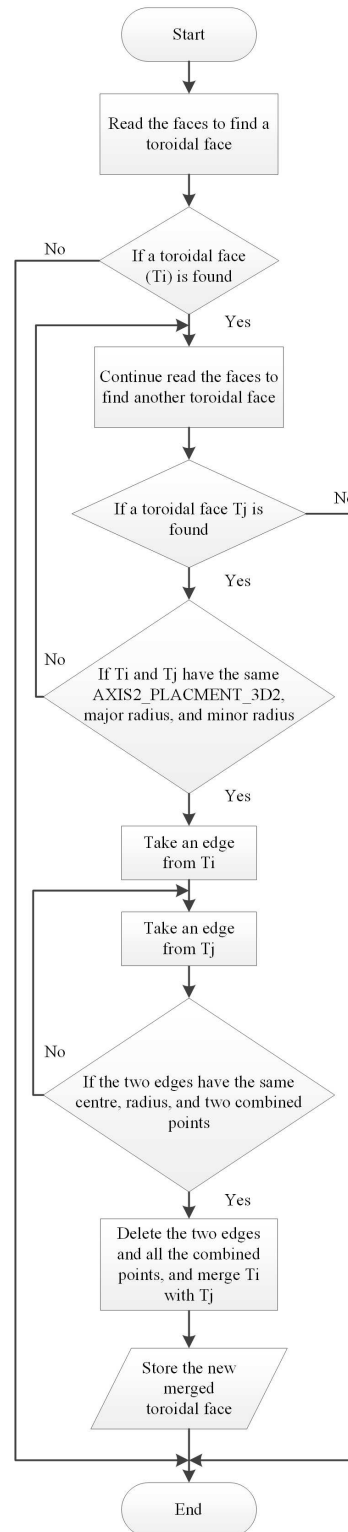


Figure 4.9 An algorithm for merging the information of two adjacent toroidal faces

#### 4.3.4 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 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 features analyser. Thus, a faces sorting algorithm has been developed in order to not only facilitate the task of the feature 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 (radial thrust force) and Z represents the axis of rotation or length (axial feed force). 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 4.10 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) circular plane at the bottom of the hole, and (e) circular 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 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 this example should be as follows:

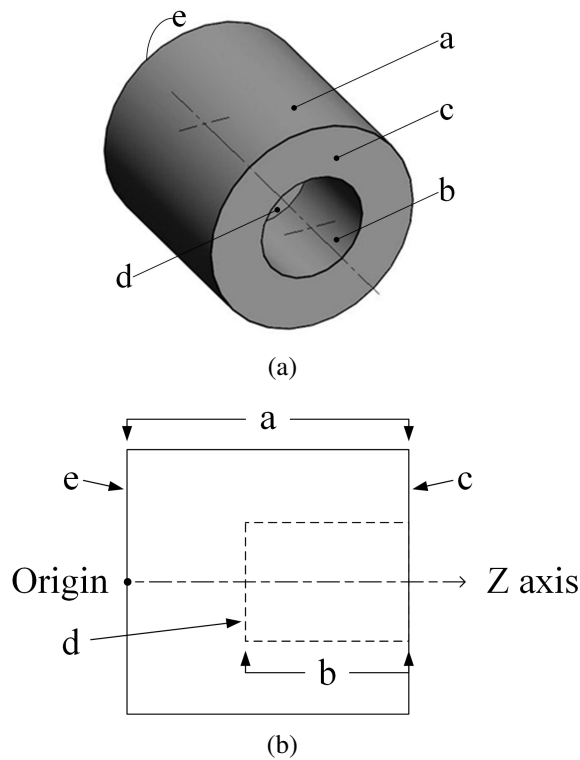


Figure 4.10 A cylinder with a blind hole (a) schematic representation and (b) a side view of the cylinder

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).

#### 4.3.5 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, 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 figure 4.11, its original STEP file information does not clearly indicate 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 other internal features. This is achieved in three steps as follows:

***Step one: 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.

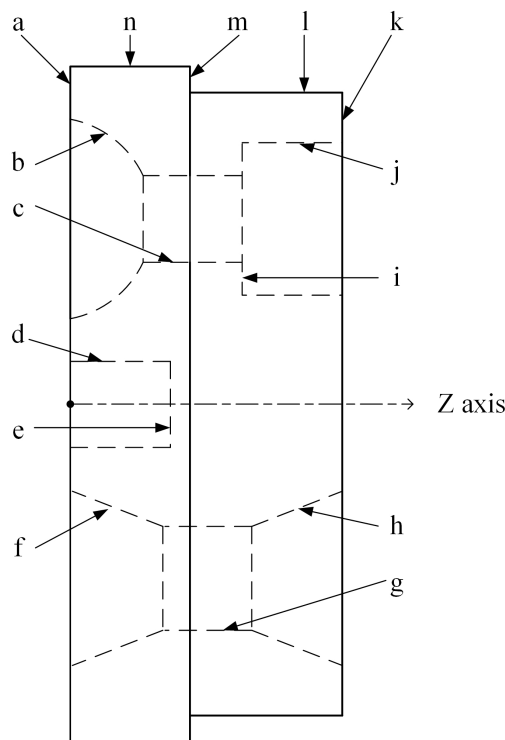


Figure 4.11 A part design includes 14 external and internal faces

**Step two: 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 three: checking for other possibilities**

After the virtual subtracting of the previous internal nodes has been completed, the remaining shape appears as in figure 4.12. It takes a mere glance to distinguish that there is another hole in the shape, which requires recognition and then 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 two, 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  $d$ ).

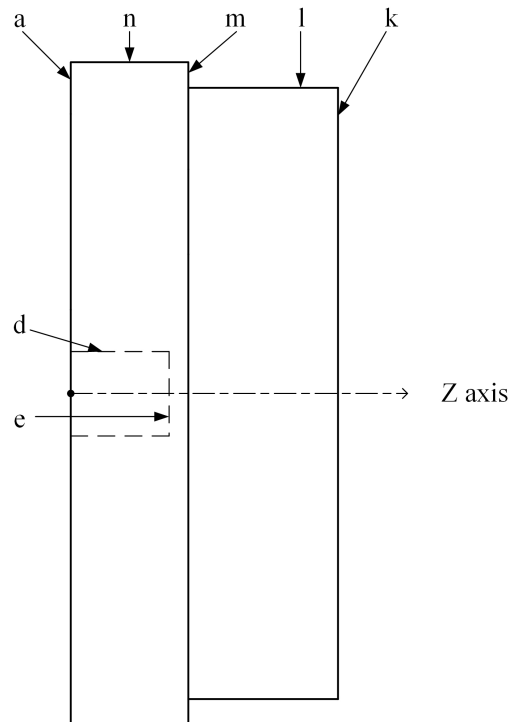


Figure 4.12 The remainder of the part after the virtually subtraction of two nodes of internal faces

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 [162] 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 [160], 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.

#### 4.4 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 features 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.

#### 4.4.1 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 nodes of faces. 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 one to five faces. Thus, the predefined features set is categorised into five groups, with those in each group having the same number of faces. Regardless of the number of faces that 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 figure 4.13 and saved in the database, as in table 4.1.



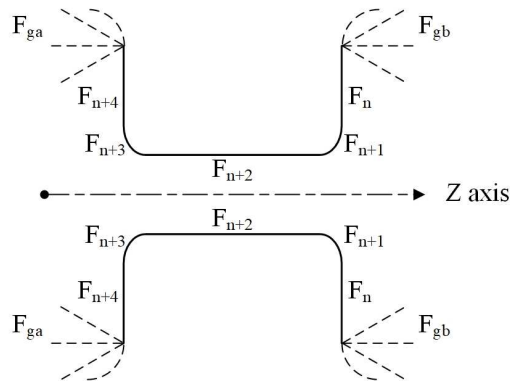


Figure 4.13 A square groove with two round corners in the base

The SI-AFR system works as loops. When going back 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 match is found, the first loop is continued, and the bundle is updated by retaining 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 match 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 4.14 shows a map of the predefined features recognition loops.

Table 4.1 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	$(F_n (\text{MaxX}) \text{ or } F_{n+4} (\text{MaxX})) - F_{n+2} (\text{MaxX})$

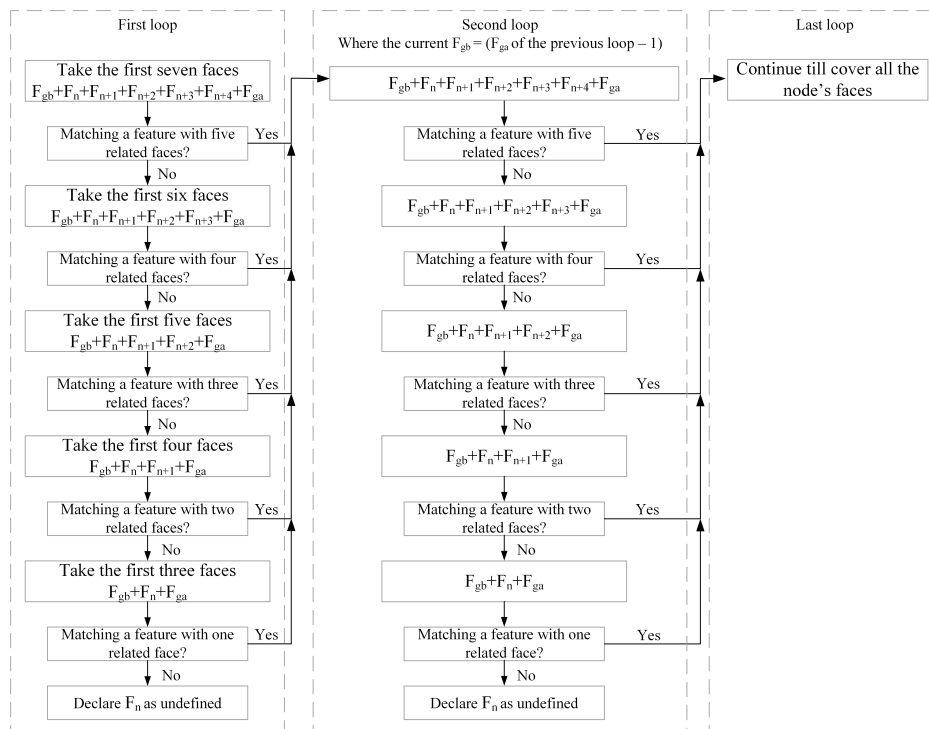


Figure 4.14 A map of the predefined features recognition loops

One of the contributions with this methodology is the reduction of the stages required to recognise certain 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 figure 4.15. This blind square groove consists of three faces, these being: a cylindrical 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 figure 4.13 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.

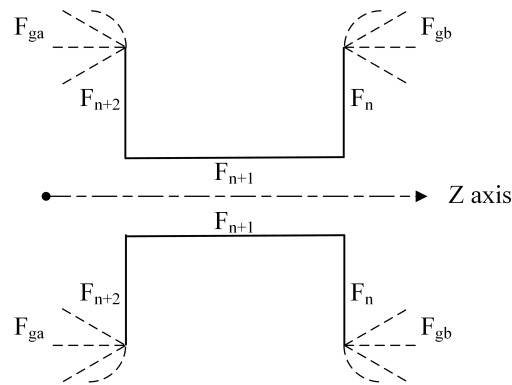


Figure 4.15 A blind square groove

#### 4.4.2 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 features 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.

##### 4.4.2.1 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 4.16 and table 4.2 shows an example of the upper profile of a new feature and how its data are extracted, respectively.

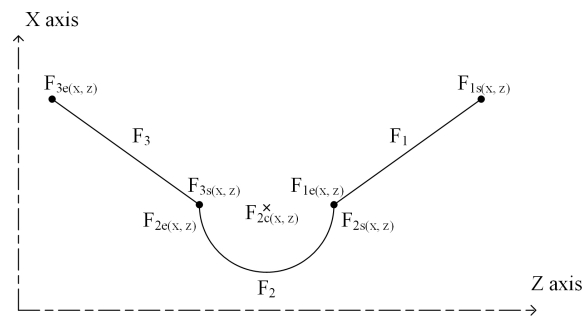


Figure 4.16 An example of a new feature

#### 4.4.2.2 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 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, figure 4.17a 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 figure 4.17b because it has different values. Therefore, the feature's information in this example (figure 4.17a) should be saved as follows:

Table 4.2 Extracting a new feature's data

Feature name:	"As the user suggested"
Number of related faces	Three
F <sub>1</sub>	Type of surface: conical (left).
	F <sub>1sz</sub> = value.
	F <sub>1sx</sub> = value.
	F <sub>1ez</sub> = value.
	F <sub>1ex</sub> = value.
	F <sub>1s-to- F<sub>1e</sub></sub> : linear.
F <sub>2</sub>	Type of surface: toroidal (concave).
	F <sub>2sz</sub> = value.
	F <sub>2sx</sub> = value.
	F <sub>2ez</sub> = value.
	F <sub>2ex</sub> = value.
	F <sub>2cz</sub> = value.
	F <sub>2cx</sub> = value.
	Major radius.
	Manor radius.
F <sub>2s-to- F<sub>2e</sub></sub> : circler.	
	Direction: CW.
F <sub>3</sub>	Type of surface: conical (right).
	F <sub>3sz</sub> = value.
	F <sub>3sx</sub> = value.
	F <sub>3ez</sub> = value.
	F <sub>3ex</sub> = value.
	F <sub>3s-to- F<sub>3e</sub></sub> : linear.
F <sub>gb</sub> and F <sub>ga</sub>	Type of surface: "As they appear in the design".

1. Face1: left taper (linear);
2. Face2: 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:
  - $F_{2sz} > F_{3sz}$ ;

- $F_{2sx} = F_{3sx}$ ;
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).

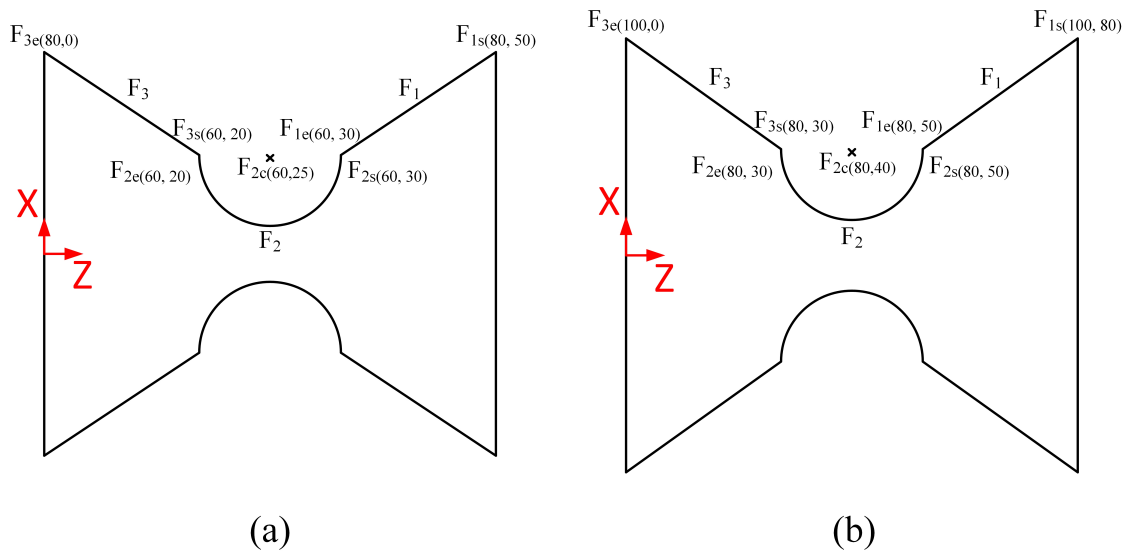


Figure 4.17 Saving a new feature based on relationships between its geometrical values (a) a new feature with specific geometrical values and (b) a feature with same characteristics and different values

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.

#### 4.4.2.3 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 fifty-four. As mentioned, this number can be 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.

## 4.5 Case study

In order to evaluate the proposed system's 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.

### 4.5.1 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 and three internal ones. Figure 4.18 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. Certain processes are manipulated: checking 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 4.4.1. 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: 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 4.19, table 4.3, and table 4.4 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.

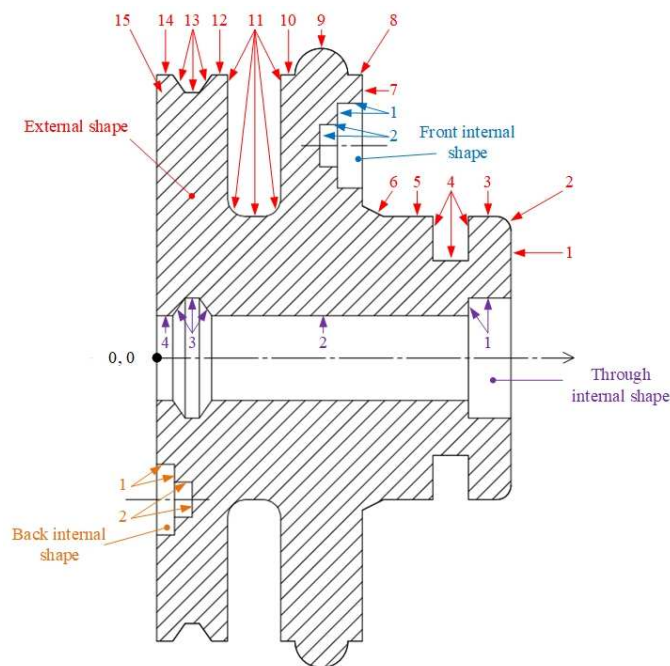


Figure 4.18 Top section of case study 1

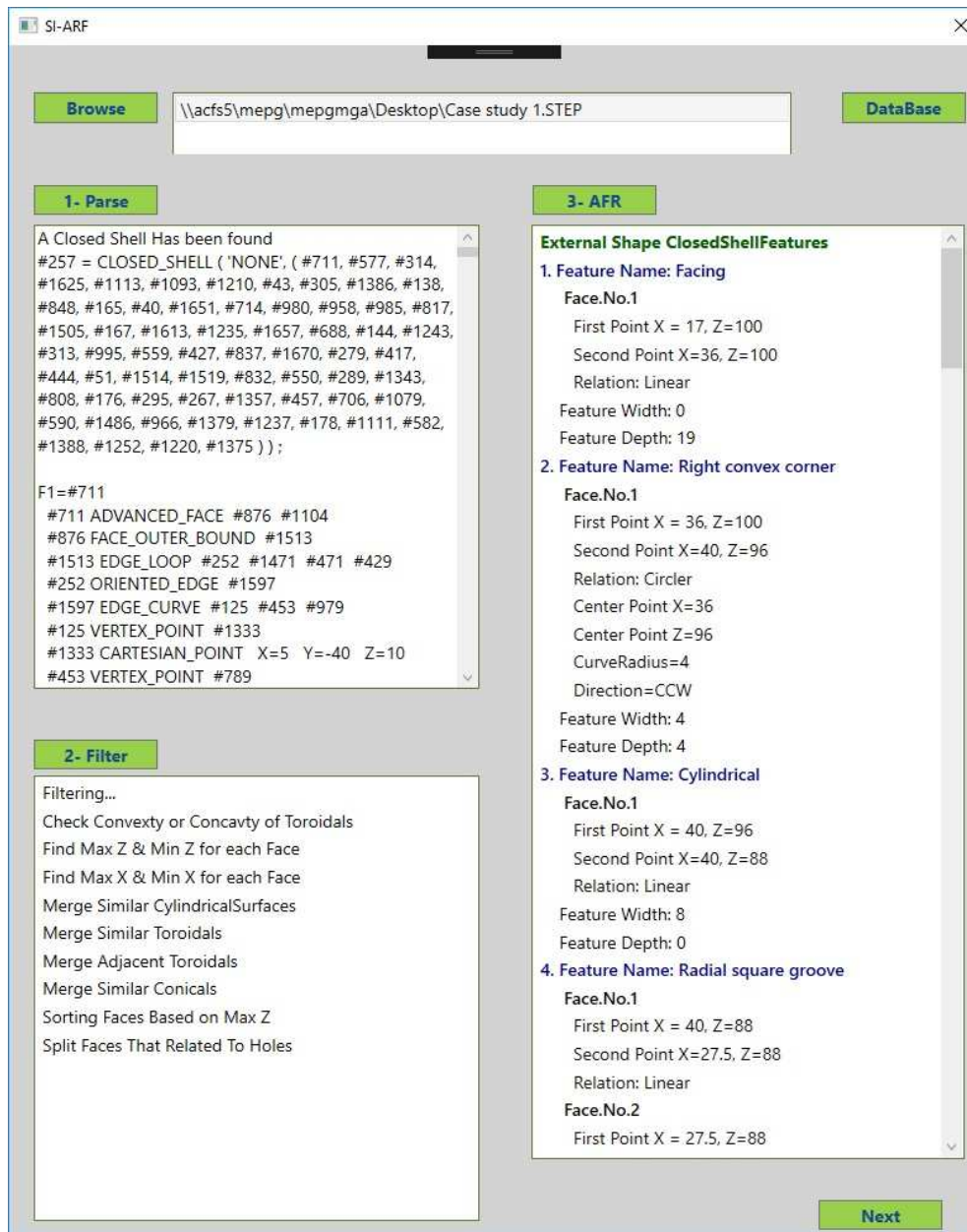


Figure 4.19 The system's main window after applying the three steps for case study 1

Table 4.3 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	Circular	36	4	CCW	4	4		
3	Cylindrical	1	40	96	40	88	Linear				8	0		
		1	40	88	27.5	88	Linear							
4	Radial square groove	2	27.5	88	27.5	78	Linear				10	12.5		
		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	Circular	80	7.5	CCW	15	7.5		
10	Cylindrical	1	80	39	80	35	Linear				4	0		
		1	80	35	45	35	Linear							
		2	45	35	40	30	Circular	45	5	CW				
11	Radial groove two rounded corners	3	40	30	40	25	Linear				15	40		
		4	40	25	45	20	Circular	45	5	CW				
		5	45	20	80	20	Linear							
12	Cylindrical	1	80	20	80	15.5	Linear				4.5	0		
		1	80	15.5	75	12	Linear							
13	Two sides tapered groove	2	75	12	75	8	Linear				11	5		
		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 4.4 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			
		1	0	0	15.5	12	12	17	Linear			
3	Two sides tapered groove	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			

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 figure 4.20. Such information can be used while 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.

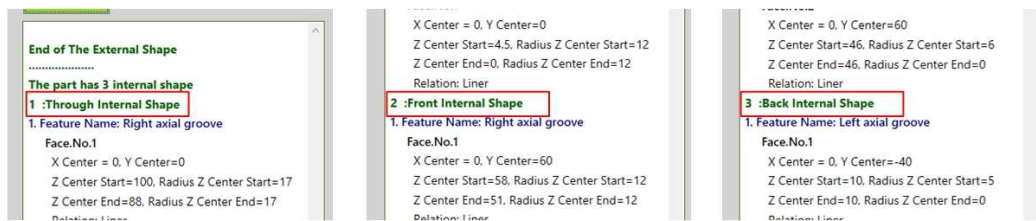


Figure 4.20 Three internal shapes with different locations and specifications

#### 4.5.2 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 figure 4.21. 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 figure 4.22, 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 figure 4.23. 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 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.

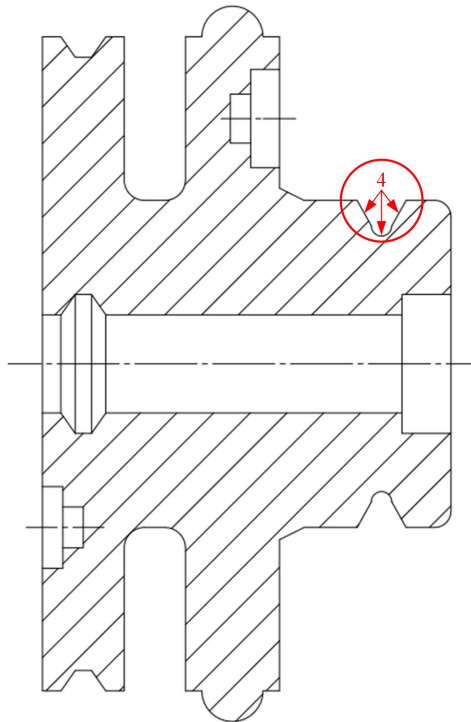


Figure 4.21 A part design with a new feature

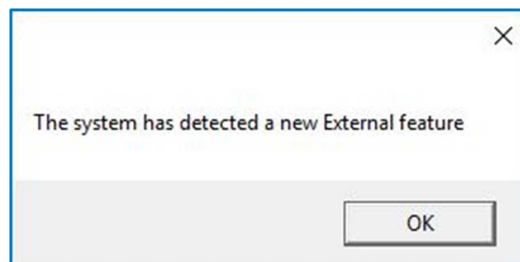
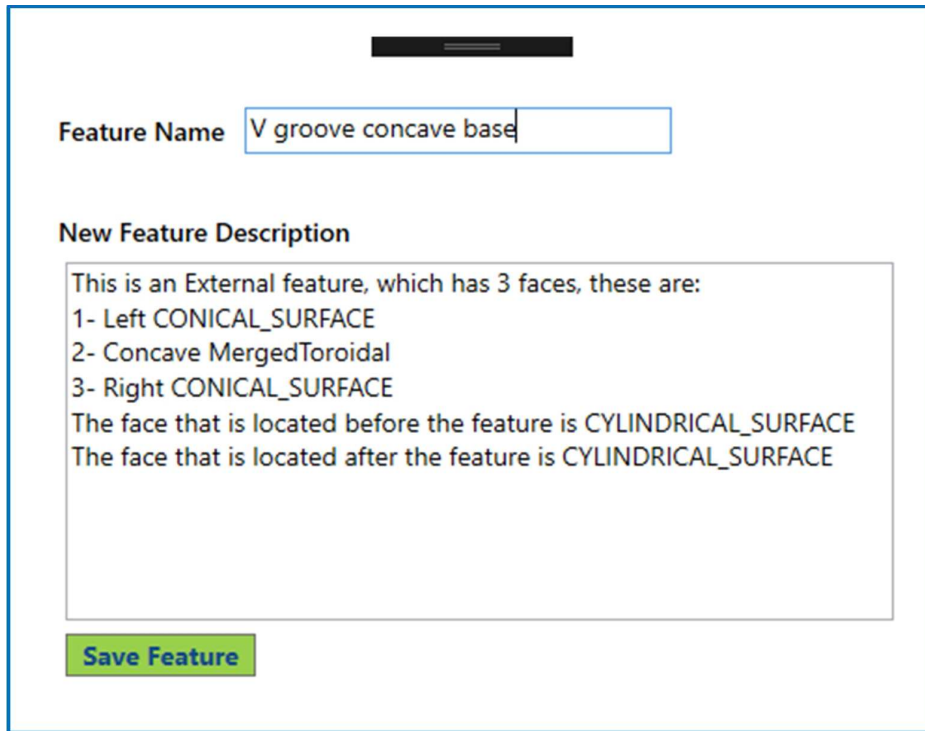


Figure 4.22 A declaration window about a new feature



Feature Name

**New Feature Description**

This is an External feature, which has 3 faces, these are:  
1- Left CONICAL\_SURFACE  
2- Concave MergedToroidal  
3- Right CONICAL\_SURFACE  
The face that is located before the feature is CYLINDRICAL\_SURFACE  
The face that is located after the feature is CYLINDRICAL\_SURFACE

Figure 4.23 A description and naming window

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 fifty-five, instead of the initial number of fifty-four.

## 4.6 Discussion and summary

Computer-aided design (CAD) and computer-aided process planning (CAPP) are two essential concepts in the manufacturing industry. However, they are isolated from each other due to their different nature 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 chapter, 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 rounded corners in figure 4.13, 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 features intersecting issues. However, the main contribution of this work is concentrated in the second part of the system, since new features can be detected, identified, and added to the predefined features 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.

Despite the SI-AFR system being able to minimise features intersecting issues, extra work and programming are still required for this matter to be completely resolved in order to optimise the manufacturing process by achieving high production quality with minimum cost and time. This will be a part of the next chapter along with extracting of geometric dimensioning and tolerancing (GD&T) from 3D models and attaching them with the recognised manufacturing features from the SI-AFR system.



## **Chapter 5**

# **Intersection of manufacturing features**

### **5.1 Introduction**

The availability of rich geometrical and topological data that PDE standards can provide, has influenced researchers to integrate design and manufacturing systems using feature technologies. The use of features gives meaning to a part's attributes, helps in analysing its geometry into recognisable and meaningful regions, and consequently improves the communication between design and manufacture [163]. Automated feature recognition (AFR) is one of the main approaches of feature technology; hence, different techniques have been used to develop a wide range of AFR systems. Whilst different AFR systems do not have the same abilities regarding the number and types of features they can recognise, most of them have common limitations: (i) the recognition of a specific set of predefined features, (ii) the limited ability to learn how to recognise a new feature, and (iii) unable to solve features intersecting issues. The previous chapter has proposed a smart and interactive AFR (SI-AFR) system that is able to declare, analyse, extract information, give names, and add new rotational features to the system database, as well as solving simple features intersecting issues. However, more investigation is required for solving complex cases of manufacturing features intersections. This chapter explains the issue in detail and presents a solution for the intersections of recognised manufacturing features from the SI-AFR system.

## 5.2 Defining the problem

As mentioned, the SI-AFR system is able to recognise new features in addition to the predefined ones; meanwhile, it solves simple features intersecting issues. The square groove with two rounded corners at the base in figure 4.13 demonstrates the case of solving a simple features intersecting issue. This is by defining the entities that form the square groove as one feature, instead of five individual ones, in one step during the feature recognition procedure. However, extra analysis and processing are required with complex cases of features intersecting. For example, the SI-AFR system analyses the part in figure 5.1a as eight rotational features: F1 facing, F2 cylindrical, F3 left step (shoulder), F4 square groove with two rounded corners, F5 right step (shoulder), F6 right taper, F7 cylindrical, and F8 facing. Although it appears as an ideal analysis of the part's features and reflects its functionality, this might not be practical when it comes to machining. The reason behind this is related to the fact that this shape is machined from a blind cylindrical workpiece figure 5.1b. In order to achieve the rough cutting cycle, which includes removing most of the material from the original stock, this workpiece should not be split as five individual volumes (A, B, C, D, and E) based on the part's features figure 5.1c. This is because machining each feature before moving to another one consumes time and cost, and might cause technical issues, such as damaging a surface of a feature and interrupting the cutting tool movement. Thus, a deep vision and specific processing are required to consider all features' intersections of a part, in a way that considers all the possibilities of regenerating intermediate features, which are a result from intersections of original features, and chooses the ideal possibility. Figure 5.1d gives an example of an intermediate feature (zone B), which is a square groove that is formed from the intersection of original features F3, F4, and F5. However, it takes merely a glance at this suggestion to evaluate it as a non-ideal solution. This is because the intermediate square groove feature at zone B intersects with two other volumes A and C, hence this solution reduces the issue but is not an optimal solution. Another material removing strategy is shown in figure 5.1e where the removed volume is divided into three parts (A, B, and C). Zone A directly effects in forming F2 and F6, as well as in F3, F4, and F5 in an indirect way; whereas the intermediate feature B is composed of F3 and F5, and reduces the extra material above F4. Finally, zone C is now ready to be machined, and F4 will be produced with all its five entities.

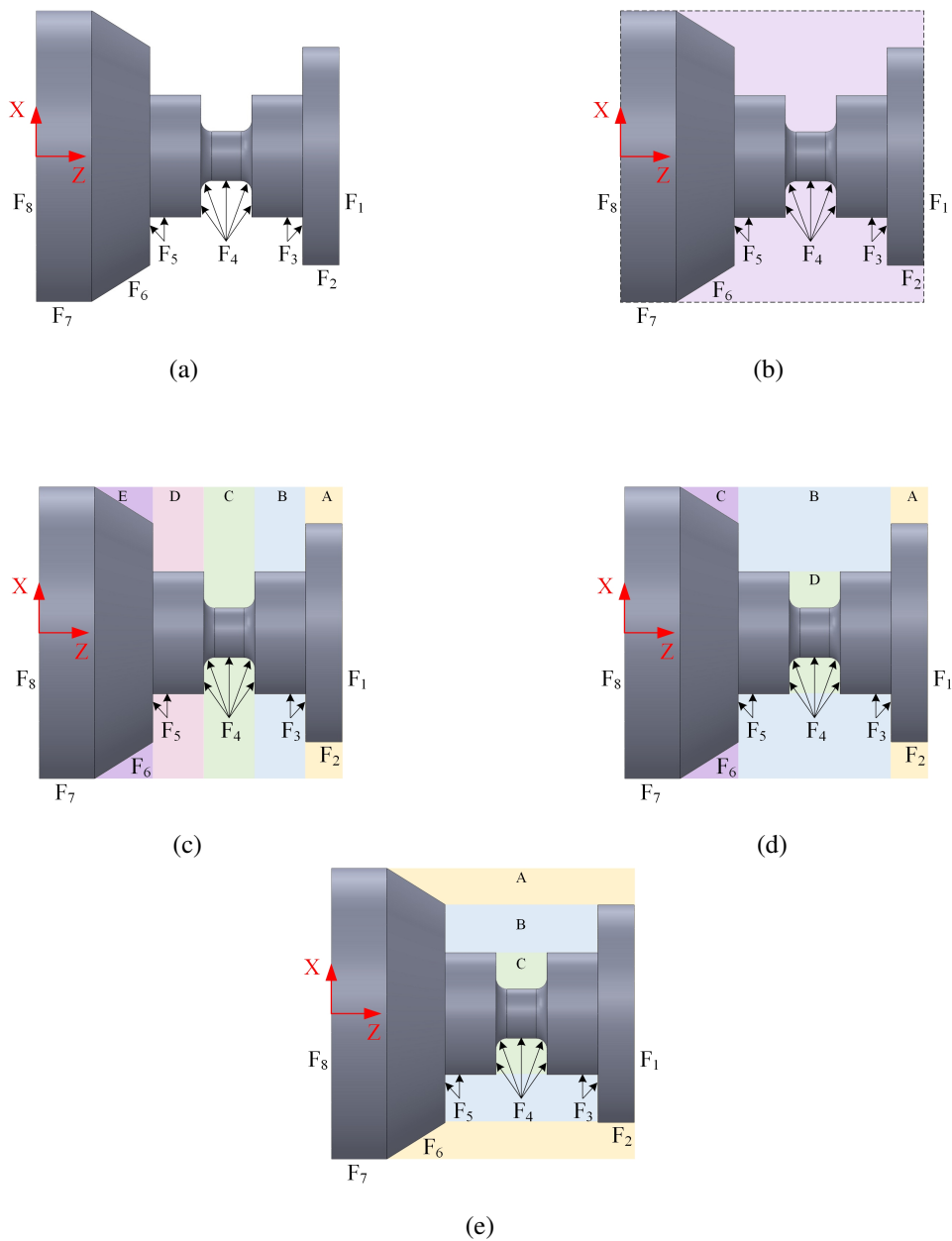


Figure 5.1 Different strategies for considering features intersecting (a) a design model, (b) original blank, (c), (d), and (e) three different strategies of dividing the machining area

From a manufacturer point of view, the machining of the part based on the last dividing strategy of volumes is considered reasonable. This is because the cutting tool path is decreased in an efficient manner that assists forming one or more feature in a direct and indirect way. However, this is a simple

case of features interacting with one another, and has few possibilities of volume divisions which produce intermediate features. More prospects are generated from complex parts with many features, hence an efficient methodology should be developed in such cases. The next section proposes an approach for solving features intersections issues based on the concept of volume subtraction.

### 5.3 A methodology of subtracting

The recognised manufacturing features from the SI-AFR system are considered “*final features*”. These describe the final shape of the product and should be produced in an accurate way. However, during the manufacturing process, one or more interactions occur between the final features and produce intermediate or “*intersecting features*”. This is an inevitable matter in most, if not all, the cases of rotational parts, since the stock is a blank cylinder and demands several cutting paths to reach the final features and consequently the required shape of the product. In contrast with the final features, the intersecting ones can take many shapes and possibilities depending on the vision of the process planner. Hence, the analysis of the final features can take multiple combinations of intersecting features to be removed from the initial workpiece. As mentioned in section 5.2, these combinations vary between unreliable, acceptable, and optimum solutions. For the purpose of getting an optimal option for each features intersecting case, a “features subtraction” system has been developed. The proposed system scans all the final features in both X and Z axes, divides the part into areas based on developed rules, and in some cases updates the geometrical data of the final features. This section explains the methodology of the features subtraction step by step. This is by using different examples in each step to illustrate different cases.

However, before analysing the final features and concluding the intersecting ones, the system starts with calculating the minimum required blank size. This is done by scanning all the start, end, and tangent points of each feature, which the SI-AFR system has extracted in terms of the X and Z co-ordinate points. Since the adopted coordinate system is the same for turning machines, the maximum X value represents the radius of the stock and Z represents its length. Figure 5.2 shows an example of which points are taken as maximum X and Z, and consequently the blank workpiece is the shaded area.

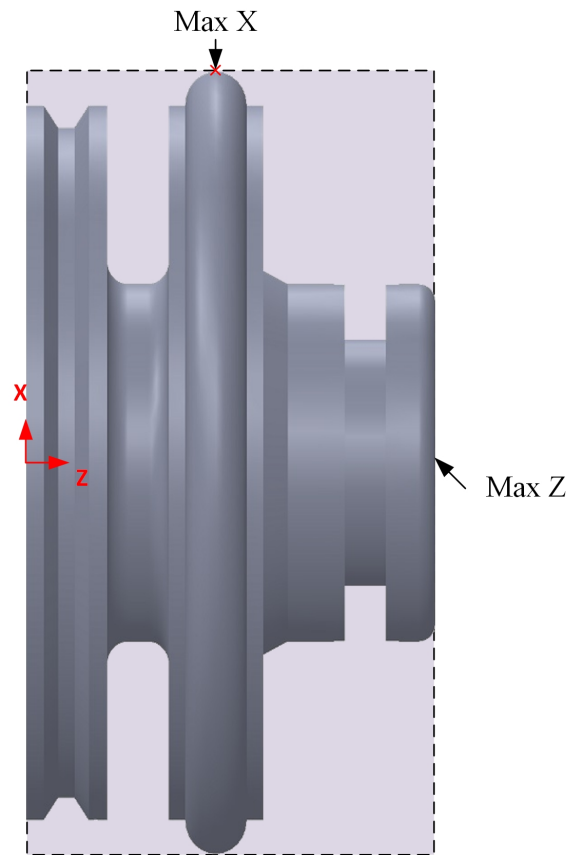


Figure 5.2 An example of which points are taking as maximum X and Z

As mentioned in chapter 4, the SI-AFR system describes any 3D model as one external shape and one or more internal shapes if it has holes or internal features. Since the external and internal shapes have different manufacturing requirements, the features subtraction part deals with each of them individually.

### 5.3.1 Features subtraction of external shapes

#### Step one: finding features with max. X

The first step of the features subtraction system is scanning all the faces of each final feature in the part design. This is to check X values of all the start, end, and tangent points, and find which features have an X value that is equal to the observed X value. The observed X value at this point is the same of the blank radius; however, it can take another value, which will be explained later. Depending on the design, there might be one or more features that have the same observed X value. For example,

the part in figure 5.2 has only one feature shares the same X value with the blank, which is the tangent point of the convex. Whereas, three of the final features of the part in figure 5.3 have the same X value of the blank: (i) the end point of the convex, (ii) the start and end points of the cylindrical surface, and (iii) the start point of the first face in the left step feature.

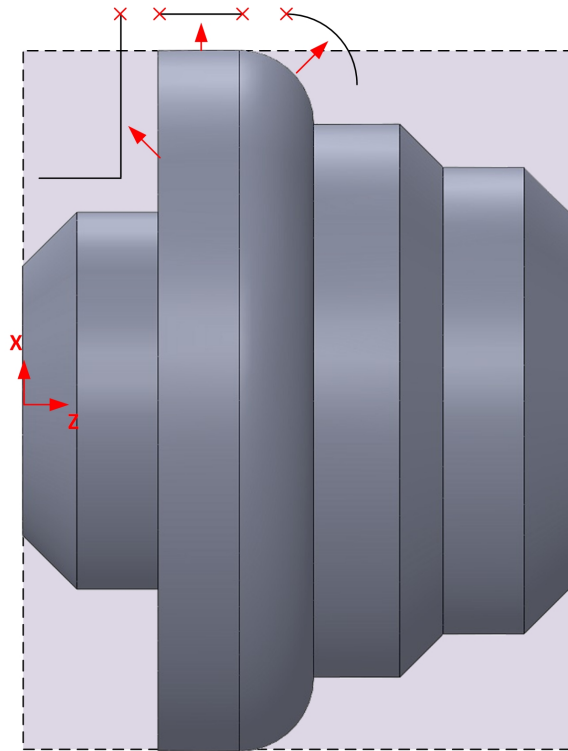


Figure 5.3 Three features have the same X value of the blank

### Step two: grouping the features

As mentioned in step one, there might be more than one feature that have the same X value of the observed one. However, these features must be checked to determine if they are connected to each other. The part design in figure 5.3 shows that all the features, which include the X value of the blank at their end or start points, are connected to each other in one point. A different case is shown in figure 5.4, where the total number of the features with the same X value of the blank is eight. The first three features (convex corner, cylindrical surface, and left taper) are connected to each other as the red points show. The other five are separated from the first group and connected to each other at the green points. Regardless the design of the part, the proposed system gathers the features that

include the same X value of the blank, checks the connection between them, and puts them in groups as a preparation for the next step.

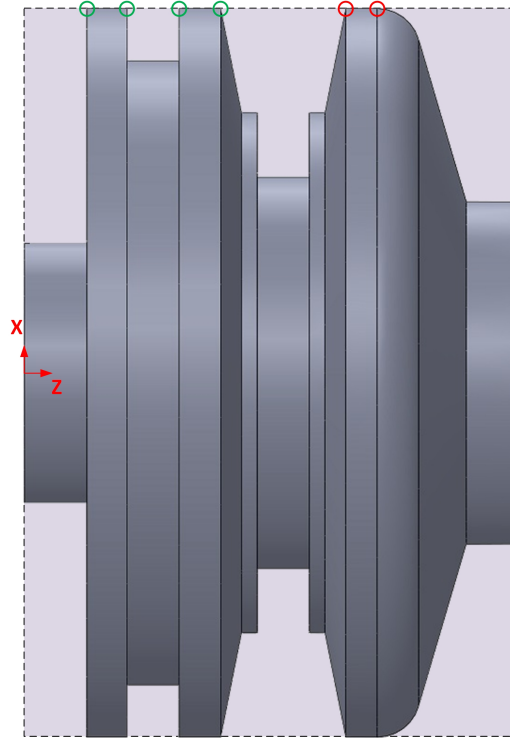


Figure 5.4 Two groups of features have the same X value of the blank

***Step three: scanning features based on a reference one and observed X value***

In this step, the system takes the prepared groups of features one by one. Then, a procedure is applied to each feature in a group. Firstly, the system scans the final features on the right side of the observed one, starting from the first neighbour and so on. The scanning is stopped if one of two conditions is realised. The first condition to be checked is if a feature includes the same or bigger observed X value, and second is if the scanning reaches the far right side of the blank. Figure 5.4 shows an example for each of these possibilities. Firstly, by scanning the final features on the right side of the convex round feature in the first group, the process lasts till the right end of the blank because it does not find a feature that contain the observed X value. Hence, the system considers four features including the reference (convex round, right taper, cylindrical face, and facing). The second example is when the system takes the right taper feature in the second group and scans its right side. After four final features, the system declares the existence of a feature that includes the observed X

value, and stops the scanning process. Consequently, only five features are recorded in this case (right taper, cylindrical face, square groove, cylindrical face, and left taper). Regardless which of these conditions is achieved, the system applies the same procedure for all the reference features. This is by scanning the right side of the reference feature first and records the result, then scans the left side of it and records the result as well before moving to the next feature. All the recorded results, which are bundles of final features, are exported to the next step. Before moving to the next step, it is important to clarify that the process of scanning the left side of a feature has the same logic as the right side.

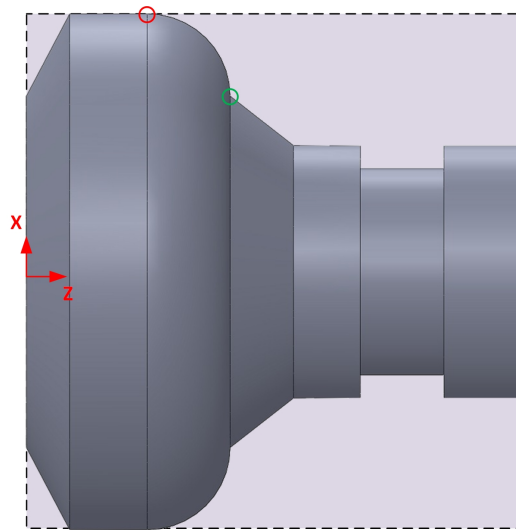
In some cases, the scanning process directly stops when the right and left neighbours of a referenced feature are checked. This is because both of them have the observed X value in at least one of their points. Such a referenced feature is classified as an *independent final feature*. The square groove feature in the second group (figure 5.4) shows an example of the described case. It is surrounded by cylindrical faces, which include the observed X value, on both sides. Hence, the square groove here is an independent final feature, and is not exported to the next step.

#### ***Step four: analysing each bundle of features***

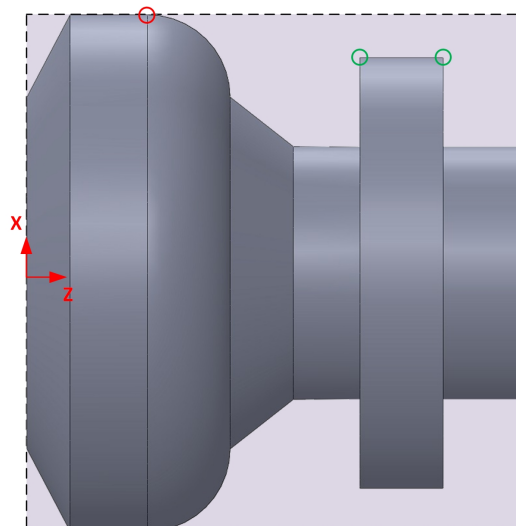
As mentioned in step three, the system records a bundle of features after every scanning process on the right or left side of a referenced feature. Each bundle contains the referenced feature itself in addition to others depending on the case. Obviously, one of the referenced feature's points has the observed X value that the system specified in step one, which should be the highest X value in comparison to other X values of all the features points in this bundle. The system searches all the features in the bundle to find the second highest X value. This can be in any position depending on the case. In order to illustrate possibilities where the second highest X value can be in a bundle of features, figure 5.5 shows two different examples. In these examples, all the observed X points are marked using red circles; whereas, the green circles are used to refer to the second highest X values. Regarding figure 5.5a, the convex rounded corner includes the observed X value at its endpoint, and it is one of three features that share this value (convex corner, cylindrical surface, left taper). By scanning its right side, the system records a bundle of features, these are from left to right: convex corner, right taper, cylindrical surface, square groove, cylindrical surface, and facing. In this case, the second highest X value belongs to the referenced feature itself "convex corner" as its start point



and the end point of the right taper as well. Whereas, by scanning the right side of the convex corner in figure 5.5b as a referenced feature, the system records the following bundle of features: convex corner, right taper, left shoulder cylindrical surface, right shoulder, and facing. The second highest X value in this bundle exists at three features (left shoulder cylindrical surface, and right shoulder). These two cases would affect the decision of the system in terms of which volume should be removed first from the blank and consequently which features take the priority in the manufacturing.



(a)



(b)

Figure 5.5 Two different examples where the second highest X value can be

***Step five: specifying faces of intermediate features***

The purpose of this section is solving issues of features intersections, following the concept of removing volumes from a blank in a reasonable sequence. Each volume means either an independent final feature or an intermediate one. Whilst faces of final features are recognised using the proposed smart interactive feature recognition system, those of intermediate ones need to be defined in this step. The creation of intermediate features and their faces is based on two factors: the faces of the first and last feature is in the bundle from step three and the second highest X value taken from step four. These two factors will help to draw the boundaries of only one intermediate feature. However, this depends on the geometrical and topological data of the design. For example, regarding the part design in figure 5.5a and as mentioned in step four, the right side of the convex corner is scanned: convex corner, right taper, cylindrical surface, square groove, cylindrical surface, and facing. The system calculates the first face data of the intermediate feature by checking the feature in the far right side of the bundle, which is in this case is the facing. This is to compare all its X values with the extracted second highest X value and specify its topology. Since all the X values of the facing feature are lower than the extracted second highest X value, the first face data will be calculated as follows:

- F1 is a vertical line;
- F1 start point X = the observed X value from step one;
- F1 start point Z = maximum Z value of the feature at the far right side of the bundle;
- F1 end point X = the second highest X value;
- F1 end point Z = maximum Z value of the feature at the far right side of the bundle.

In order to facilitate the calculation of the second face of the intermediate feature, the third face is found first. Whilst the system considers the feature on the far right side in the bundle to find the first face data, the feature on the far left is used to find the third face. The same logic of checking and comparing the X values of the feature with the second highest X value is applied. In this case the feature is the convex corner and its X end point is equal to the observed X value, whilst its X start point is equal the second highest X. Hence, the third face will take exactly the same geometrical and topological data of that shape. The third face data in this example will be:

- F3 is a convex curve;
- F3 start point X = the start point X value of the convex corner;
- F3 start point Z = the start point Z value of the convex corner;
- F3 end point X = the end point X value of the convex corner;
- F3 end point Z = the end point Z value of the convex corner;
- F3 centre point X = the centre point X value of the convex corner;
- F3 centre point Z = the centre point Z value of the convex corner;
- F3 direction = CCW.

Regardless the shape and data of the first and third faces of an intermediate feature, the following relation is always true:

$$F1 \text{ end point } X = F3 \text{ start point } X$$

This has only one meaning "*the second face that links the first and third ones is always a horizontal line*". The second face data can be concluded as:

- F2 is a horizontal line;
- F2 start point X = F1 end point X;
- F2 start point Z = F1 end point Z;
- F2 end point X = F3 start point X;
- F2 end point Z = F3 start point Z.

Figure 5.6 shows an example of an intermediate feature with its three faces. The volume of this feature, which is shaded in light green, will be the first to be removed from the original blank that is shaded with light purple.

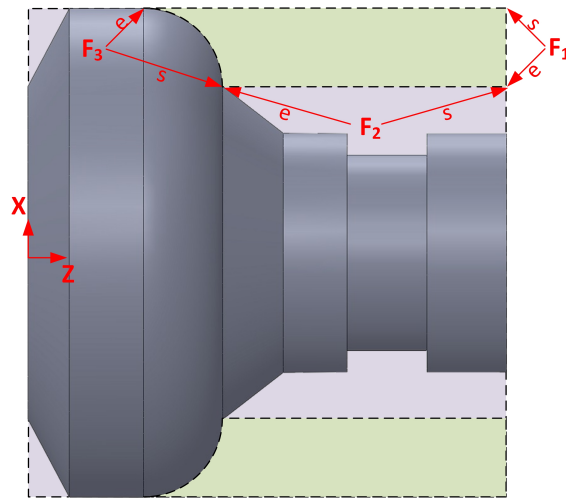


Figure 5.6 An example of an intermediate feature with its three faces

The part design in figure 5.5b can be utilised to show a different example of calculating an intermediate feature. Whilst all the three faces follow the same logic in finding their data, the third face does not take all the original data of the convex corner as in the previous example. This is because the second highest X value is higher than start point X value of the convex corner feature; hence, the start point X and Z of the third face is calculated based on a perpendicular line that passes the point with the second highest X value and intersects the convex corner feature. The third face data in this case will be calculated as follows:

- F3 is a convex curve;
- F3 start point X = the second highest X value;
- F3 start point Z = is calculated using a circle equation;
- F3 end point X = the end point X value of the convex corner;
- F3 end point Z = the end point Z value of the convex corner;
- F3 centre point X = the centre point X value of the convex corner;
- F3 centre point Z = the centre point Z value of the convex corner;
- F3 direction = CCW.

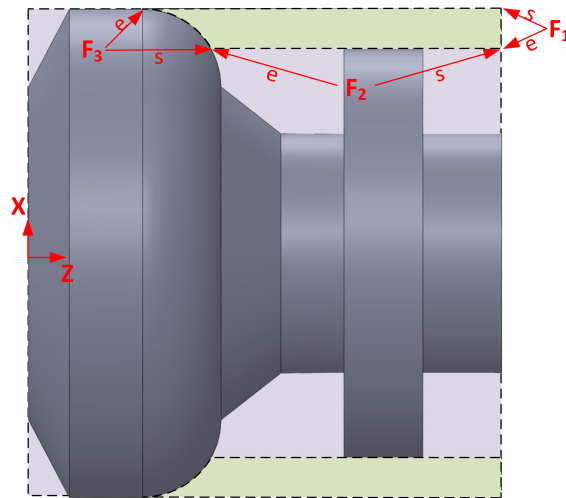


Figure 5.7 A special case of an intermediate feature

Figure 5.7 shows how the third face of the intermediate feature is considered. However, it should be noted that the intermediate feature does not cover the whole shape of the original convex corner feature. The system updates the data of this feature in which the start point X and Z of the updated convex will be the same of the original one, but the end point X and Z will take the values of the F3 start point X and Z. Then, the updated convex will be considered in the next loop of features intersections. This can be noticed in figure 5.8 where only a part of the convex has been removed with the first intermediate feature from the blank. The rest of it will go through the next loop of checking features intersections.

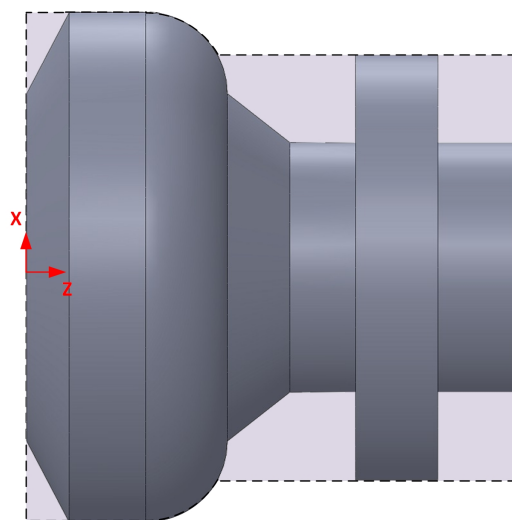


Figure 5.8 A part and its blank after removing the first intermediate feature

***Step six: continuing the loop***

The same process as that described above is carried out in a loop, for each step determining the intermediate features to be removed from the blank, until all the final features are covered. For example, after defining the faces of the intermediate feature in figure 5.8, the same bundle of features (updated convex corner, right taper, left shoulder cylindrical surface, right shoulder, and facing) will be reassessed according to step one. This time the observed X value will be the maximum X value in this bundle. In this manner steps 1-6 are repeated, and figure 5.9a, b, c, and d show an example of this process for this particular design.

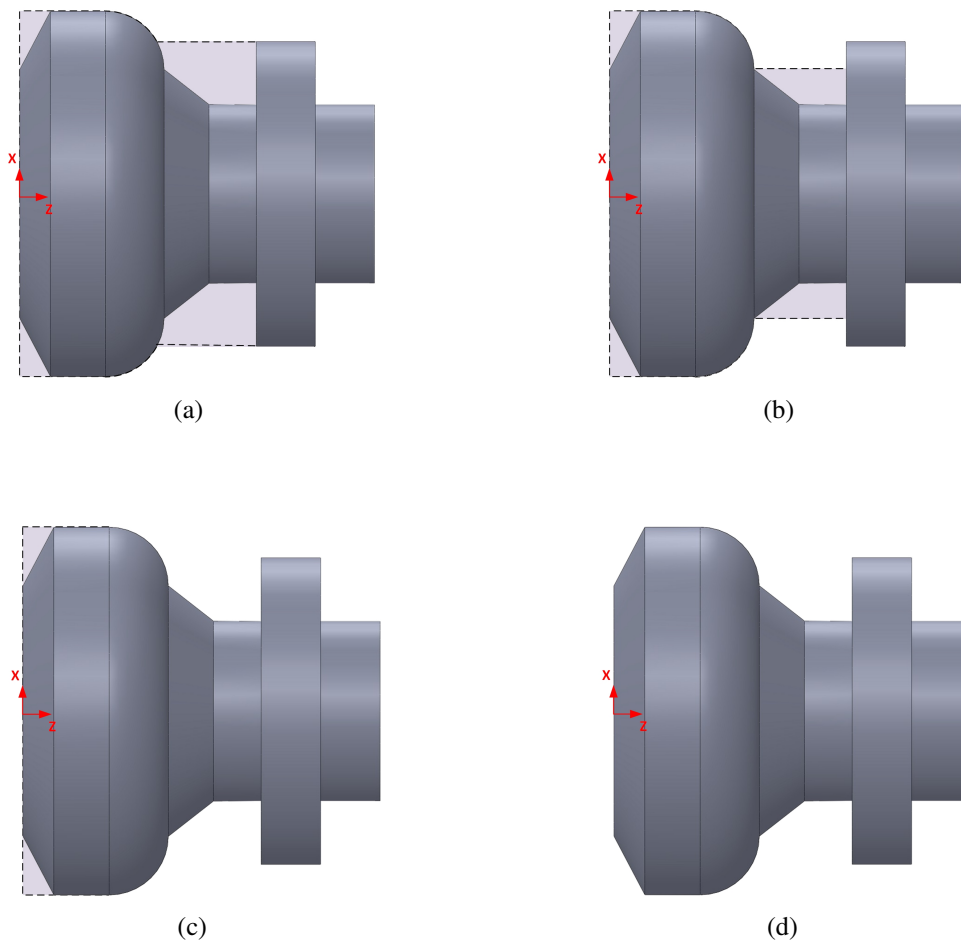


Figure 5.9 Steps of machining intermediate and final dependent features (a) second, (b) third, and (c) fourth, and (d) final shape

***Step seven: avoiding repetition***

In order to avoid the same intermediate features, those that have already been determined are stored in temporary memory of the system. In figure 5.10, the scanning process to the left side of the left taper feature would give features: left taper, facing, concave corner, cylindrical surface, square groove, cylindrical surface, facing, and convex corner. Also, the scanning process to the right side of the convex corner feature would give features: convex corner, facing, cylindrical surface, square groove, cylindrical surface, concave corner, facing, and left taper. Although these bundles are in reversed order, the same intermediate features will be detected. The system can check these instances of duplication, to ensure that there is no repetition of work.

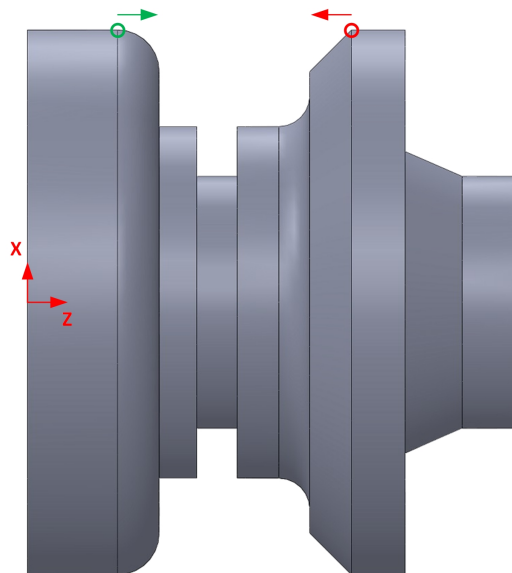


Figure 5.10 A part includes a duplication case of an intermediate feature

In contrast to the final features that can be formed from one to five faces, and take any shape, the intermediate features will only have certain types of 16 shapes. This is because each is always formed from three faces and the second face is always a horizontal line. Figure 5.11 shows all the possibilities that an intermediate feature can take.

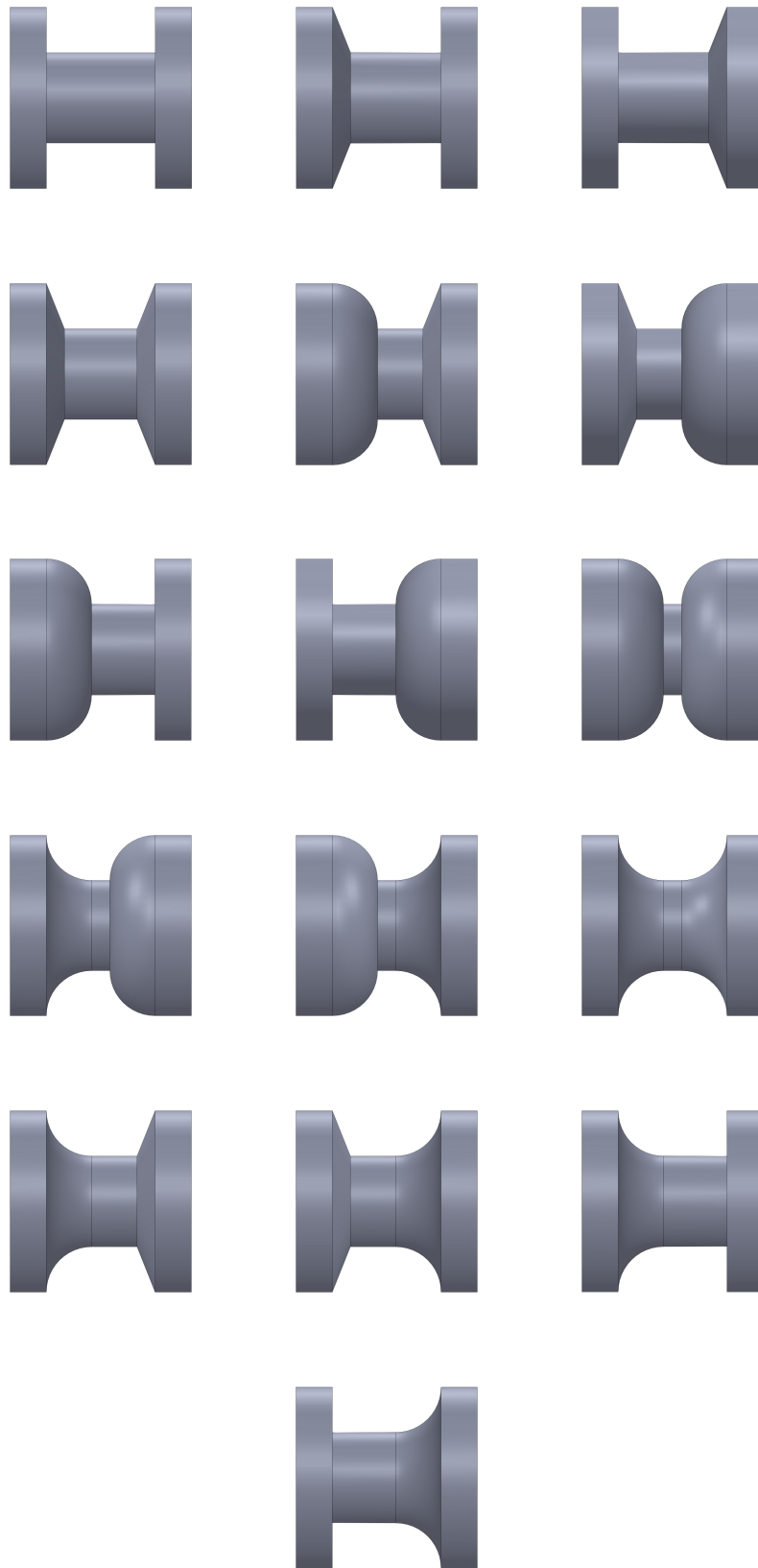


Figure 5.11 The sixteen possibilities of intermediate features



### 5.3.2 Features subtraction of internal shapes

Internal and external features are described in a different way, and the requirements to manufacture them is different. However, for solving features intersecting issues they both follow the same logic, with minor adjustments. In the case of internal shapes, the blank size is not considered. Instead, the minimum X value of the features of an internal shape is deducted. The minimum X value is observed, rather than the maximum that is used in step one. Also, instead of then using the second highest X value, the second lowest X value is used. In figure 5.12 the sequence of creating intermediate features for internal shapes is shown.

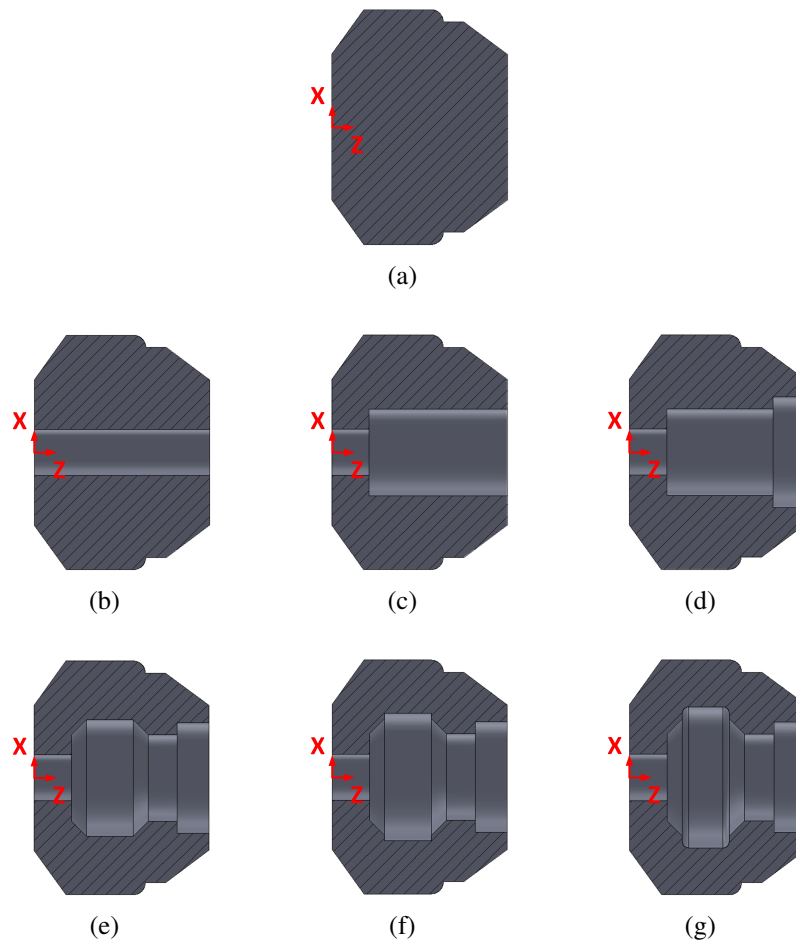


Figure 5.12 The sequence of forming an internal shape

## 5.4 Case study

The proposed features subtraction system is built using C# language, and is considered as a continuation of the SI-AFR system. In order to evaluate its ability, the same case study from chapter four (see figure 4.18) has been tested. The design model contains an external and three internal shapes to show that the proposed features intersection methodology can adequately deal with both situations. The result from the SI-AFR system shows that the external shape of the part design has 15 final features, and the through, front, and back internal shapes have four, two, and two final features, respectively. The following sections provide detail about both the external and internal shapes in terms of solving features intersection.

### 5.4.1 External shape

The SI-AFR system recognises fifteen external final features of the part design and puts them in order from right to left based on the Z axis. Table 4.3 gives details of these features, which are extracted to be used later in the developed features intersection system. The first step of scanning the final features declares that the convex (F9) has the maximum X value (87.5mm) at its apex, which is the same as the blank radius. Since there is only one feature with the maximum X value at this point, step two is overridden directly moving to the third step, which is scanning the final features on the right side of the observed feature. As a result of step three, nine final features are considered on the right side of the convex feature (facing, right convex corner, cylindrical face, radial square groove, cylindrical face, right taper, facing, cylindrical face, and the right half of the observed convex feature). The system then reads the data of these nine features to find the second maximum X value, which is in this case the joint point between the cylindrical face and the convex (F8 and F9) with an X value of (80mm). Hence, the three faces of the first intermediate feature will be:

#### *First face*

- F1 start point X = 87.5;
- F1 start point Z = 100;
- F1 end point X = 80;

- F1 end point  $Z = 100$ ;
- The movement from start to end point is linear.

*Third face*

- F3 start point  $X = 80$ ;
- F3 start point  $Z = 54$ ;
- F3 end point  $X = 87.5$ ;
- F3 end point  $Z = 46.5$ ;
- The movement from start to end point is circular;
- F3 centre point  $X = 80$ ;
- F3 centre point  $Z = 46.5$ ;
- The direction of movement is counter clockwise.

*Second face*

- F2 start point  $X = 80$ ;
- F2 start point  $Z = 100$ ;
- F2 end point  $X = 80$ ;
- F2 end point  $Z = 54$ ;
- The movement from start to end point is linear.

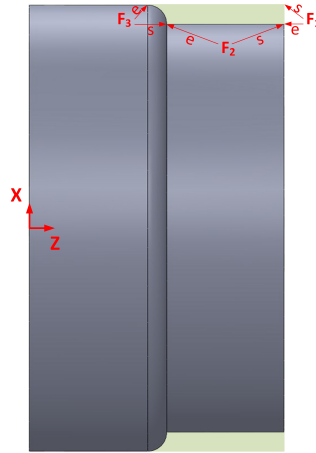


Figure 5.13 Removing the first intermediate feature

As appears in figure 5.13, the first intermediate feature not only removes part of the blank but also covers the machining of the cylindrical face (F8) and the right half of the convex (F9). Although the system scans the left side of the observed convex feature at this point, it is preferable to the reader to continue the loops on the right side of the convex till all the final features are covered. The bundle of nine features is updated to be eight by subtracting the convex (F9) because its right half was totally covered by the first intermediate feature. Repeating the five steps of the system will give the following result:

1. The facing and cylindrical face (F7 and F8) have the same maximum X value (80mm);
2. Scanning the right and left sides of the cylindrical face (F8) will stop directly because the two neighbours have maximum value of (80mm). Thus, the cylindrical face is considered a final feature, and it was already covered by the first intermediate feature;
3. Scanning the right side of the facing (F7) gives a bundle of seven final features including the observed feature;
4. The second highest X value is the joint point between the facing and right taper features (43mm);
5. The three faces of the second intermediate feature will be:

- ***First face***

F1 start point X = 80;

F1 start point Z = 100;

F1 end point X = 43;

F1 end point Z = 100;

The movement from start to end point is linear.

- ***Third face***

F3 start point X = 43;

F3 start point Z = 58;

F3 end point X = 80;

F3 end point Z = 58;

The movement from start to end point is linear.

- ***Second face***

F2 start point X = 43;

F2 start point Z = 100;

F2 end point X = 43;

F2 end point Z = 58;

The movement from start to end point is linear.

6. Scanning the left side of the facing (F7) will stop directly because the next neighbour has the same maximum X value (80mm).

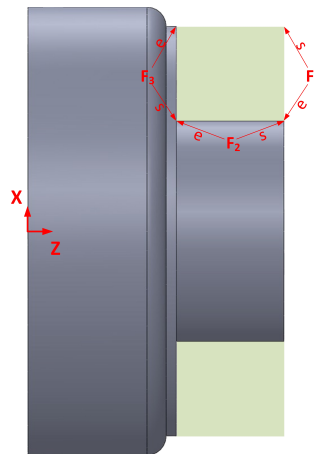


Figure 5.14 Removing the second intermediate feature

Removing the second intermediate feature covers the machining of the facing (F7), as shown in figure 5.14. The loop continues, and this time the bundle of the seven final features is updated by subtracting the facing feature and keeping the other six. Applying the first five steps and considering the updated bundle gives the following result:

1. Only the right taper (F6) has the maximum X value (43mm);
2. Scanning the left side of the right taper will stop directly;
3. Scanning the right side of the right taper gives a bundle of six final features including the observed feature;
4. The second highest X value is the joint point between the right taper and cylindrical face features (40mm);
5. The three faces of the second intermediate feature will be:

- ***First face***

F1 start point X = 43;

F1 start point Z = 100;

F1 end point X = 40;

F1 end point Z = 100;

The movement from the start to end point is linear.

- *Third face*

F3 start point X = 40;

F3 start point Z = 64;

F3 end point X = 43;

F3 end point Z = 58;

The movement from the start to end point is linear.

- *Second face*

F2 start point X = 40;

F2 start point Z = 100;

F2 end point X = 40;

F2 end point Z = 64;

The movement from the start to end point is linear.

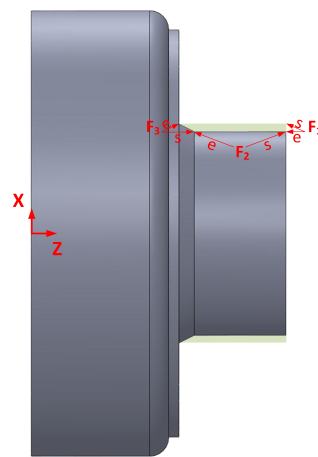


Figure 5.15 Removing the third intermediate feature

Figure 5.15 shows that the third intermediate feature covers the machining of three final features: the right taper and two cylindrical faces (F6, F5, and F3). There are three remaining features: (i) radial square groove, (ii) right convex corner, and (iii) facing (F4, F2, and F1, respectively). Scanning the right and left side of each remaining feature will stop directly; hence, it is declared as a final independent feature and their data is taken from the original result from the AFR system (table4.3).

Figures 5.16 a and b shows the machining of the right convex corner and radial square groove, respectively. At this point all the features on the right side of the convex (F9) are covered.

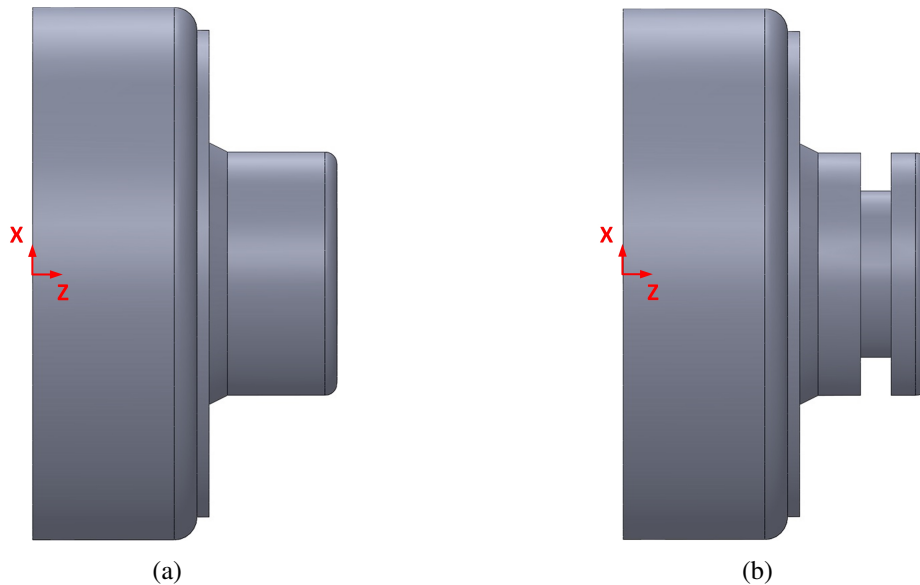


Figure 5.16 The machining process (a) machining of the right convex corner, and (b) machining of the radial square groove

Back to the left side of the convex (F9), the scanning process considers seven final features: left half of the convex, cylindrical face, radial groove with two rounded corners, cylindrical face, two sides tapered groove, cylindrical face, and facing (F9, F10, F11, F12, F13, F14, and F15, respectively). As mentioned, the maximum X value of this bundle is (87.5mm) at the apex of the convex feature, whereas the second highest X value is (80mm), which is the meeting point between the convex and cylindrical face (F9 and F10). The three faces of the fourth intermediate feature will be:

***First face***

- F1 start point X = 87.5;
- F1 start point Z = 46.5;
- F1 end point X = 80;
- F1 end point Z = 39;
- The movement from start to end point is circular;



- F1 centre point  $X = 80$ ;
- F1 centre point  $Z = 46.5$ ;
- The direction of movement is counter clockwise.

#### *Third face*

- F3 start point  $X = 80$ ;
- F3 start point  $Z = 0$ ;
- F3 end point  $X = 87.5$ ;
- F3 end point  $Z = 0$ ;
- The movement from start to end point is linear.

#### *Second face*

- F2 start point  $X = 80$ ;
- F2 start point  $Z = 39$ ;
- F2 end point  $X = 80$ ;
- F2 end point  $Z = 0$ ;
- The movement from start to end point is linear.

Figure 5.17 shows that machining the fourth intermediate feature covers the left half of the convex and three cylindrical faces (F9, F10, F12, and F14). The three remaining features are: (i) radial groove with two rounded corners, (ii) two sides tapered groove, and (iii) facing. Scanning the right and left side of each remaining feature will stop directly, and the feature is declared as a final independent feature. Figures 5.18 a and b show the machining of the radial groove with two rounded corners and the two sides tapered groove, respectively. This is the end of the external shape when all the initial 15 features are covered in 10 intermediate and independent features, as it is shown in table 5.1.

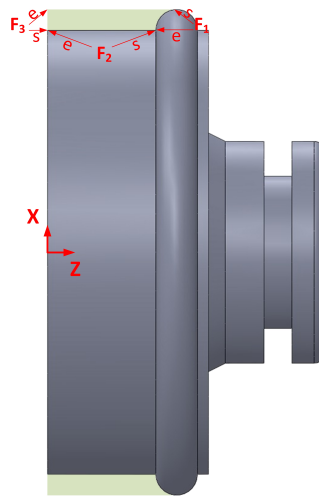


Figure 5.17 Removing the fourth intermediate feature

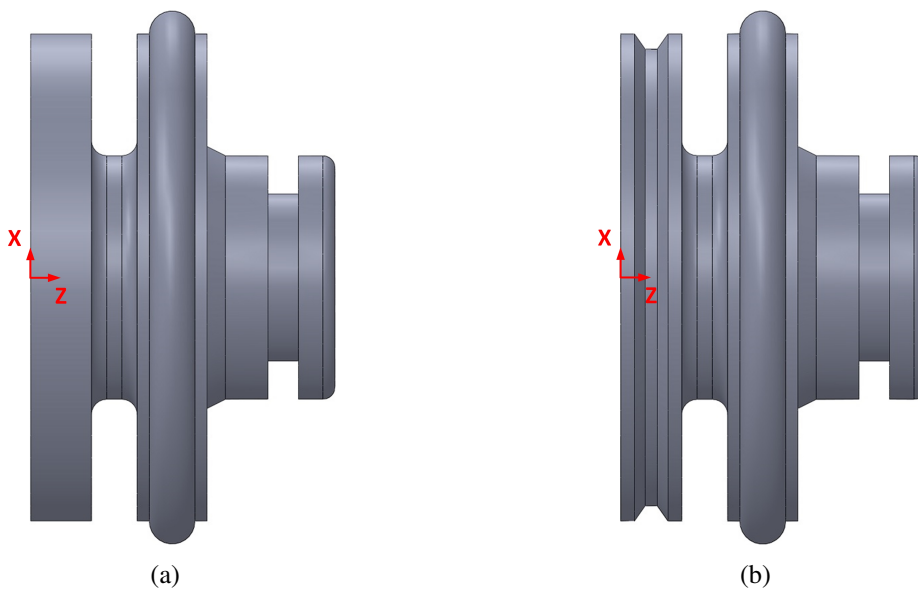


Figure 5.18 The machining process (a) machining of the radial groove with two rounded corners, and (b) machining of the two sides tapered groove



### 5.4.2 Internal shape

The SI-AFR system recognises three internal shapes of the part design with different positions and numbers of features: (i) through internal shape (four final features), (ii) front internal shape (two final features), and (iii) back internal shape (two final features), see table 4.4. Despite the position of the internal shape, the same logic is followed in solving features intersecting. As mentioned, only the through internal shape requires an addition step as a preparation before starting the expected procedure.

#### *Through internal shape*

The first step of scanning the final features of this internal shape declares that the minimum X value is (12mm), which is the minimum radius value between the axis of the shape and any of its faces. Before moving to the second step of grouping the features, the observed X value is used to find the data of the first intermediate feature that forms the main hole of the shape. The length of this intermediate feature is calculated from the far right to the far left of the shape, and its radius takes the same value of the observed X value. The three faces of the first intermediate feature are:

#### *First face*

- F1 start point  $z = 100$ ;
- F1 radius start point  $Z = 0$ ;
- F1 end point  $z = 100$ ;
- F1 radius end point  $z = 12$ ;
- The movement from start to end point is linear;
- F1 X centre = 0;
- F1 Y centre = 0.

#### *Third face*

- F3 start point  $z = 0$ ;
- F3 radius start point  $Z = 12$ ;

- F3 end point  $z = 0$ ;
- F3 radius end point  $z = 0$ ;
- The movement from start to end point is linear;
- F3 X centre = 0;
- F3 Y centre = 0.

#### *Second face*

- F2 start point  $z = 100$ ;
- F2 radius start point  $Z = 12$ ;
- F2 end point  $z = 0$ ;
- F2 radius end point  $z = 12$ ;
- The movement from start to end point is linear;
- F2 X centre = 0;
- F2 Y centre = 0.

Figure 5.19 shows that the machining of the first intermediate feature covers two cylindrical faces (F2 and F4). The two remaining features are the right axial groove and the two sides tapered groove. Scanning the right and left sides of each remaining feature will stop directly and the feature is declared as a final independent feature. Figure 5.20 a and b show the machining of the two remaining features.

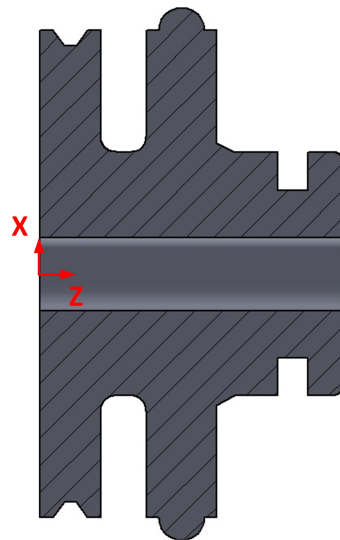


Figure 5.19 Removing the first intermediate feature from the through internal shape

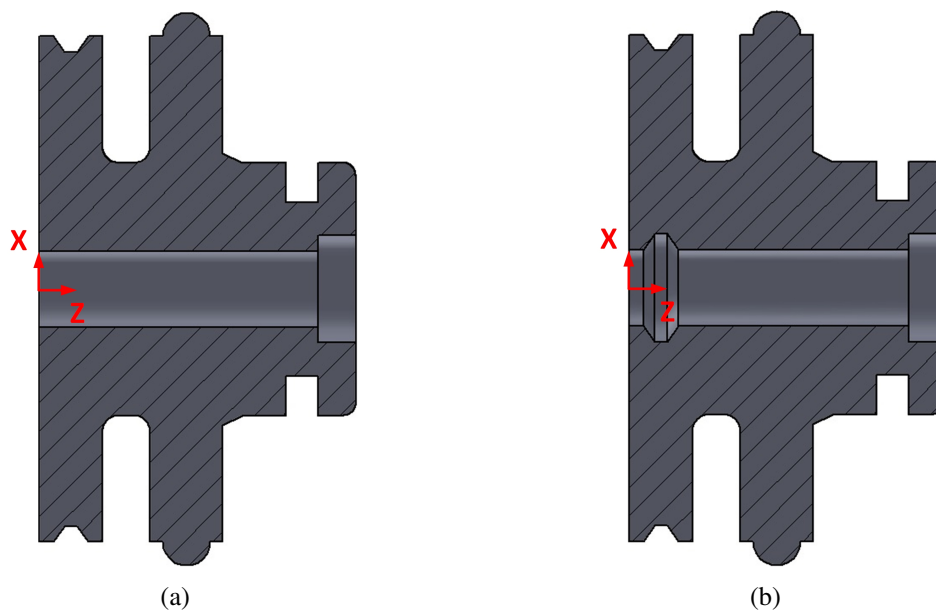


Figure 5.20 The machining process (a) machining of the right axial groove, and (b) machining of the two sides tapered groove

### *Front internal shape*

The expression of *Front internal shape* has one meaning "this internal shape has a closed end, and the centre of this closed end, which is always zero, represents the first observed minimum X value". Such a case means skipping step two of grouping features based on the observed X value because there will always be one feature holding this value, and in this example it is the right axial groove

(F2). At this point, the scanning process is only required to be on the right side of the feature; hence, the bundle includes two features: (i) right axial groove (F1), and (ii) right axial groove (F2). The second minimum X value is the connecting point between the first and second features. The three faces of the first intermediate feature are:

*First face*

- F1 start point  $z = 58$ ;
- F1 radius start point  $Z = 0$ ;
- F1 end point  $z = 58$ ;
- F1 radius end point  $z = 6$ ;
- The movement from start to end point is circular;
- F1 X centre = 0;
- F1 Y centre = 60.

*Third face*

- F3 start point  $z = 46$ ;
- F3 radius start point  $Z = 6$ ;
- F3 end point  $z = 46$ ;
- F3 radius end point  $z = 0$ ;
- The movement from start to end point is linear;
- F3 X centre = 0;
- F3 Y centre = 60.

*Second face*

- F2 start point  $z = 58$ ;

- F2 radius start point  $Z = 6$ ;
- F2 end point  $z = 46$ ;
- F2 radius end point  $z = 0$ ;
- The movement from start to end point is linear;
- F2 X centre = 0;
- F2 Y centre = 60.

Machining the first intermediate feature covers the right axial groove (F2), and the remaining right axial groove is declared as a final independent feature. Figure 5.21 a and b shows the machining of these two features, respectively.

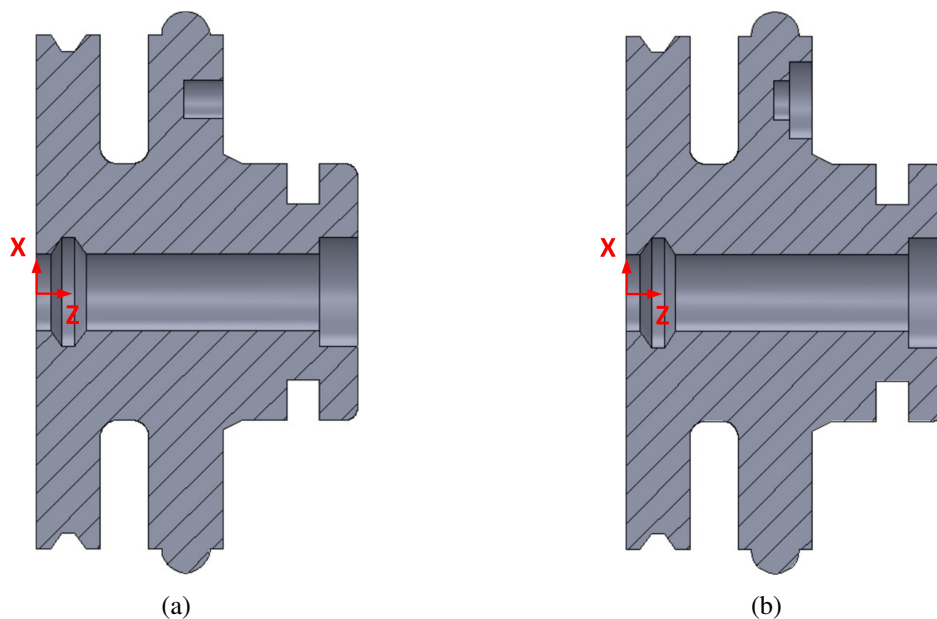


Figure 5.21 The machining process (a) machining of the first intermediate feature, and (b) machining of the right axial groove (F1)

**Back internal shape** The same logic of finding the minimum X value in a front internal shape is followed at a back internal shape. However, instead of scanning the right side of the observed feature, the left side is scanned here. In this example, an intermediate and final feature (left axial groove 2) are declared as it is shown in figure 5.22 a and b, respectively. Table 5.2 shows intermediate and independent features of the through, front, and back internal shapes.



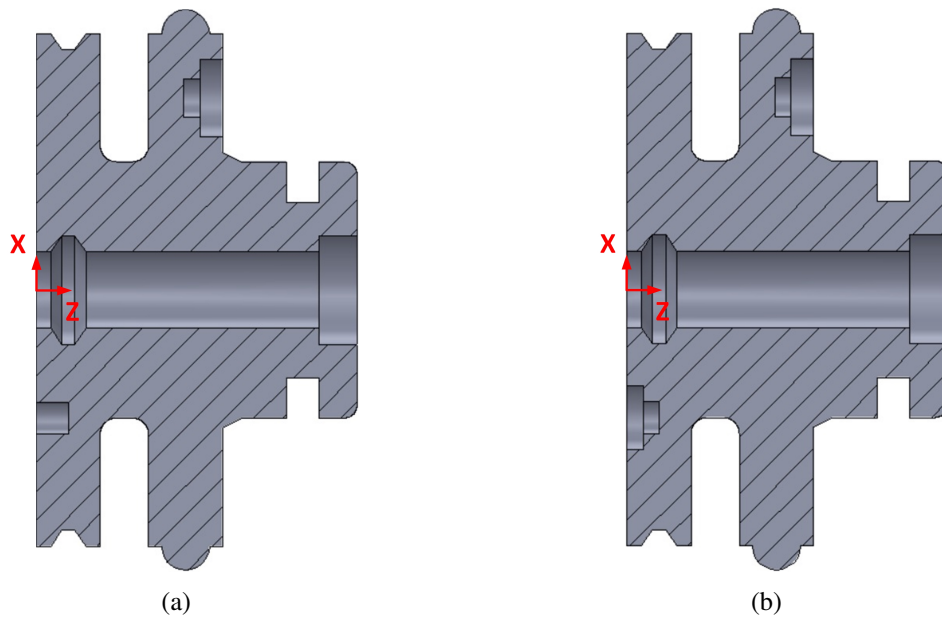


Figure 5.22 The machining process (a) machining of the first intermediate feature , and (b) machining of the left axial groove (F2)

Table 5.2 The details of the recognized internal features

Internal shapes												
Feature No.	Feature Name	Face No.	Xc	Yc	Zcs	RZcs	Zcc	RZcc	Start-to-end	Xcc	CR	Dir.
Through internal shape												
1	Intermediate feature	1	0	0	100	0	100	12	Linear			
		2	0	0	100	12	0	12	Linear			
		3	0	0	0	12	0	0	Linear			
1	Right axial groove	1	0	0	100	17	88	17	Linear			
		2	0	0	88	17	88	12	Linear			
3	Two sides tapered groove	1	0	0	15.5	12	12	17	Linear			
		2	0	0	12	17	8	17	Linear			
		3	0	0	8	17	4.5	12	Linear			
Front internal shape												
1	Intermediate feature	1	0	60	58	0	58	6	Linear			
		2	0	60	58	6	46	6	Linear			
		3	0	60	46	6	46	0	Linear			
2	Right axial groove	1	0	60	58	12	51	12	Linear			
		2	0	60	51	12	51	6	Linear			
Back internal shape												
1	Intermediate feature	1	0	-40	10	0	10	5	Linear			
		2	0	-40	10	5	0	5	Linear			
		3	0	-40	0	5	0	0	Linear			
2	Left axial groove	1	0	-40	5	10	5	5	Linear			
		2	0	-40	5	10	0	10	Linear			

Figure 5.23 shows the resulting window after applying the proposed features subtraction system. Regarding the external shape, the fifteen recognised features from the SI-AFR system are formed by machining ten intermediate and final dependent features resulting from the features subtraction system. Whilst the through internal shape has four initial features that is formed by three intermediate and final dependent features, each of the front and back internal shapes have two initial features, and is formed by two intermediate and final dependent features.

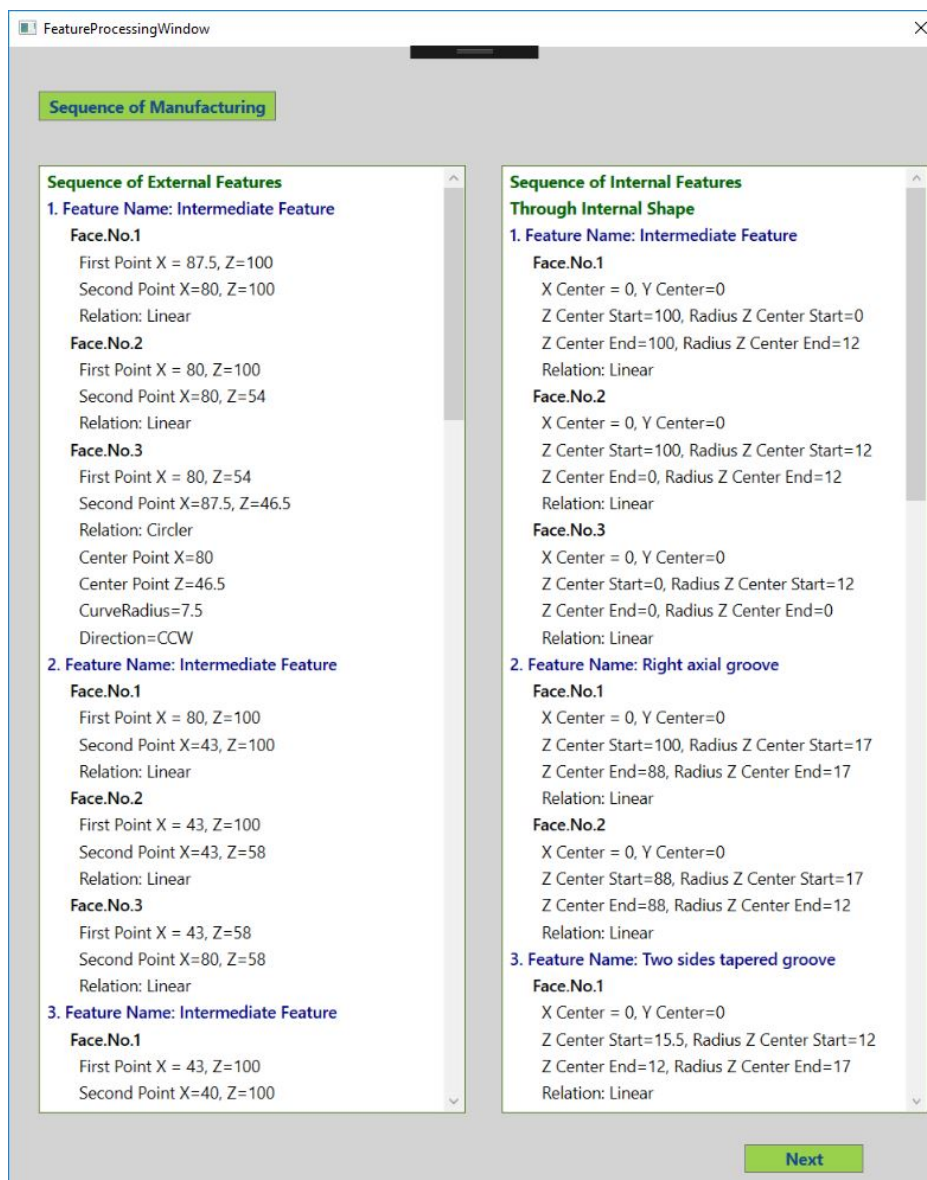


Figure 5.23 The main window

## 5.5 Discussion and summary

The proposed SI-AFR system can recognise predefined and new features, as well as solving simple features intersecting issues. However, complex cases of manufacturing features intersections require more investigation. This is to reduce the machining time and cost, and avoid technical issues, such as damaging a feature and/or interrupting the cutting tool movement. Such a method avoids unnecessarily wasting resources in terms of material, and damaged tools, which would otherwise need to be replaced sooner. These efforts combined, contribute greatly towards efforts in sustainability. Depending on the part design and the process planner vision, different strategies can be suggested to solve a case of features intersecting. These solutions might vary between unreliable, acceptable, and optimal.

In this chapter, a “features subtraction” system has been developed to present an optimal solution for each features intersecting case. The recognised features from the SI-AFR system, which are final manufacturing features, are used as an input of the proposed system. The system has seven steps of analysing, processing, and calculating. The first step scans all the part’s features to find these that have maximum X value. Then, the system checks if the observed features are linked to each other and puts them in groups. The third step includes scanning the final features on the right and left side of each observed feature and putting them in bundles. However, if the scanning process on both sides is directly stopped, then the feature is declared as a final independent one. Each bundle of features is then analysed to find a second X value besides the initial one. This is to be used in the fifth step that determines data of an intermediate feature. Each intermediate feature removes a part from the blank and covers the machining of one or more final features. In some cases the intermediate feature covers a part of a final feature; therefore, the system considers the uncovered part of that final feature in the next loop. Step six is for checking the rest of features from each processed bundle and sending them again through the whole process from step1-6. Finally, the system examines all the determined intermediate features in order to avoid machining the repeated ones.

The feature subtracting system follows the same logic in determining intermediate features of external and internal shapes. However, this requires one adjustment which is calculating an intermediate feature based on minimum and second minimum X values in a features bundle of an internal shape; whereas, maximum and second maximum X values are used in external shapes. The

system has been evaluated using a case study that includes an external shape and three different cases of internal shapes. The result provides an optimal solution for features intersecting problems.

## Chapter 6

# Geometric dimensioning and tolerancing

### 6.1 Introduction

One of the essential design information that effects process planning and manufacturing is dimensions. Dimension is a term that specifies a numerical value of a parameter that belongs to the geometry of a mechanical part or assembly [164]. What geometric dimensioning means depends in the first place on one's role. From the designer's point of view, dimensions describe the design intent of an assembly or individual parts. To a process planner, dimensions are considered a guide to generate a sequence of operations and tool path. Also, it is a success or failure criterion of a manufacturer's work. However, manufacturing of components cannot guarantee precise dimensions of a designed part even when using the latest technologies, and hence tolerances are required [80]. The tolerances, which are specified by a designer, represent the acceptable range of deviation of dimensions [73]. Traditionally, the tolerances were simply added to the dimensions and considered as critical for fit and function (linear tolerancing). This concept is no longer satisfactory as the recent technology revolution and the use of geometric dimensioning and tolerancing (GD&T) have helped develop the idea that features of a part need to be controlled in different characteristics, such as size, form, orientation, location, profile, and runout [165]. This is to optimize the production by minimizing the cost and time of manufacturing and reduce scrap.

Most current CAD systems support GD&T specification using attributes attached to a designed model. It is possible to add different GD&T information to the corresponding geometric entities

of a design, for example a point, face, edge etc. [79]. Whilst commercial CAD systems support specification of GD&T on 3D models at a presentation level, currently they do not provide semantic representation of such information. This statement reflects the reality of CAD systems regardless of the capabilities of PDE formats that are used in saving and transferring the 3D models. For instance, STEP AP214 and recent AP242 have been developed to include a design's geometrical and topological information as well as GD&T. The availability of rich geometrical and topological data that PDE standards can provide has influenced researchers to integrate design and manufacturing systems using feature technologies. However, saving a design as a STEP file will only keep and transfer geometrical and topological data and neglect GD&T ones, because CAD vendors have not implemented translators for this purpose yet. Available geometrical information cannot be used instead of the indispensable GD&T information in delivering the complete picture of a design to a CAPP system.

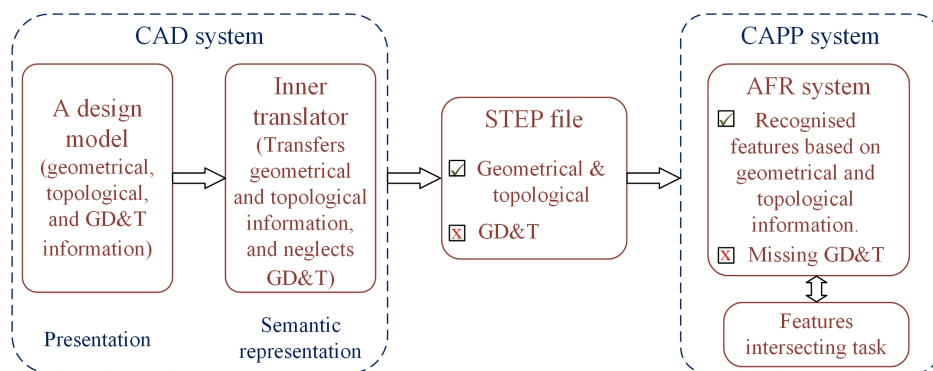


Figure 6.1 A CAD system with its inner translator

This chapter describes the attempt to automatically extract GD&T data from a CAD model as a semantic representation, and merge it with the recognised intermediate and final features from the developed AFR system that is described in this thesis. However, the automatic dimensioning and tolerancing is out of the scope of this project, as it would require extensive effort and is a PhD in and of itself. Figure 6.1 shows a CAD system with its inner translator which transfers geometrical and topological information of a design model into a semantic representation and neglects GD&T information. Also, it illustrates the important role of STEP files in connecting between CAD and AFR systems.

## 6.2 Problem definition: Semantic representation of GD&T data

The initial versions of the CAD systems were limited to perform 2D sketching presentations. Due to rapid growth and high competition in the manufacturing industry, upgraded versions of these CAD packages were developed, which are able to achieve complex 3D model. However, for both cases of 2D sketches or 3D models, all CAD systems contain detailed information of a design's features which includes geometrical and topological data, saving this data at the representation level. Such functionality provides the user an initial dimensioning of each feature in the part. Figure 6.2a shows a cylindrical part with three different diameters and lengths which are a, b, and c. Saving this part as a STEP file at the representation level provides the value of each length based on its geometrical information, and logically, these values of lengths can be used as individual dimensions (figure 6.2b). Also, it is possible to use tolerance standards to add a tolerance value for each dimension. However, putting dimensions based only on geometrical information and choosing tolerance values from standards might not satisfy the design's requirements, as well as the designer and manufacturer's point of view. This can be illustrated in figure 6.2c when a main datum is chosen and a possible case of practical GD&T is schemed by a designer based on the design's requirements. This case demonstrates the relation between the part's features and different GD&T schemes. Such relation is required by CAPP due to its influence on some tasks, such as determining the machining operations, machining sequence, and tool path.

Regardless of the designer's vision about GD&T during the presentation level, it should be interpreted as a representation model to be used in CAPP systems. In this matter, different representation models have been developed: EXPRESS, surface graph, representational primitive, technologically and topologically related surface (TTRS), unified modelling language (UML), extensive markup language (XML), category theory, Geo-Spelling, and ontology-based models. Among the mentioned representation models, the most widely used in industry is the EXPRESS model in the form of STEP files. However, the use of EXPRESS establishes for correct dimensions and tolerance at the representation model, but it does not explicitly provide semantic information. This is because so far CAD systems do not support this functionality. The process of extracting GD&T schemes from 3D models, linking each dimension and tolerance with recognised features, and representing them in



a semantic and computer-understandable way is necessary in generating a complete automatic and smart CAPP system.

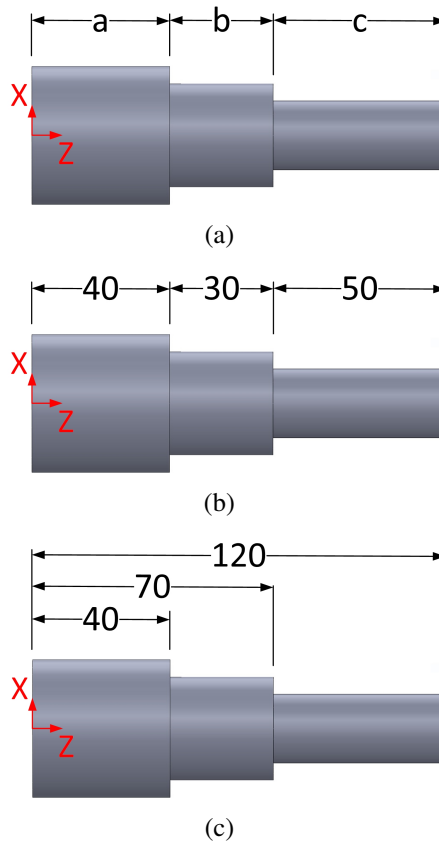


Figure 6.2 A cylindrical part with three different diameters and lengths (a) the representation of the cylindrical part in the design phase, (b) dimensions as understood by the STEP file, (c) dimensions specified by the designer

### 6.3 Investigation of the solution

Numerous CAD systems and PDE standards save and transfer GD&T information with varying levels of detail. The first step was to find which of these could provide sufficient GD&T information, in order to pass this data to the CAPP system. Whilst STEP files AP203, AP214, and AP242 are able to transfer geometrical and topological information of a design, only the last version of AP214 and AP242 were developed to provide GD&T information. While Creo (Parametric Educational Edition M020) and Solidworks (Education Edition/SP3 2017) exports to STEP AP242 file, which includes 3D product and manufacturing information, so far both have not developed the capability to include

this additional information. More precisely, while dimensions and tolerances are added to a 3D model and saved as a STEP 242 file, these values are floating numbers and do not refer to specific datums, and is therefore not useful in its current form. A workshop of researchers was created in an attempt to solve this complex issue and to connect the floating numbers of dimensions and tolerances to specific surfaces, by inferring this from the available geometrical data. The researchers included an applications consultant from PTC, and an expert of computer-aided analysis of dimensions and tolerances, who has published numerous papers on the topic [80, 166–168]. While working on this together for six months, it was possible to link each dimension with its tolerance, following the same strategy of tracking geometrical and topological data that is used in the STEP file parser that was developed for this research. Details of how the parser works is described in 4.2. This was possible because this information is presented in a logical hierarchical format, making the connections between tolerance and dimension possible. But during this six month investigation the link between a dimension and its corresponding surface could not be determined.

## **6.4 Discussion and summary**

Dimensions and tolerances are indispensable requirements to describe a design intent. Although CAD systems have this at the representation level, these are missing when exporting the design via PDE files. It is necessary for CAD vendors to develop an internal translator that attaches such information to the STEP 242 file, as it has the ability to store this information. It is also possible with further work for individuals to create a tool that links the dimensions and tolerances to its respective datum.

# Chapter 7

## Results and conclusion

### 7.1 Introduction

This chapter explains the indispensable requirement of developing a smart AFR system and solving features intersecting issues, in creating a reliable ACAPP system. This is by illustrating how the rich geometrical and topological data of the recognised final and intermediate features can be used in automatically generating the expected seven outputs of ACAPP, as well as providing some strategies of calculating them. However, this chapter is not intended to optimise these outputs since its main current objective is showing how each of the outputs completely or partly relies on the results from the previous chapters. Also, it shows how ACAPP instructions can be provided to the user in terms of a report and documentation.

### 7.2 Launching the ACAPP system

The developed ACAPP system starts with showing the minimum requirement of the blank size in terms of diameter and length via an interface window. Then, it allows the user to enter the diameter and length of the available blank in the workshop, and these should be bigger or at least the same size that is required, otherwise the system will reject the measurement(s).

The next step includes scanning the same STEP file, which is used as an input to the SI-AFR system, to find the material that is specified by the designer. This is by finding the entity line that

contains the 'material name' statement, and observing the first pointer of it. Then, the explicit material name will be found in the entity line of the observed pointer. However, as mentioned in chapter 6, it depends on the CAD vendors if their inner translator extracts the material information from the design model and attaches it to the STEP file. For example, if the design model is saved as STEP file AP203 or AP214 using SOLIDWORKS Education Edition/SP3 2017, the material information would be lost; whereas, if the process is achieved using Creo Parametric Educational Edition M020, the information is retained. This is a good example that shows how the information in STEP files does not only depend on how the file is designed, but also on the ability of the inner translator of a CAD software. This is also the case for surface roughness, where different CAD software do not export this value, though the STEP files can include it. The following example shows two entity lines that are included in the STEP AP203 file of a design model that is saved using Creo and the *ALUMINUM\_WROUGHT* is selected as a material of the design:

```
#1777=REPRESENTATION('material name',(#1775),#1713);  
#1775=DESCRIPTIVE_REPRESENTATION_ITEM('ALUMINUM_WROUGHT', 'Sample materials data from Granta Design\\N\\See grantadesign.com/PTC for more');
```

### 7.3 Process selection

Since this work considers rotational parts that have different external and internal features, five processes are taken into account: external turning, drilling, boring, internal turning, and parting. As mentioned, the current inner translators of CAD software do not transfer GD&T and surface roughness information; therefore, a grinding process is not added to the mentioned processes. The developed SI-AFR system does not only recognise a part's features, it also isolates the external features from internal ones; hence, it facilitates selecting appropriate machining process(es) for each of them. For example, the external turning and parting processes are required for manufacturing the external features; whereas, drilling and internal turning processes are specified for internal features, as well as a boring process in some cases. The selection of processes based on the geometrical and topological data of the recognised features and their position in a part is not an isolated task in CAPP because it can affect and/or facilitate the other tasks. For instance, it would reduce the tool selection

to a specific set that matches the selected process. However, this is not the only interaction between two tasks of CAPP, since it is multidisciplinary and any of its inputs or outputs can affect the others as is shown in the following sections.

## 7.4 Cutting tools

The tool selection process includes the determination of two main parameters: the tool-holder and insert. Specifically, the tool-holder comprises of the clamping system, type, point angle, hand of cut, size, and more. While the insert encompasses shape, size, grade, nose radius etc. These parameters are encoded in the ACAPP system based on ISO standards for cutting tools. Figure 7.1 and 7.2 shows the insert shapes and the holder styles that will be used in this work, respectively.

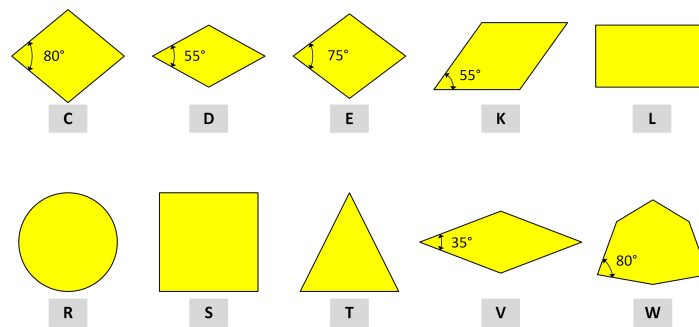


Figure 7.1 Examples of insert shapes that are encoded based on ISO standards

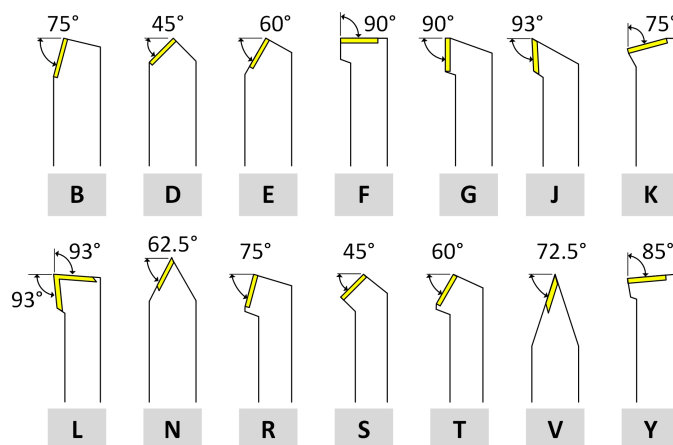


Figure 7.2 Examples of holder styles that are encoded based on ISO standards

The ACAPP system observes three main phases during the tool selection process. The first step is regarding the machining cycle, as to whether it is roughing or finishing cutting. The second step is based on the geometrical and topological information of a feature and its location on the part. The last step of tool selection is related to the machining process since it may be external turning, drilling, boring, internal turning, or parting. A rule-based approach has been developed in order to manage these three overlapping steps by analysing the feature data and location, as well as the required process to manufacture it, during the tool selection process. A fourth phase can be added to this task, which includes the selection of cutting tools based on the material of the model design. However, only Tungsten carbide tools are considered in the current work because they cover a wide range of materials.

#### **7.4.1 Roughing and finishing**

Since the majority of the material is removed during the rough turning process, this requires maximum power of machining. Hence, the most preferred insert in terms of strength is the one with the largest point angle. The insert with  $80^\circ$  of point angle and  $95^\circ$  of approach angle is observed first for roughing unless this point angle conflicts with the feature geometry. Whilst the features subtraction task presents a whole vision of transferring a blank into the desired shape, this is sufficient knowledge for a step-by-step process for rough turning. On the other hand, the SI-AFR system provides full information of a part's features in its final shape, and this data can be adopted directly in the finish turning cycle. The system prepares the tool selection as the following sequence:

- external roughing feature;
- internal roughing features;
- external finishing features;
- internal finishing features.

However, this logic is followed in the tool selection process, but it takes a different order during the sequence of operation task, as will be explained later.

### 7.4.2 Feature location

This task starts by checking if the location of the feature is on the right, left, or middle of the part. Being on the right of the part means that the feature has at least one  $Z$  value that is the same as the maximum  $Z$  value of the blank, whereas being on the left means that the feature has a  $Z$  value of 0. If none of these conditions exists, the feature is in the middle. In order to clarify the role of the feature location on the tool selection, an example of two features is illustrated in figure 7.3. The *right step* feature is located on the right of a part (figure 7.3a), and the *radial square groove* feature is in the middle of a part (figure 7.3b). Although the material that should be removed from each feature is exactly the same regarding the volume and shape, the location of the features affects the decision of tool selection. The entrance to the right step feature is free from the right side; hence an insert with  $80^\circ$  of point angle and a right hand tool-holder are considered. A different set of cutting tools is required to machine the radial square groove feature, which in this case is a square shape insert and a neutral tool-holder.

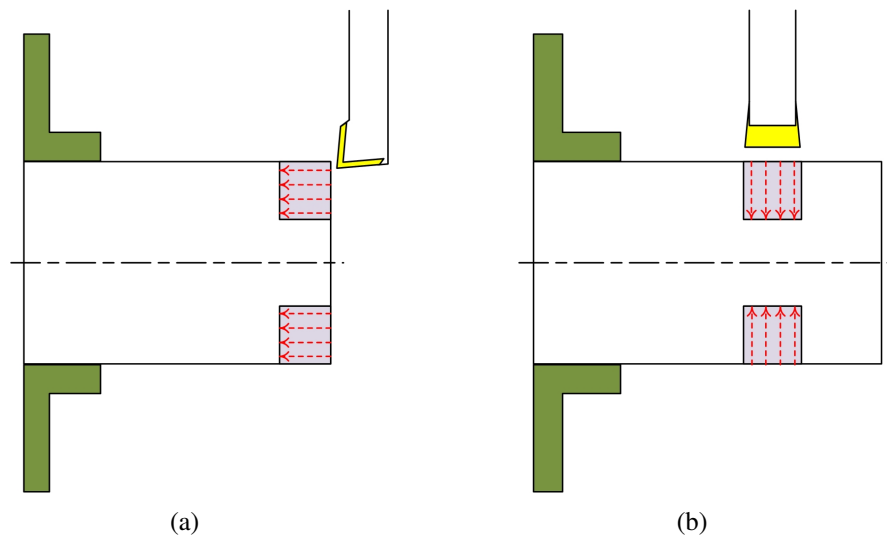
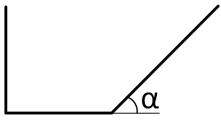
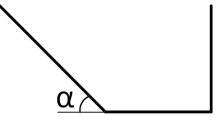
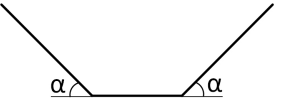
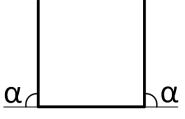
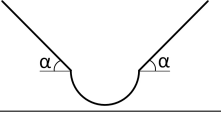


Figure 7.3 The effect of the feature position in selecting cutting tools (a) a right step feature on the right side of the part, and (b) a radial square groove feature in the middle of the part

### 7.4.3 Feature geometry

As mentioned, the insert with  $80^\circ$  of point angle is considered a starting point of the tool selection. However, if this large angle is incompatible with the feature geometry and causes disfigurement, an

Table 7.1 Examples of different grooves geometry and the criteria of choosing appropriate tool parameters

No.	Shape of groove	Control criteria	Insert shape	Tool holder type	Tool hand
1		$0 < \alpha \leq 22$	T	J	Right
		$22 < \alpha \leq 27$	D	J	Right
		$27 < \alpha \leq 44$	V	J	Right
		$44 < \alpha \leq 50$	V	J	Right
2		$0 < \alpha \leq 22$	T	J	Left
		$22 < \alpha \leq 27$	D	J	Left
		$27 < \alpha \leq 44$	V	J	Left
		$44 < \alpha \leq 50$	V	J	Left
3		$50 < \alpha \leq 55$	T	E	Right
		$55 < \alpha \leq 60$	D	N	Right
		$60 < \alpha \leq 70$	V	V	Right
4		$70 < \alpha \leq 90$	S	F	Neutral
5		$75 < \alpha \leq 90$	S	F	Neutral
		concave	V	V	Right

insert with smaller point angle should be examined. The analysis of features geometry goes through a complex rules structure, especially for different groove features and those that are located in the middle or at the left side of the part. There are different strategies to machine the same groove, and each of them might require a different set of cutting tools. Since this work does not aim to optimise the tool selection but explains how it can be achieved based on recognised features, any tool set that can machine the feature will be accepted. Table 7.1 shows examples of different grooves geometry and the criteria of choosing appropriate tool parameters. It demonstrates that the side angle(s) of a groove affects the selection of the insert shape, tool-holder type, and the tool hand. Although the first four examples in the table are simple grooves with different values of side angles, the same results of tool selection are considered if these grooves include rounded or chamfer corner(s) on the base.

As considered in this thesis (mentioned in chapter five), an intermediate feature is formed from three faces, and can be treated as a groove feature depending on its location in the part. Hence, the same criteria that is used in the tool selection for grooves can be used with intermediate features. In



most of the cases of grooves and intermediate features, the selection of one cutting tool can produce the feature. This is particularly if the base of the feature is linear regardless of whether it has rounded or chamfer corners on it. However, if the base of the groove is circular, such as the fifth feature in table 7.1, this feature might require two sets of cutting tools to be machined. In this example, the first set includes a square insert shape, which can machine only the upper half of the feature. The second set has an insert shape with  $35^\circ$  of a point angle and is able to machine the concave part.

The manufacturing of any internal shape starts with a turning or milling process. Although same end-mill cutting tools can be used for either process, they are different in their nature. In a turning process, the chuck holds the workpiece and rotates when the cutting tool is only moving along the machining area; whereas, the cutting tool rotates and moves along the machining area in a milling process. The location of an internal shape is considered in selecting turning or milling processes. The former is selected if the internal shape is located at the axis of rotation, whilst the latter is used for those with an offset from the axis. Figure 7.4 a and b shows an internal shape that is located at the axis of rotation and with an offset from it, respectively. The size of the first end-mill bit is calculated based on the diameter of the first recognised feature in that internal shape. Depending on the size and complexity of the internal shape features, these might require boring and internal turning or milling processes. The internal turning process is selected to machine internal shapes that are located at the axis of rotation. Such a process requires internal turning cutting tools, which are selected following the same strategy and control criteria in the tool selection process for external shapes.

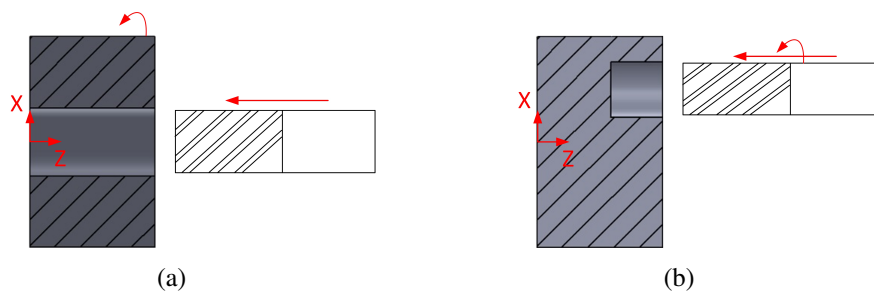


Figure 7.4 The effect of the internal shape location in selecting the cutting process (a) an internal shape is located at the axis of rotation, and (b) an internal shape is located with an offset from the axis of rotation

All the possible geometries of the predefined and the expected new features have been covered in this work regarding the cutting tool selection task. Thus, the ACAPP system can automatically choose

an appropriate tool holder and insert from the tool library. However, new methodologies of selection can be developed based on the recognised final and intermediate features in order to optimise this task in terms of tool life and minimum cost.

## 7.5 Sequence of operations

Both the tool selection process and sequence of operations require a recognition of manufacturing features to be achieved. Whilst the former follows a steady sequence of examination: rough external, finish external, rough internal, and finish internal features, the default sequence in the latter is rough external, rough internal, finish external, and finish internal features. This is to avoid the deformation of the part that might occur due to the redistribution of the internal stresses (manufacturing inheritance). However, the sequence of operations might take different orders or overlap. For example the rough external may be considered first, but not completed, before going on to the rough internal. This depends on the complexity of the design, how a feature recognition system can describe it, and the size of the blank.

The SI-AFR system presents further valuable information besides the main objective of recognising manufacturing features. Describing the location of an internal shape is a good example of this provided information. In order to explain the role of such information in finding an appropriate sequence of operations, the model design of the case study 1 (chapter 4) is taken as an illustration. The part has fifteen external features and three internal shapes, which are declared: through, front, and back shapes. An external turning process in terms of roughing is achieved first for all the external features except the last facing feature, since it requires cutting the part from the blank. Then, a rough machining process is accomplished for the front and through internal shapes using drilling, boring, and internal turning processes. The next two steps include a finish machining for both the external shape and both the through and front internal shapes. The existence of a back internal shape in the part certainly means that the part should be flipped to complete the machining. This is after cutting the part that would produce the last facing feature. Finally, the sequence of drilling, boring, and internal turning is repeated again, and this time to manufacture the back internal shape. However, even this suggested sequence of operations to manufacture this part might be edited depending on other factors.

For example, if the length of the blank is the same as the part's length, some of the external features cannot be manufactured during the first set-up. This is because the chuck of the turning machine clamps the blank in a way that covers some of the features that should be machined. Hence, a different sequence of operations should be considered by the CAPP system. Figure 7.5 shows two different sequences of operations based on the features of the part and clamping zone.

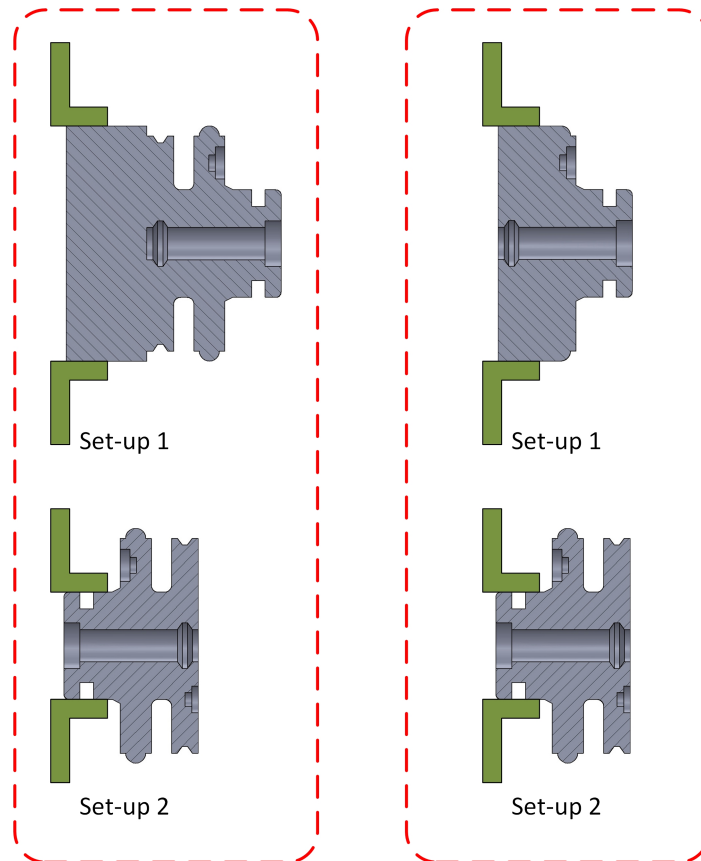


Figure 7.5 Two different sequence of operations strategies based on the features of the part and the clamping zone

## 7.6 Generating G-code

After applying the developed SI-AFR and features subtraction systems, the part's manufacturing features are now fully recognised in terms of geometry and topology. Writing G-code to manufacture any part does not only require deep knowledge about its features, but also the selected process(es), cutting tools, and the sequence of operations. However, these three aspects cannot be considered

independent since building each of them requires a strong base of feature recognition. Based on the set-up numbers that results from the sequence of operations, the ACAPP system might generate more than one G-code file to manufacture one part. These files are presented with a description of how to use them.

The ACAPP system generates each G-code file as a standard structure. The first line presents the program number that starts with a capital 'O' followed by four digits without spaces. The second line includes putting the part's name, which is extracted from the HEADER section of the STEP file, between two brackets. Whilst the program number is changed and takes the sequence of the machining set-up, the part's name stays the same. The third line in the file describes the operation performed by the program which is MACHINE EXTERNAL, MACHINE INTERNAL, or MACHINE EXTERNAL AND INTERNAL . Next, line four specifies the stock diameter entered by the user. The fifth line assigns the distance between the Z-zero and the right end of the workpiece. In this work, the Z-zero reference of the first set-up represents the left end of the part; hence, the distance between the Z-zero and the right end of the workpiece should be at least the same as the part's length. In contrast, if the second set-up includes flipping the workpiece at the machine, the Z-zero reference represents the right side of the flipped part. This means all the Z values should be minus values during the second set-up. Figure 7.6 a and b shows the Z-zero reference in the first and second set-up, respectively. All the mentioned lines do not take numbering since their objective is only to introduce the user with the work.

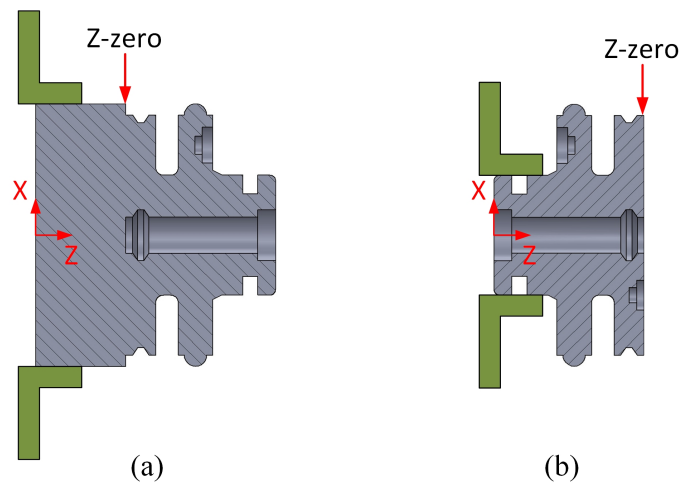


Figure 7.6 Setting the Z-zero point(a) the Z-zero at the first set-up, and (b) the Z-zero at the second set-up after flipping the workpiece

The system generates another set of preparing blocks before starting the code that machines the part. These blocks are numbered and include: making a safe home out move for a tool change, calling the required tool, selecting the working plane, turning on the spindle in a specific speed and direction of rotation, and moving the tool rapidly to a selected point to machine the first feature. In order to facilitate the calculation of the tool path, and consequently writing the G blocks, it is divided into sections. Each section is generated to machine a specific intermediate or final feature. The ACAPP system follows a constant procedure in generating and writing the G code for each section: (i) putting the feature's name between two brackets, (ii) checking if the selected tool is the same that was used with the previous feature or it should be changed, and (iii) specifying the start point of the tool path based on the hand of cut. There are different strategies to machine each final and intermediate feature during the roughing and finishing cycles. Whilst the finishing cycle gives the expected final part, the roughing is more complex in terms of selecting an appropriate machining strategy. This is because most of the material is removed during this phase. A machining strategy means how a cutting tool accesses a feature, moves to form it, and steps out of the feature. The following aspects are considered before selecting each machining strategy:

- the feature geometry;
- the selected tool for machining the feature;

- the tool accessibility to the feature.

Each machining strategy can be applied using different G code. Although adopting G1, G2, and G3 together can machine most if not all the features, this set is considered unpractical, particularly in machining roughing cycles. Part of this demerit is due to the requirement of writing many blocks to cover the machining of one feature. Hence, canned cycles are preferred as replacements of the basic G codes due to their feasibility. In this work, three G canned cycles (G71, G72, and G75) are utilised to cover the roughing phase of the features. Each G canned cycle is used to serve a particular objective based on specific conditions. For example, G71 is chosen for achieving roughing facing to the right end of the workpiece using an insert with 80° of point angle. The canned cycle G72 is assigned to manufacture intermediate and final features that have at least one toroidal or conical surface on the side. Depending on the geometry of the feature, G72 can be used with a wide range of inserts that have different point angle values or square section ones. If the ACAPP system selects a square section insert and G72 to machine a feature, the width of the tool is considered while giving the tool path in the G72 block. Figure 7.7a shows an example of such a case where the feature starts at ( $Z = 20\text{mm}$ ) and the tool path starts at ( $Z = 16\text{mm}$ ) since the tool width is 4mm. However, the same start point of the feature is used in the tool path if the insert shape is a triangle (figure 7.7b). In the case of a square groove, where the both sides are perpendicular to the base, the G75 canned cycle is applied as a machining strategy. Since machining a square groove requires a selection of a square insert shape, the same concept of considering the tool width in the tool path is followed here.

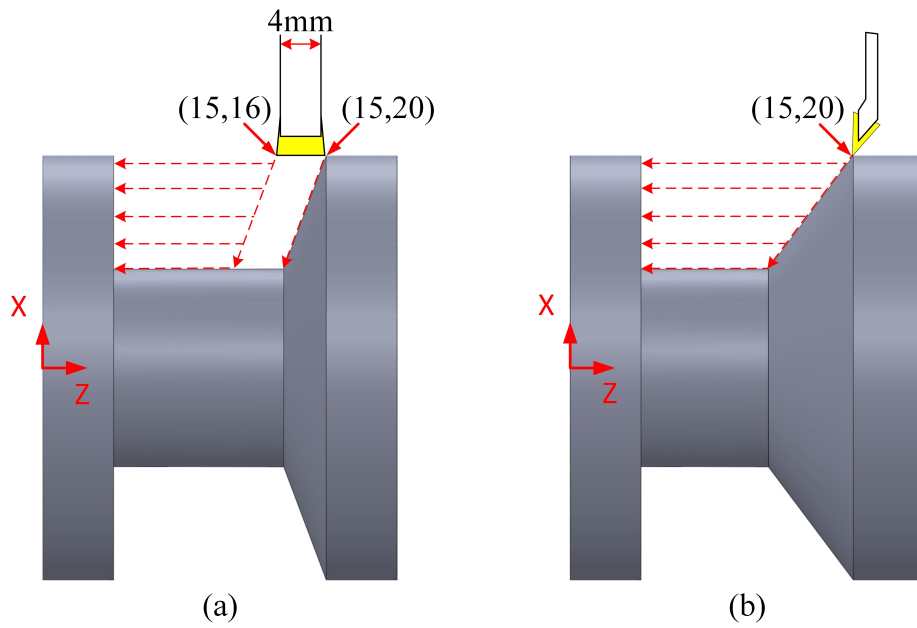


Figure 7.7 Calculating the start point based on the tool geometry (a) a tool path start point based on the tool width, and (b) a tool path start point is equal to that of the feature

## 7.7 Case study

The results of case study 1 from chapter four and five have been used to validate the proposed ACAPP system. This include analysing the geometrical, topological, and location information for each final and intermediate feature to automatically generate process planning outputs.

### **Blank size**

As mentioned, the ACAPP system starts with showing the largest diameter and length of the designed part, which are not editable. Then, the user enters the dimensions of the available stock, and these should be at least the same as the extracted ones otherwise the system refuses the stock and stops the process. This step is very important since the system does not only validate the entered stock's dimensions for it also uses them as a reference when the tool approaches the stock. For example, the maximum diameter of the part design in case study 1 is 175mm; hence, if the available stock has a diameter of 178mm, the system will consider this value instead of the extracted 175mm when moving the tool near to the stock. This is to avoid any possible damage to the tool and/or the stock.

### **Process selection**

At an early stage in the SI-AFR system, all the faces, and consequently the features, that form the design model are classified based on their location to either externals or internals (4.3.5). Also, each internal shape is defined whether it is located at the axis of rotation or with an offset from it. Such information facilitates the process selection task in the ACAPP system. Table 7.2 shows the machining process for each external and internal feature.

### **Tools selection**

The next step includes the selection of cutting tool(s) for each feature. These are shown in a list comprising details of the tool number, tool offset, insert shape, tool holder type, diameter (if available), and whether it is for external or internal cutting. Table 7.3 and figure 7.8 shows the total number of tools with their details and the features to be machined by it, respectively. As mentioned, the selected tools can produce the recognised features as it is expected; however, this selection is not optimal in terms of the number of tools, tool life, and cost. For example, four end\_mill tools with different diameters (10, 12, 20, and 24 mm) are selected for drilling and milling operations. This can be reduced to only one tool by taking the smallest end\_mill (10mm) and change the machining strategy to produce the larger holes. Although optimising the tools selection is possible, it is not the goal of the research described in this thesis.



Table 7.2 The machining process for each external and internal feature

Feature No.	Feature Name	Machining process
External		
1	Intermediate feature	External turning
2	Intermediate feature	External turning
3	Intermediate feature	External turning
4	Right convex corner	External turning
5	Facing	External turning
6	Radial square groove	External turning
7	Intermediate feature	External turning
8	Radial groove two rounded corners	External turning
9	Two sides tapered groove	External turning
10	Facing	Partin
Through internal shape		
1	Intermediate feature	Drilling
2	Right axial groove	Boring
3	Two sides tapered groove	Internal turning
Front internal shape		
1	Intermediate feature	Drilling
2	Right axial groove	Milling
Back internal shape		
1	Intermediate feature	Drilling
2	Left axial groove	Milling

Table 7.3 The total number of tools required to manufacture the part design with their details

Tool and offset number	Insert shape	Holder type	Diameter	Ext. or Int.	Number of features
0101	C	L	_	External	5
0202	S	F	_	External	4
0303	D	N	_	External	1
0404	End_mill	_	24	Internal	2
0505	C	L	_	Internal	1
0606	D	N	_	Internal	1
0707	End_mill	_	12	Internal	1
0808	End_mill	_	10	Internal	1
0909	End_mill	_	20	Internal	1

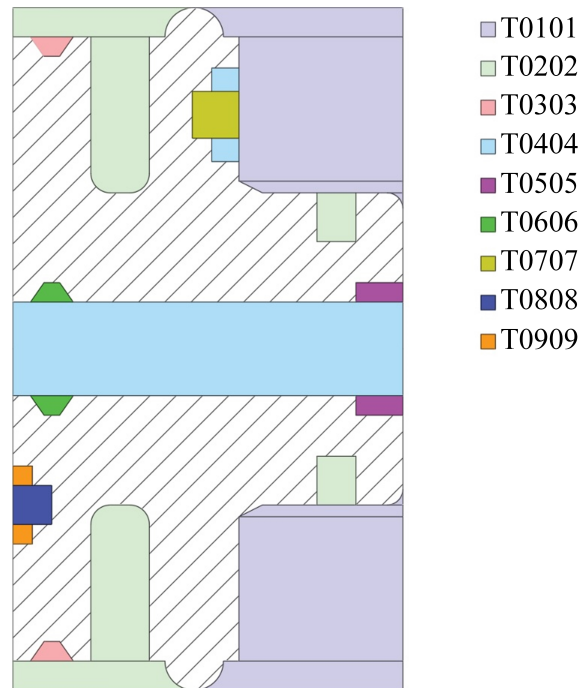


Figure 7.8 Examples of insert shapes that are encoded based on ISO standards

### Sequence of operations

The sequence of operations starts with an external turning for the first nine external features and stops before the last one which is "Facing". This is to check if the model design contains through and/or front internal shapes since they should be machined before parting the workpiece. Then, the ACAPP system takes into consideration machining all the features of the through internal in a sequence of drilling, boring, and internal turning. The features of the front internal shape are machined after, and these require a sequence of drilling and milling operations. Next, a parting process is carried out which produces the last facing feature. The system ends the current G-code, shows a message to flip the workpiece, and generating another G-code to manufacture the back internal shape. This takes the same sequence of operations that was applied to the front internal. Table 7.4 shows the sequence of operations for all the external and internal features. At this point, this sequence can be adjusted to reduce the total time of manufacturing. For example, it is possible to machine the second feature (right axial groove) in the front internal directly after machining the first intermediate feature in the through internal shape. This is because they require the same cutting tool; hence, the extra time of tool changing can be avoided. However, the current sequence is accepted though it is not optimised.

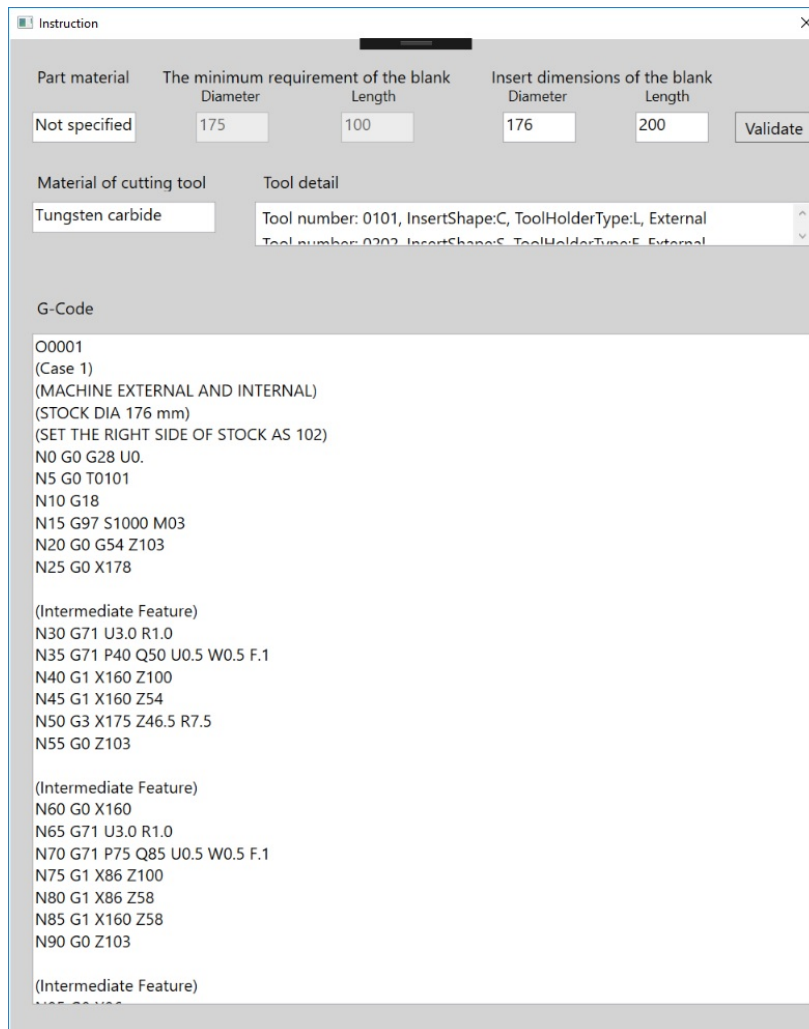
Table 7.4 The sequence of operations for all the external and internal feature

Operation No.	Features	Machining process
1	External/ features (1-9)	External turning
2	Through internal/feature 1	Drilling
3	Through internal/feature 2	Boring
4	Through internal/feature 3	Internal turning
5	Front internal/feature 1	Drilling
6	Front internal/feature 2	Milling
7	External/ feature (10)	Parting
8	Back internal/feature 1	Drilling
9	Front internal/feature 2	Milling

### Generating G code

Whilst the ACAPP system invests the results from the SI-AFR and features subtraction systems to automatically generate the expected process planning outputs, generating the G-code has limitations. This is because the current version of the ACAPP system has been developed to serve the general concept of process planning, and not a specific factory or set of CNC machines. For example, the ACAPP system is programmed to generate the basic cutting movements (G0, G1, G2, and G3) and the canned cycles (G71, G72, and G75), as well as auxiliary setting functions (G28 and G97). These G codes are common and can be used in most if not all the CNC machines. However, they can manufacture only external and internal shapes which are located at the axis of rotation. Whereas the internal shapes that are located within an offset from the axis of rotation require a different set of coding, and this code varies from machine to machine depending on the manufacturer. Thus, only the process and tool selections, as well as sequence of operations, are considered for such internal shapes without generating a G-code for manufacturing them. The output G-code is shown in appendix C.

The G-code is shown on screen via the ACAPP system's window and saved as an individual file at the same directory of the input STEP file. Figure 7.9 shows the final ACAPP system's window.



The screenshot shows a software window titled "Instruction" with a close button (X) in the top right corner. The window is divided into several sections:

- Part material:** A text box containing "Not specified".
- The minimum requirement of the blank:** Two input fields for "Diameter" (175) and "Length" (100).
- Insert dimensions of the blank:** Two input fields for "Diameter" (176) and "Length" (200), followed by a "Validate" button.
- Material of cutting tool:** A text box containing "Tungsten carbide".
- Tool detail:** A dropdown menu showing "Tool number: 0101, InsertShape:C, ToolHolderType:L, External" and "Tool number: 0202, InsertShape:F, ToolHolderType:F, External".
- G-Code:** A large text area containing the following code:

```
O0001
(Case 1)
(MACHINE EXTERNAL AND INTERNAL)
(STOCK DIA 176 mm)
(SET THE RIGHT SIDE OF STOCK AS 102)
N0 G0 G28 U0.
N5 G0 T0101
N10 G18
N15 G97 S1000 M03
N20 G0 G54 Z103
N25 G0 X178

(Intermediate Feature)
N30 G71 U3.0 R1.0
N35 G71 P40 Q50 U0.5 W0.5 F.1
N40 G1 X160 Z100
N45 G1 X160 Z54
N50 G3 X175 Z46.5 R7.5
N55 G0 Z103

(Intermediate Feature)
N60 G0 X160
N65 G71 U3.0 R1.0
N70 G71 P75 Q85 U0.5 W0.5 F.1
N75 G1 X86 Z100
N80 G1 X86 Z58
N85 G1 X160 Z58
N90 G0 Z103

(Intermediate Feature)
N95 G0 X160
```

Figure 7.9 The final ACAPP system window

## 7.8 Methodology evaluation

### 7.8.1 Comparison to CAM software

The development of the ACAPP system based on the SI-AFR and features subtraction concepts has many advantages. These can be clarified and evaluated via a comparison with different CAM software. There is no doubt that CAM software has been providing significant services to the manufacturing industry; however, they have limited capabilities and many constraints. Almost all CAM software follow the same steps in analysing designs and generating G&M code files. This starts by reading the input design, predicting the stock shape and size, and asks the user to confirm the speculation.

Regarding the rotational parts, the default analysis of CAM software for any design shows its profile as a connected sequence of segments created as a single planar object. Such an analysis requires a user that is highly experienced in order to select specific segments; hence, divide the design into machining areas. The model design in figure 7.10a shows a simple example. The part profile of the design is coloured in red, and it covers the part as a continuous segment without any consideration of its features. Since it is very difficult to select a single appropriate tool to machine the whole profile at once, the user should divide the machining area into sub-areas (features) manually, based on the experience. This is by selecting specific end points at the profile segment and connecting them (figure 7.10b). Next, the user analyses each sub-area in order to select an appropriate tool for its shape. For example, an insert shape (C) can be selected to machine the pink area; whereas, the blue area requires an insert shape of (S). After selecting a set of cutting tools, the user needs to provide the CAM software with information that includes the depth of cut, spindle speed, feed rate, and a start point of machining for each area. Whilst a CAM software can detect the machining strategy for the pink area after specifying the start and end points, the machining strategy for the blue area is selected manually. This is because CAM software do not recognise such an area as a square groove without interference. The user selects a machining strategy for each type of groove based on its geometry, the selected tool, the specified start point, and experience level of the individual. Meanwhile, both the process selection and the sequence of operations require the user's interference. For instance, it must be delivered to the CAM software that the pink area is machined before the blue one. This is also the case with the internal shapes when a drilling process should be achieved before the boring.

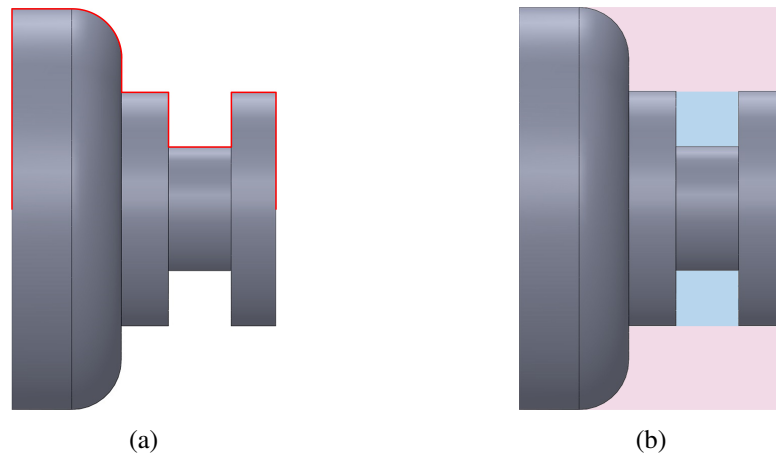


Figure 7.10 Dividing the design into machining areas in CAM software (a) a part profile as a connected sequence of segments in CAM software, and (b) divide the machining area into sub-areas based on the user's experience

After achieving the mentioned tasks manually, the CAM software generates a G-code that machines the part. It is always recommended that the CAM software user should be the same person who uses the resultant G-code on the CNC machine. This is because passing the G-code to the machine operator without identifying its details, such as the symbol of each selected tool, reference plane, etc., is pointless. Otherwise, the user should write a report explaining all the selections in detail and attach it with the G-code and deliver them together to the machine operator.

In contrast with a CAM software, the ACAPP system does not require any experience from the user because all the tasks are achieved automatically. This will also reduce the time required in generating the expected process planning outputs to only one second. This includes the G-code with details about the selected tools and how to set the machine in terms of workpiece zero point. Furthermore, the resultant instruction is clear and sufficient if passed to the machine operator with the G-code.

### 7.8.2 Practical validation of the ACAPP system

In order to practically validate the results from the ACAPP system, an experiment has been carried out at the Brunel University workshop. This is by using Colchester Harrison-Alpha-1350 XS CNC turning machine, which runs on GE FANUC "21i-TB operating system. Since this is the only CNC

turning machine available, the dimensions of the case study had to be scaled down based on the capabilities of the machine. However, all the features from the case study remain.

When attempting to machine this part with the available tools and machine capabilities, there were two minor modifications that were required to the design and machining process. The first modification is related to the radial two sides tapered groove at the through internal shape. It needed to be redesigned and moved closer to the right end of the part so that the available tool holder can manufacture it. The second modification is regarding the sequence of operations. In particular, the first intermediate feature of the through internal shape is constructed first using a different machine, because installing the required length of the end-mill tool to the CNC turning machine causes a collision while using another tool to machine the part.

For the part that is machined, a left section of the model design is shown in figure 7.11. The final external and internal features from the SI-AFR system and the intermediate and independent external and internal features from the features subtraction are shown in tables 7.5, 7.6, 7.7, and 7.8, respectively. The proposed ACAPP system provides high efficiency regarding the required time to accomplish the feature recognition and subtraction tasks, as well as generating the expected outputs. For example, the total time that the system takes to recognise the external and internal features of the model design in figure 7.11 is 5 seconds. This includes the parsing process, achieving all the manipulating algorithms, and recognising all the predefined features. Also, it takes one second for each the feature subtraction and generating the mentioned process planning outputs. Figures 7.12, 7.13, and 7.14 shows the instruction window, the initial blank, and the final part, respectively. Finally, the G-code to manufacture this part can be found in appendix D.

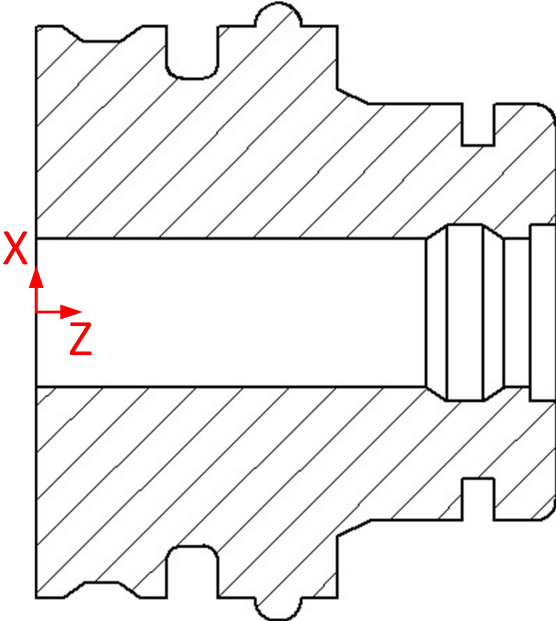


Figure 7.11 The left section of the model design



Table 7.5 The recognized external features of the model design used in the practical validation

External shape													
Feature No.	Feature Name	Face No.	Xsp	Zsp	Xep	Zep	Start-to-end	Xcc	Zcc	CR	Dir.	W	D
1	Facing	1	17	100	36	100	Liner					0	19
2	Right convex corner	1	36	100	40	96	Circler	36	96	4	CCW	4	4
3	Cylindrical	1	40	96	40	88	Liner					8	0
4	Radial square groove	1	40	88	32	88	Liner					6	8
		2	32	88	32	82	Liner						
		3	32	82	40	82	Liner						
5	Cylindrical	1	40	82	40	64	Liner					18	0
6	Right taper	1	40	64	43	58	Liner					6	3
7	Facing	1	43	58	55	58	Liner					0	12
8	Cylindrical	1	55	58	55	51	Liner					7	0
9	Convex	1	55	51	55	42	Circler	55	46.5	4.5	CCW	9	4.5
10	Cylindrical	1	55	42	55	35	Liner					7	0
		1	55	35	48	35	Liner						
		2	48	35	45	32	Circler	48	32	3	CW		
		3	45	32	45	28	Liner					10	10
		4	45	28	48	25	Circler	48	28	3	CW		
		5	48	25	55	25	Liner						
12	Cylindrical	1	55	25	55	20	Liner					5	0
		1	55	20	52	16	Liner						
		2	52	16	52	9	Liner					15	3
		3	52	9	55	5	Liner						
14	Cylindrical	1	55	5	55	0	Liner					5	0
15	Facing	1	55	0	14.25	0	Liner					0	40.75

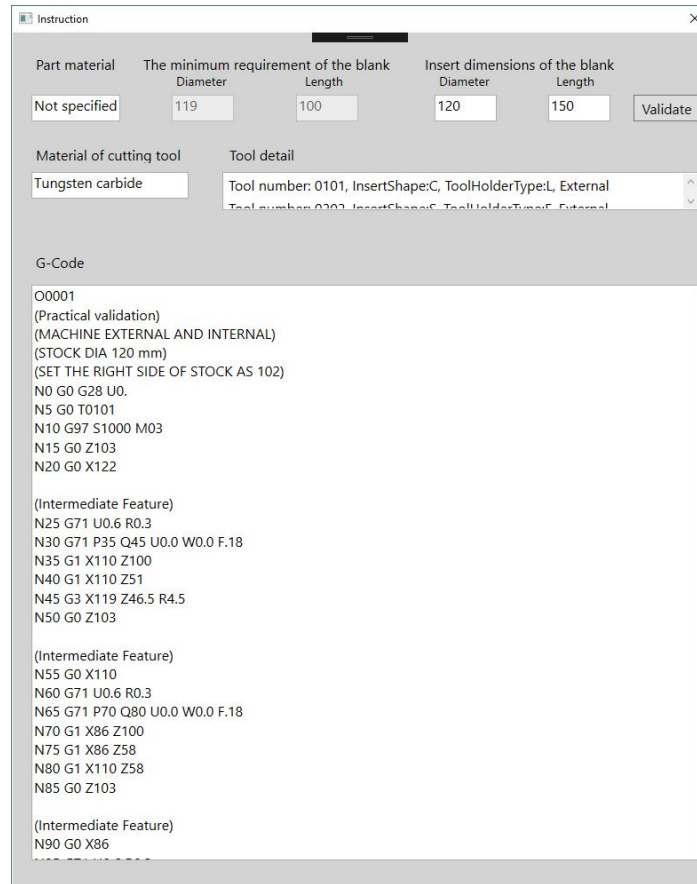


Table 7.7 External intermediate and independent features of the model design used in the practical validation

		External shape												
Feature No.	Feature Name	Face No.	Xsp	Zsp	Xep	Zep	Start-to-end	Xcc	Zcc	CR	Dir.			
1	Intermediate feature	1	59.5	100	55	100	Linear							
		2	55	100	55	51	Linear							
		3	55	51	59.5	46.5	Circler	55	46.5	4.5	CCW			
2	Intermediate feature	1	55	100	43	100	Linear							
		2	43	100	43	58	Linear							
		3	43	58	55	58	Linear							
3	Intermediate feature	1	43	100	40	100	Linear							
		2	40	100	40	64	Linear							
		3	40	64	43	58	Linear							
4	Right convex corner	1	36	100	40	96	Circler	36	96	4	CCW			
		1	17	100	36	100	Linear							
6	Radial square groove	1	40	88	32	88	Linear							
		2	32	88	32	82	Linear							
		3	32	82	40	82	Linear							
7	Intermediate feature	1	59.5	46.5	55	42	Circler	55	46.5	4.5	CCW			
		2	55	42	55	0	Linear							
		3	55	0	59.5	0	Linear							
8	Radial groove two rounded corners	1	55	35	48	35	Linear							
		2	48	35	45	32	Circler	48	32	3	CW			
		3	45	32	45	28	Linear							
		4	45	28	48	25	Circler	48	28	3	CW			
		5	48	25	55	25	Linear							
9	Two sides tapered groove	1	55	20	52	16	Linear							
		2	52	16	52	9	Linear							
		3	52	9	55	5	Linear							
10	Facing	1	55	0	14.25	0	Linear							

Table 7.8 Internal intermediate and independent features of the model design used in the practical validation

Feature No.	Feature Name	Internal shapes										
		Face No.	Xc	Yc	Zcs	RZcs	Zce	RZce	Start-to-end	Xcc	CR	Dir.
1	Intermediate feature	1	0	0	100	0	100	14.5	Linear			
		2	0	0	100	14.25	0	14.25	Linear			
		3	0	0	0	14.25	0	0	Linear			
1	Right axial groove	1	0	0	100	17	95	17	Linear			
		2	0	0	95	17	95	14.25	Linear			
3	Two sides tapered groove	1	0	0	90	14.25	86	17	Linear			
		2	0	0	86	17	79	17	Linear			
		3	0	0	79	17	75	14.25	Linear			



The screenshot shows a software window titled "Instruction" with a close button (X) in the top right corner. The window is divided into several sections:

- Part material:** A text box containing "Not specified".
- The minimum requirement of the blank:** A table with two columns: "Diameter" and "Length". The "Diameter" column has a value of "119" and the "Length" column has a value of "100".
- Insert dimensions of the blank:** A table with two columns: "Diameter" and "Length". The "Diameter" column has a value of "120" and the "Length" column has a value of "150".
- Material of cutting tool:** A text box containing "Tungsten carbide".
- Tool detail:** A dropdown menu showing "Tool number: 0101, InsertShape:C, ToolHolderType:L, External".
- G-Code:** A large text area containing the following code:

```
O0001
(PRACTICAL VALIDATION)
(MACHINE EXTERNAL AND INTERNAL)
(STOCK DIA 120 mm)
(SET THE RIGHT SIDE OF STOCK AS 102)
N0 G0 G28 U0.
N5 G0 T0101
N10 G97 S1000 M03
N15 G0 Z103
N20 G0 X122

(Intermediate Feature)
N25 G71 U0.6 R0.3
N30 G71 P35 Q45 U0.0 W0.0 F.18
N35 G1 X110 Z100
N40 G1 X110 Z51
N45 G3 X119 Z46.5 R4.5
N50 G0 Z103

(Intermediate Feature)
N55 G0 X110
N60 G71 U0.6 R0.3
N65 G71 P70 Q80 U0.0 W0.0 F.18
N70 G1 X86 Z100
N75 G1 X86 Z58
N80 G1 X110 Z58
N85 G0 Z103

(Intermediate Feature)
N90 G0 X86
```

Figure 7.12 The instruction window generated of the model design used in the practical validation

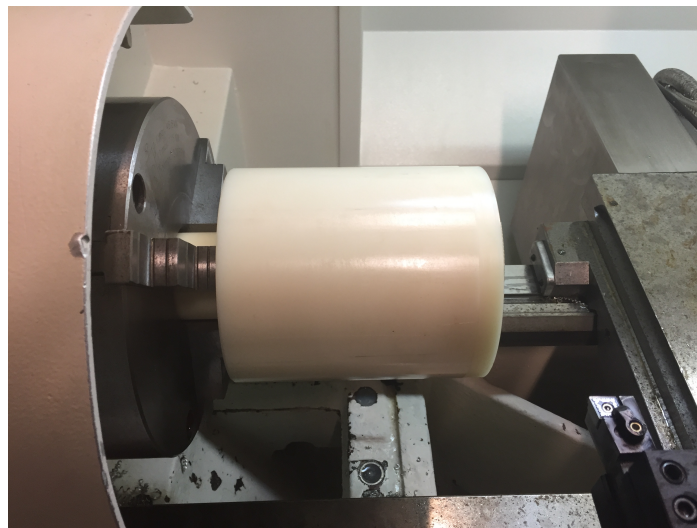


Figure 7.13 The initial blank of the model design used in the practical validation

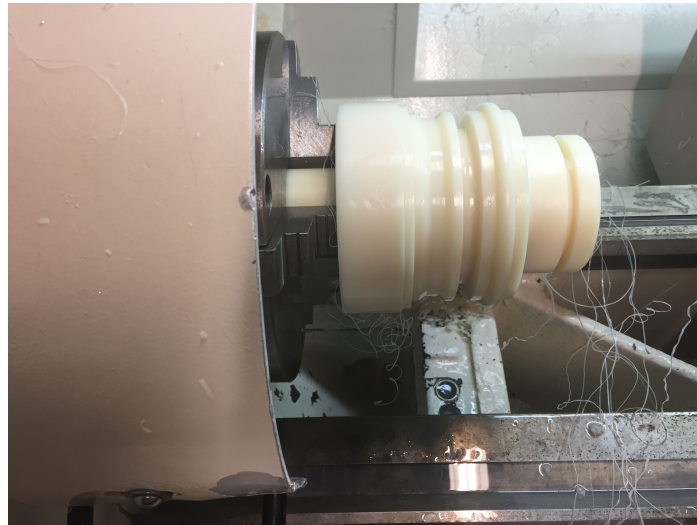


Figure 7.14 The final machined part of the model design used in the practical validation

## 7.9 Discussion and summary

The ACAPP system has been developed to show the important role of the SI-AFR and features subtraction systems in the automatic generation of process planning outputs. This is by analysing the results from both systems, which include geometrical, topological, and location information for each final and intermediate feature. In contrast with most CAM software that require human intervention in achieving process planning tasks, such a method terminates this necessity because the tasks are accomplished automatically. The system has been practically evaluated, and the result shows that the manufactured part is produced within expected dimensions. Also, the process of generating CAPP outputs takes a few seconds and without any need for user expertise. However, the current version of the system has been tested only for general purpose, and hence a small adjustment would be required to the system for special cases. For example, holes that are not located in the axis of rotation necessitate a specific set of G-code to be manufactured. This depends on the machine's operating system and capabilities.

Designer, process planner, and manufacturer would benefit from this smart ACAPP system. The designer can consider the manufacturing implications of a part while in the design process. The process planner's task is simplified significantly by the smart ACAPP system, which automatically performs the steps that in traditional manufacturing would be carried out manually. Regarding

the manufacturer, the system provides information about setting up the machine and cutting tools, selecting critical datum of the workpiece, and step-by-step instructions of the manufacturing process.

This chapter covers the automatic generation of some process planning outputs, such as the process selection, cutting tools, sequence of operations, and generating G-code. However, these are not optimised in terms of reducing the time and production cost. Whilst the optimisation of these tasks is valuable, it is out the scope of the current work. This objective can be achieved in the future work besides the uncovered process planning outputs like the selecting of jigs and fixtures and estimating the time and cost to manufacture the part.

## **Chapter 8**

# **Recommendations for future work**

The feature technology, in its two main parts of feature recognition and subtraction, provides sufficient geometrical and topological information about design models. Whilst this research has shown the significant role of using feature technology to automatically generate CAPP outputs, future work is necessary to optimise the results in terms of reducing the time and production cost further and accomplish the uncovered tasks of CAPP. However, CAPP is multidisciplinary, and the feature technology is only one aspect of its inputs. Hence, other inputs of CAPP are also possible directions for future research. This is to take current CAPP systems to a higher level of automation. The vision of future work is outlined in the sections below.

### **8.1 Covering CAPP inputs**

Although the feature technology is considered one of the most important inputs in creating an ACAPP system, other inputs are not trivial and should be considered as well. For example, dimensions and tolerances of a design can affect the selection or the sequence of machining operations. This statement is also true regarding the surface roughness because the machining operations are different in their abilities in terms of achieving the required tolerance and/or surface finish value. Whilst different STEP files, such as AP214 and AP242, are prepared to transfer such information of dimensions, tolerances, and surface finish, the inner translators of most CAD vendors do not support this property. Hence, this information is lost or delivered in an unsatisfactory way while saving the design as a STEP file.



This requires more investigations in an attempt to develop a tool that extracts such information from designs and attaches it to the recognised features.

Another input is process and machine capabilities that is considered based on global standards for this work. The ACAPP system can be more precise and realistic if it is customised according to the available process and machine capabilities of a specific factory. This is to generate a feasible process plan that guarantees its result is compatible with the design intention. This can also include the model of the machine and its set of the G-code, since this varies between different machines. For example, a canned cycle for rough turning might be defined using a specific G number and block in a CNC machine, whereas another machine has a different definition for the same canned cycle.

## **8.2 Optimising and considering the uncovered CAPP outputs**

Although the ACAPP system automatically generates some process planning outputs, these are not optimised. However, the optimisation of these tasks is an objective in future work. For example, the tool selection tasks is currently achieved for each manufacturing feature individually. Hence, if the part includes two grooves, and these have different angle values on their sides, the ACAPP system will select two different cutting tools for machining them. Consequently, this selection results in extra cost and time for tools setting. Future work includes checking if one of the selected tools is able to manufacture both grooves, which reduces the time and production cost. The sequence of machining features is another example of optimization since it is currently achieved in an order based on the locations of the features. This might result in extra time for tools travelling and changing. Such an issue can be solved by checking which features are prepared to be manufactured using the same tool before it is changed.

Gripping a workpiece in a safe, fast, accurate and rigid way on a lathe is a critical requirement for successful turning. The geometry of a part is one variable that can affect the selection of the holding method and clamping zone in turning processes. Hence, the feature technology can also be used in selecting jigs and fixtures.

In order to estimate the production cost of a part, this depends in the first place on the workpiece material and the tools. The ACAPP system is provided with the stock dimensions and selects the

cutting tools based on the features technology. Future work includes linking the ACAPP system with a database of materials and tools expenses for the purpose of calculating the workpiece and tools cost.

### **8.3 Extending functionality to prismatic parts**

Process planning and STEP files are general broad tools in manufacturing industry, and they are not simply limited to rotational parts, therefore the functionality of the current ACAPP system can be extended to include prismatic parts. This is a non-trivial problem, as it requires depth of understanding of prismatic parts and the implementation is more complex than for rotational parts.

### **8.4 How AI will effect manufacturing in future**

AI techniques is capable of completely revolutionising the manufacturing process as it would significantly reduce human intervention, and the need for human expertise. Consequently, this would avoid any mistakes in generating instructions for manufacturing parts, and the human variability in the decision-making process.

# References

- [1] X. Xu, L. Wang, and S. T. Newman, "Computer-aided process planning – A critical review of recent developments and future trends," *International Journal of Computer Integrated Manufacturing*, vol. 24, no. 1, pp. 1–31, 2011.
- [2] F. Cay and C. Chassapis, "An IT view on perspectives of computer aided process planning research," *Computers in Industry*, vol. 34, no. 3, pp. 307–337, 1997.
- [3] H. B. Marri, A. Gunasekaran, and R. J. Grieve, "Computer-Aided Process Planning: A State of Art," pp. 261–268, 1998.
- [4] M. Kumar and S. Rajotia, "Integration of scheduling with computer aided process planning," *Journal of Materials Processing Technology*, vol. 138, no. 1-3, pp. 297–300, 2003.
- [5] J. F. Bard and T. A. Feo, "The cutting path and tool selection problem in computer aided process planning," *Journal of Manufacturing Systems*, vol. 8, no. 1, pp. 17–26, 1989.
- [6] Y. Yusof and K. Latif, "Survey on computer-aided process planning," *International Journal of Advanced Manufacturing Technology*, vol. 75, no. 1-4, pp. 77–89, 2014.
- [7] G. Halevi and R. D. Weill, "Computer-aided process planning (CAPP)," in *Principles of Process Planning*. Dordrecht: Springer Netherlands, 1995, pp. 317–332.
- [8] J. Ciurana, I. Ferrer, and J. X. Gao, "Activity model and computer aided system for defining sheet metal process planning," *Journal of Materials Processing Technology*, vol. 173, no. 2, pp. 213–222, 2006.
- [9] M. M. M. Sarcar, K. M. Rao, and K. L. Narayan, *Computer aided design and manufacturing*. PHI Learning Pvt. Ltd., 2008.
- [10] A. Verroust, F. Schonek, and D. Roller, "Rule-oriented method for parameterized computer-aided design," *Computer-Aided Design*, vol. 24, no. 10, pp. 531–540, 1992.
- [11] A. Jezernik and G. Hren, "A solution to integrate computer-aided design (CAD) and virtual reality (VR) databases in design and manufacturing processes," *International Journal of Advanced Manufacturing Technology*, vol. 22, no. 11-12, pp. 768–774, 2003.
- [12] K. Crow, "computer-aided process planning; DRM Associates," 1992.
- [13] Y. Qin, W. Lu, Q. Qi, X. Liu, Y. Zhong, P. J. Scott, and X. Jiang, "Status, Comparison, and Issues of Computer-Aided Design Model Data Exchange Methods Based on Standardized Neutral Files and Web Ontology Language File," *Journal of Computing and Information Science in Engineering*, vol. 17, no. 1, p. 010801, 2016.
- [14] I. Horváth and R. W. Vroom, "Ubiquitous computer aided design: A broken promise or a Sleeping Beauty?" *CAD Computer Aided Design*, vol. 59, pp. 161–175, 2015. [Online]. Available: <http://dx.doi.org/10.1016/j.cad.2014.10.006>

- [15] X. V. Wang and X. W. Xu, "A collaborative product data exchange environment based on STEP," *International Journal of Computer Integrated Manufacturing*, vol. 28, no. 1, pp. 75–86, 2015.
- [16] S. R. Seeram, M. A. Ali, and S. Karimulla, "Automatic recognition of internal features of axisymmetric parts from 2-D images," *Indian Journal of Science and Technology*, vol. 9, no. 44, 2016.
- [17] S. PIRZADA, "COMPUTER-AIDED PROCESS PLANNING (CAPP)," Ph.D. dissertation, New Jersey Institute of Technology, 1991.
- [18] W. D. Engelke, *How to Integrate CAD/ CAM Systems Management and Technology*, Illustrated, Ed. CRC Press, 1987.
- [19] M. Lundgren, M. Hedlind, and T. Kjellberg, "PROCESS PLANNING METHODOLOGY GUIDE FOR INTERACTIVE LEARNING Magnus," in *The sixth Swedish Production Symposium*, 2014.
- [20] I. Ham and S. C.-Y. Lu, "Computer-aided process planning: The Present and the Future (CAPP)," *CIRP Annals - Manufacturing Technology*, vol. 37, no. 2, pp. 591–601, 1988.
- [21] G. P. Sutton, "Survey of Process Planning Practices and Needs," *Maznufacturing Systems*, vol. 8, no. 1, pp. 69–71, 1988.
- [22] V. Krishna, N. V. S. Shankar, and S. Babu, "Feature Based Modeling and Automated Process Plan Generation for Turning Components," *Advances in Production Engineering & Management*, vol. 6, no. 3, pp. 153–162, 2011.
- [23] M. Al-Wswasi, A. Ivanov, and H. Makatsoris, "A survey on smart automated computer-aided process planning (ACAPP) techniques," *The International Journal of Advanced Manufacturing Technology*, 2018. [Online]. Available: <https://doi.org/10.1007/s00170-018-1966-1>
- [24] D. E. Culler and W. Burd, "A framework for extending computer aided process planning to include business activities and computer aided design and manufacturing (CAD/CAM) data retrieval," *Robotics and Computer-Integrated Manufacturing*, vol. 23, no. 3, pp. 339–350, 2007.
- [25] M. Kanumury and T. Chang, "Process planning in an automated manufacturing environment," *Journal of Manufacturing Systems*, vol. 10, no. 1, pp. 67–78, 1991.
- [26] J. Y. Park and B. Khoshnevis, "A real-time computer-aided process planning system as a support tool for economic product design," *Journal of Manufacturing Systems*, vol. 12, no. 2, pp. 181–193, 1993.
- [27] M. Al-wswasi and A. Ivanov, "A novel and smart interactive feature recognition system for rotational parts using a STEP file," *The International Journal of Advanced Manufacturing Technology*, 2019. [Online]. Available: <https://doi.org/10.1007/s00170-019-03849-1>
- [28] M. Kang, J. Han, and J. G. Moon, "An approach for interlinking design and process planning," *Journal of Materials Processing Technology*, vol. 139, no. 1-3 SPEC, pp. 589–595, 2003.
- [29] X. Zhou, Y. Qiu, G. Hua, H. Wang, and X. Ruan, "A feasible approach to the integration of CAD and CAPP," *CAD Computer Aided Design*, vol. 39, no. 4, pp. 324–338, 2007.
- [30] E. S. Abouel Nasr and A. K. Kamrani, "A new methodology for extracting manufacturing features from CAD system," *Computers and Industrial Engineering*, vol. 51, no. 3, pp. 389–415, 2006.

- [31] A. Ball, "Preserving Computer-Aided Design (CAD)," *Technology Watch Reports*, no. April, 2013.
- [32] O. W. Salomons, F. J. A. M. van Houten, and H. J. J. Kals, "Review of research in feature-based design," *Journal of Manufacturing Systems*, vol. 12, no. 2, pp. 113–132, 1993.
- [33] H.-c. Chang, W. F. Lu, and F. X. Liu, "Applied Artificial Intelligence : An International Journal Machining process planning of prismatic parts using case- based reasoning and past process knowledge," vol. 9514, no. October 2013, pp. 37–41, 2010.
- [34] E. M. Sanfilippo and S. Borgo, "What are features? An ontology-based review of the literature," *CAD Computer Aided Design*, vol. 80, pp. 9–18, 2016.
- [35] L. Ding and Y. Yue, "Novel ANN-based feature recognition incorporating design by features," *Computers in Industry*, vol. 55, no. 2, pp. 197–222, 2004.
- [36] S. P. Turley, D. M. Diederich, B. K. Jayanthi, A. Datar, C. B. Ligetti, and D. A. Finke, "Automated Process Planning and CNC-Code Generation," *2014 Industrial and Systems Engineering Research Conference*, vol. 2138, no. November 2015, pp. 2138–2145, 2014.
- [37] J. C. T. Benavente, J. C. E. Ferreira, C. M. Goulart, and V. G. de Oliveira, "A STEP-NC compliant system for the remote design and manufacture of mechanical components through the Internet," *International Journal of Computer Integrated Manufacturing*, vol. 26, no. 5, pp. 412–428, 2013.
- [38] Y. Zhang, X. Luo, B. Zhang, and S. Zhang, "Semantic approach to the automatic recognition of machining features," *The International Journal of Advanced Manufacturing Technology*, pp. 1–21, 2016.
- [39] N. N. Z. Gindy, Y. Yue, and C.-F. Zhu, "Automated feature validation for creating/editing feature-based component data models," *International Journal of Production Research*, vol. 36, no. 9, pp. 2479–2495, 1998.
- [40] X. Zhang, A. Nassehi, and S. T. Newman, "Feature recognition from CNC part programs for milling operations," *International Journal of Advanced Manufacturing Technology*, vol. 70, no. 1-4, pp. 397–412, 2014.
- [41] B. Emmanuel, "Intelligent Techniques for Automatic Feature Recognition in CAD Models," Ph.D. dissertation, University of Wales, Cardiff, 2004.
- [42] B. R. Babic, N. Nešić, and Z. Miljkovic, "Automatic feature recognition using artificial neural networks to integrate design and manufacturing: Review of automatic feature recognition systems," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 25, no. 03, pp. 289–304, 2011.
- [43] M. Flasiński and J. Jurek, "Fundamental methodological issues of syntactic pattern recognition," *Pattern Analysis and Applications*, vol. 17, no. 3, pp. 465–480, 2014.
- [44] K. S. Fu, "Introduction to Syntactic Pattern Recognition," in *Syntactic Pattern Recognition, Applications*, K. S. Fu, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1977, pp. 1–30.
- [45] D. Yip-Hoi, D. Dutta, and Z. Huang, "A customizable machining feature extraction methodology for turned components," *Journal of Manufacturing Systems*, vol. 22, no. 2, pp. 82–98, 2003.

- [46] H. S. Ketan and Z. H. Yaqoub, "BUILT AUTOMATIC FEATURE RECOGNITION SYSTEM BASED ON SWEEPING PRIMITIVE RULE," in *The Seventh Jordanian International Mechanical Engineering Conference*, 2010, p. 15.
- [47] D. B. Perng, Z. Chen, and R. K. Li, "Automatic 3D machining feature extraction from 3D CSG solid input," *Computer-Aided Design*, vol. 22, no. 5, pp. 285–295, 1990.
- [48] S. Meeran and M. J. Pratt, "Automated feature recognition from 2D drawings," *Computer-Aided Design*, vol. 25, no. 1, pp. 7–17, 1993.
- [49] A. Arivazhagan, N. K. Mehta, and P. K. Jain, "Development of a feature recognition module for tapered and curved base features," *International Journal of Advanced Manufacturing Technology*, vol. 39, no. 3-4, pp. 319–332, 2008.
- [50] A. K. Verma and S. Rajotia, "A review of machining feature recognition methodologies," *International Journal of Computer Integrated Manufacturing*, vol. 23, no. 4, pp. 353–368, 2010.
- [51] S. Joshi and T. C. Chang, "Graph-based heuristics for recognition of machined features from a 3D solid model," *Computer-Aided Design*, vol. 20, no. 2, pp. 58–66, 1988.
- [52] M. Marefat and R. L. Kashyap, "Geometric Reasoning for Recognition of Three-Dimensional Object Features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 949–965, 1990.
- [53] J. Han, M. Pratt, and W. C. Regli, "Manufacturing feature recognition from solid models: A Status Report," vol. 16, no. 6, pp. 1–31, 2000.
- [54] J. ZHU, M. KATO, T. TANAKA, H. YOSHIOKA, and Y. SAITO, "Graph based automatic process planning system for multi-tasking machine," *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, vol. 9, no. 3, pp. 1–15, 2015.
- [55] M. Wooldridge, *An introduction to multiagent systems*. John Wiley & Sons, 2009.
- [56] A.-J. Fougères and E. Ostrosi, "Intelligent agents for feature modelling in computer aided design," *Journal of Computational Design and Engineering*, vol. 5, no. 1, pp. 19–40, 2018. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S2288430017300520>
- [57] J. H. Vandenbrande and A. A. G. Requicha, "Spatial Reasoning for the Automatic Recognition of Machinable Features in Solid Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 12, pp. 1269–1285, 1993.
- [58] J. Han and A. A. Requicha, "Integration of feature based design and feature recognition," *Computer-Aided Design*, vol. 29, no. 5, pp. 393–403, 1997.
- [59] W. C. Regli, "Title of Dissertation: Geometric Algorithms for Recognition of Features from Solid Models," *III, Doctor of Philosophy*, 1995.
- [60] M. R. Henderson and D. C. Anderson, "Computer recognition and extraction of form features: A CAD/CAM link," *Computers in Industry*, vol. 5, no. 4, pp. 329–339, 1984.
- [61] S. Sivakumar and V. Dhanalakshmi, "A feature-based system for CAD/CAM integration through STEP file for cylindrical parts," *Indian Journal of Engineering and Materials Sciences*, vol. 20, no. 1, pp. 21–26, 2013.

- [62] J. Oussama, E. Abdelilah, and R. Ahmed, "Manufacturing Computer Aided Process Planning For Rotational Parts . Part 1 : Automatic Feature Recognition From STEP AP203," *Int. Journal of Engineering Research and Applications*, vol. 4, no. 5, pp. 14–25, 2014.
- [63] M. R. Hendersona, G. Srinatha, R. Stagea, K. Walkera, and R. William, "Boundary Representation-based Feature Identification," in *Advances in Feature Based Manufacturing*, 1994, ch. chapter on.
- [64] S. Prabhakar and M. R. Henderson, "Automatic form-feature recognition using neural-network-based techniques on boundary representations of solid models," *Computer-Aided Design*, vol. 24, no. 7, pp. 381–393, 1992.
- [65] V. B. Sunil and S. S. Pande, "Automatic recognition of machining features using artificial neural networks," *The International Journal of Advanced Manufacturing Technology*, vol. 41, no. 9-10, pp. 932–947, 2009. [Online]. Available: <http://link.springer.com/10.1007/s00170-008-1536-z>
- [66] V. Rameshbabu and M. S. Shunmugam, "Hybrid feature recognition method for setup planning from STEP AP-203," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 2, pp. 393–408, 2009.
- [67] W. D. Li, S. K. Ong, and A. Y. Nee, "A hybrid method for recognizing interacting machining features," *International Journal of Production Research*, vol. 41, no. 9, pp. 1887–1908, 2003.
- [68] Y. J. Tseng and S. B. Joshi, "Recognition of interacting rotational and prismatic machining features from 3-D mill-turn parts," *International Journal of Production Research*, vol. 36, no. 11, pp. 3147–3165, 1998.
- [69] L. Liu, Z. Huang, W. Liu, and W. Wu, "Extracting the turning volume and features for a mill/turn part with multiple extreme faces," *International Journal of Advanced Manufacturing Technology*, vol. 94, no. 1-4, pp. 257–280, 2018.
- [70] Y. Shi, Y. Zhang, S. Baek, W. De Backer, and R. Harik, "Manufacturability analysis for additive manufacturing using a novel feature recognition technique," *Computer-Aided Design and Applications*, vol. 15, no. 6, pp. 941–952, 2018. [Online]. Available: <https://doi.org/10.1080/16864360.2018.1462574>
- [71] D. Gossard, R. Zuffante, and H. Sakurai, "Representing dimensions, tolerances, and features in MCAE systems," *IEEE Computer Graphics and Applications*, vol. 8, no. 2, pp. 51–59, 1988.
- [72] U. Roy, C. R. Liu, and T. C. Woo, "Review of dimensioning and tolerancing: representation and processing," *Computer-Aided Design*, vol. 23, no. 7, pp. 466–483, 1991.
- [73] T. Kandikjan, J. J. Shah, and J. K. Davidson, "A mechanism for validating dimensioning and tolerancing schemes in CAD systems," *CAD Computer Aided Design*, vol. 33, no. 10, pp. 721–737, 2001.
- [74] D. Fainguelernt, R. Weill, and P. Bourdet, "Computer Aided Tolerancing and Dimensioning in Process Planning," *CIRP Annals - Manufacturing Technology*, vol. 35, no. 1, pp. 381–386, 1986.
- [75] P. J. Drake, *Dimensioning and Tolerancing Handbook*, 1999. [Online]. Available: <http://books.google.com.co/books?id=XOxqQgAACAAJ>
- [76] Y. Qin, Q. Qi, W. Lu, X. Liu, P. J. Scott, and X. Jiang, "A review of representation models of tolerance information," *International Journal of Advanced Manufacturing Technology*, vol. 95, no. 5-8, pp. 2193–2206, 2018.

- [77] P. Hoffmann, "Analysis of tolerances and process inaccuracies in discrete part manufacturing," *Computer-Aided Design*, vol. 14, no. 2, pp. 83–88, 1982.
- [78] G. A. Britton and G. Thimm, "A Matrix Method for Calculating Working Dimensions and Offsets for Tolerance Charting," *Advanced Manufacturing Technology*, vol. 20, no. 6, pp. 448–453, 2002.
- [79] Z. Shen, J. Shah, and J. Davidson, "Analysis neutral data structure for GD&T," *Journal of Intelligent Manufacturing*, vol. 19, no. 4, pp. 455–472, 2008. [Online]. Available: <http://www.springerlink.com/index/P46P81253W148451.pdf>
- [80] A. Venkiteswaran, "Interoperability of Geometric Dimension & Tolerance Data between CAD Systems," Ph.D. dissertation, ARIZONA STATE UNIVERSITY, 2016.
- [81] N. S. Ermolaeva, K. G. Kaveline, and J. L. Spoomaker, "Materials selection combined with optimal structural design: concept and some results," *Materials & Design*, vol. 23, no. 5, pp. 459–470, 2002.
- [82] D. E. Whitney, "Manufacturing by Design," *Harvard business review*, no. July, pp. 83–92, 1988.
- [83] X. F. Zha, "A web-based advisory system for process and material selection in concurrent product design for a manufacturing environment," *International Journal of Advanced Manufacturing Technology*, vol. 25, no. 3-4, pp. 233–243, 2005.
- [84] R. Giachetti, "A decision support system for material and manufacturing process selection," *Journal of Intelligent Manufacturing*, vol. 9, pp. 265–276, 1998.
- [85] B. Arezoo, K. Ridgway, and A. M. A. Al-Ahmari, "Selection of cutting tools and conditions of machining operations using an expert system," *Computers in Industry*, vol. 42, no. 1, pp. 43–58, 2000.
- [86] R. M. Boogert, H. J. J. Kals, and F. J. A. M. van Houten, "Tool Paths and Cutting Technology in Computer-Aided Process Planning," *The International Journal of Advanced Manufacturing Technology*, vol. 11, no. 3, pp. 186–197, 1996.
- [87] O. Al-shebeeb and B. Gopalakrishnan, "Influence of Materials Properties on Process Planning Effectiveness," p. 9, 2017.
- [88] A. Requicha and S. Chan, "Representation of geometric features, tolerances, and attributes in solid modelers based on constructive geometry," *IEEE Journal on Robotics and Automation*, vol. 2, no. 3, pp. 156–166, 1986.
- [89] H.-P. (BEN) WANG and R. A. WYSK, "A knowledge-based approach for automated process planning," *International Journal of Production Research*, vol. 26, no. 6, pp. 999–1014, 1988.
- [90] P. V. S. Suresh, P. V. Rao, and S. G. Deshmukh, "A genetic algorithmic approach for optimization of surface roughness prediction model," *International Journal of Machine Tools & Manufacture*, vol. 42, no. January, pp. 675–680, 2002.
- [91] A. Bhattacharyya, R. Faría-González, and I. Ham, "Regression Analysis for Predicting Surface Finish and Its Application in the Determination of Optimum Machining Conditions," *Journal of Engineering for Industry*, vol. 92, no. 3, pp. 711–714, 1970.
- [92] A. Bhattacharya, S. Das, P. Majumder, and A. Batish, "Estimating the effect of cutting parameters on surface finish and power consumption during high speed machining of AISI 1045 steel using Taguchi design and ANOVA," *Production Engineering*, vol. 3, no. 1, pp. 31–40, 2009.



- [93] K. Cheng, *Machining dynamics: fundamentals, applications and practices*. Springer Science & Business Media, 2008.
- [94] T. Özel, Y. Karpat, L. Figueira, and J. P. Davim, “Modelling of surface finish and tool flank wear in turning of AISI D2 steel with ceramic wiper inserts,” *Journal of Materials Processing Technology*, vol. 189, no. 1-3, pp. 192–198, 2007.
- [95] Hsu-Pin Ben Wang and R. A. Wysk, “Intelligent reasoning for process planning,” *Computers in Industry*, vol. 8, no. 4, pp. 293–309, 1987.
- [96] T.-C. Chang and R. a. Wysk, “Integrating CAD and CAM through automated process planning,” *International Journal of Production Research*, vol. 22, no. 5, pp. 877–894, 1984.
- [97] H.-C. Chang, L. Dong, F. Liu, and W. Lu, “Indexing and retrieval in machining process planning using case-based reasoning,” *Artificial Intelligence in Engineering*, vol. 14, pp. 1–13, 2000.
- [98] W. Grzesik and T. Wanat, “Surface finish generated in hard turning of quenched alloy steel parts using conventional and wiper ceramic inserts,” *International Journal of Machine Tools and Manufacture*, vol. 46, no. 15, pp. 1988–1995, 2006.
- [99] K. S. Chen, M. L. Huang, and R. K. Li, “Process capability analysis for an entire product,” *International Journal of Production Research*, vol. 39, no. 17, p. 4077\_4087, 2001.
- [100] S. C. Park, “Knowledge capturing methodology in process planning,” *Computer-Aided Design*, vol. 35, pp. 1109–1117, 2003.
- [101] D. Šormaz, M. Wakh, and N.-u. Arafat, “Rule-Based Process Planning and Setup Planning With Considerations of Gd & T Requirements,” *International Journal "Advanced Quality"*, vol. 45, no. 1, pp. 13–20, 2017.
- [102] Vinod V Rampur and Dr. Sachhidanand Reur, “Computer Aided Process Planning using STEP Neutral File for Automotive Parts,” *International Journal of Engineering Research and*, vol. V6, no. 04, 2017.
- [103] P. Kostal and A. Mudrikova, “Laboratory of Flexible Manufacturing System,” *Advanced Materials Research*, vol. 429, no. January, pp. 31–36, 2012.
- [104] N. K. Jain and V. K. Jain, “Process Selection Methodology for Advanced Machining Processes,” *Journal of Advanced Manufacturing Systems*, vol. 02, no. 01, pp. 5–45, 2003.
- [105] H. C. W. Lau, C. K. M. Lee, B. Jiang, I. K. Hui, and K. F. Pun, “Development of a computer-integrated system to support CAD to CAPP,” *International Journal of Advanced Manufacturing Technology*, vol. 26, no. 9-10, pp. 1032–1042, 2005.
- [106] H. C. Zhang and M. E. Merchant, “IPPM - A Prototype to Integrate Process Planning and Job Shop Scheduling Functions,” *CIRP Annals - Manufacturing Technology*, vol. 42, no. 1, pp. 513–518, 1993.
- [107] P. Gu and Y. Zhang, “Operation sequencing in an automated process planning system,” *Journal of Intelligent Manufacturing*, vol. 4, no. 3, pp. 219–232, 1993.
- [108] J. P. Kruth, J. Detand, G. Van Zeir, J. Kempenaers, and J. Pinte, “Methods to improve the response time of a CAPP system that generates non-linear process plans,” *Advances in Engineering Software*, vol. 25, no. 1, pp. 9–17, 1996.
- [109] J. P. Kruth and J. Detand, “A CAPP System for Nonlinear Process Plans,” *CIRP Annals - Manufacturing Technology*, vol. 41, no. 1, pp. 489–492, 1992.

- [110] V. Pandey, M. K. Tiwari, and S. Kumar, "An interactive approach to solve the operation sequencing problem using simulated annealing," *International Journal of Advanced Manufacturing Technology*, vol. 29, no. 11-12, pp. 1212–1231, 2006.
- [111] K. Ridgway, "Selection of cutting tools and conditions of machining operations using an expert system Selection of cutting tools and conditions of machining operations," no. January 2016, 2000.
- [112] A. Oral and M. C. Cakir, "Automated cutting tool selection and cutting tool sequence optimisation for rotational parts," *Robotics and Computer-Integrated Manufacturing*, vol. 20, no. 2, pp. 127–141, 2004.
- [113] K. Fernandes and V. Raja, "Incorporated tool selection system using object technology," *International Journal of Machine Tools and Manufacture*, vol. 40, no. 11, pp. 1547–1555, 2000.
- [114] M. V. Ribeiro and N. L. Coppini, "Applied database system for the optimization of cutting conditions and tool selection," *Journal of Materials Processing Technology*, vol. 92-93, pp. 371–374, 1999.
- [115] Y. Zhao, K. Ridgway, and A. M. A. Al-Ahmari, "Integration of CAD and a cutting tool selection system," *Computers and Industrial Engineering*, vol. 42, no. 1, pp. 17–34, 2002.
- [116] K. Edalew, H. Abdalla, and R. Nash, "A computer-based intelligent system for automatic tool selection," *Materials & Design*, vol. 22, no. 5, pp. 337–351, 2001.
- [117] W. Yang and Y. Tarng, "Design optimization of cutting parameters for turning operations based on the Taguchi method," *Journal of Materials Processing Technology*, vol. 84, no. 1-3, pp. 122–129, 1998.
- [118] D. M. D'Addona and R. Teti, "Genetic algorithm-based optimization of cutting parameters in turning processes," *Engineering Applications of Artificial Intelligence*, vol. 19, pp. 127–133, 2006.
- [119] M. Nalbant, H. Gökkaya, and G. Sur, "Application of Taguchi method in the optimization of cutting parameters for surface roughness in turning," *Materials and Design*, vol. 28, no. 4, pp. 1379–1385, 2007.
- [120] M. S. Chua, M. Rahman, Y. S. Wong, and H. T. Loh, "Determination of Optimal Cutting Conditions Using Design of Experiments and Optimization Techniques," *International Journal of Machine Tools & Manufacture*, vol. 33, no. 2, pp. 297–305, 1993.
- [121] N. S. Vele, R. K. Bawane, and J. V. Chopade, "Experimental Investigation of Optimum Cutting Conditions for T4 Medium Carbon Steel using TiN as a Cutting Tool," *IJSRD - International Journal for Scientific Research & Development*, vol. 4, no. 07, pp. 17–19, 2016.
- [122] G. Bala Subramanyam, "Optimization of Cutting Parameters Using Genetic Algorithm and Particle Swarm Optimization," *Ijmer*, vol. 4, no. 6, pp. 45–51, 2014.
- [123] R. Q. Sardiñas, P. Reis, and J. P. Davim, "Multi-objective optimization of cutting parameters for drilling laminate composite materials by using genetic algorithms," *Composites Science and Technology*, vol. 66, no. 15, pp. 3083–3088, 2006.
- [124] D. M. D'Addona and R. Teti, "Genetic algorithm-based optimization of cutting parameters in turning processes," *Procedia CIRP*, vol. 7, pp. 323–328, 2013.

- [125] A. Agrawal and S. Verma, "Development of Computer Aided Process Planning (CAPP) for optimization of machining parameters using a genetic algorithm approach," *International Journal of Advanced Research in Management and Social Sciences*, vol. 4, no. 6, pp. 178–186, 2015.
- [126] M. D. Tsai, M. Inui, and T. Sata, "110. Operation Planning Based on Cutting Process Models," *CIRP Annals - Manufacturing Technology*, vol. 40, no. 2, pp. 95–98, 1991.
- [127] A. Vignesh and P. K. Ram, "WELD CRACK RECTIFICATION IN CATALYTIC CONVERTER BY FABRICATION," *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY*, vol. 4, no. 5, pp. 370–377, 2015.
- [128] R. D. Bhosale, S. S. Nalawade, and P. Swami, "Study & Design of Jig and Fixture for Base frame of Canopy Fabrication of Generator," *International Research Journal of Engineering and Technology*, vol. 4, no. 5, p. 4, 2017.
- [129] N. D. Jayaweera, L. U. Subasinghe, and H. G. A. R. Gajanayaka, "Design and Development of a Novel Re-Configurable Fixturing System Need of Re-Configurable Fixturing Systems," *SAE Technical Paper*, 2015.
- [130] H. Hussein, A. Mahrous, A. F. Barakat, and O. Dawood, "Computer Aided Tradition Jigs and Fixture Design," in *17th International Conference on Applied Mechanics and Mechanical Engineering.*, no. April, 2016, p. 11.
- [131] M. Y. Wang and D. M. Pelinescu, "Optimizing Fixture Layout in a Point Set Domain," *IEEE Trans. on Robotics and Automation*, vol. 17(3), no. 3, pp. 312–323, 2001.
- [132] S. Ranga, M. Jaimini, S. K. Sharma, B. S. Chauhan, and A. Kumar, "A Review on Design of Experiments ( DOE )," *International Journal of Pharmaceutical and Chemical Sciences*, vol. 3, no. 1, pp. 216–224, 2014.
- [133] G. Qin, W. Zhang, and M. Wan, "Analysis and Optimal Design of Fixture Clamping Sequence," *Journal of Manufacturing Science and Engineering*, vol. 128, no. 2, p. 482, 2006.
- [134] E. K. Henriksen, *Jig and fixture design manual*. Industrial Press Inc., 1973.
- [135] N. Bhattacharyya and A. N. Poo, "APPLYING AI IN Jigs AND Fixtures," *Robotics & Computer-Integrated Manufacturing*, vol. 3, no. 2, p. 6, 1987.
- [136] H. Wang, Y. Rong, H. Li, and P. Shaun, "Computer aided fixture design: Recent research and trends," *CAD Computer Aided Design*, vol. 42, no. 12, pp. 1085–1094, 2010.
- [137] J. Cecil, "Computer-Aided Fixture Design - A Review and Future Trends," *The International Journal of Advanced Manufacturing Technology*, vol. 18, no. 11, pp. 790–793, 2001.
- [138] Y. K. CHOI, "TOOL PATH GENERATION AND 3D TOLERANCE ANALYSIS FOR FREE-FORM SURFACES," Ph.D. dissertation, 2004.
- [139] M. BALA and T.-C. CHANG, "Automatic cutter selection and optimal cutter path generation for prismatic parts," *International Journal of Production Research*, vol. 29, no. 11, pp. 2163–2176, 1991.
- [140] Z. Wu, M. Hobgood, and M. Wolf, "Energy Mapping and Optimization in Rough Machining of Impellers," in *ASME 2016 International Manufacturing Science and Engineering Conference*, 2016, pp. 1–10.

- [141] S. MARSHALL and J. G. GRIFFITHS, "A survey of cutter path construction techniques for milling machines," *International Journal of Production Research*, vol. 32, no. 12, pp. 2861–2877, 1994.
- [142] M. Sadílek, R. Čep, Z. Sadílková, J. Valíček, and L. Petřkovská, "Increasing tool life during turning with a variable depth of cut," *Materiali in Tehnologije*, vol. 47, no. 2, pp. 199–203, 2013.
- [143] M. Sadílek, J. Dubský, Z. Sadílková, and Z. Poruba, "Cutting forces during turning with variable depth of cut," *Perspectives in Science*, vol. 7, pp. 357–363, 2016.
- [144] J. Francis, T. E. Sparks, J. Ruan, and F. Liou, "Multi-axis tool path generation for surface finish machining of a rapid manufacturing process," *International Journal of Rapid Manufacturing*, vol. 4, no. 1, pp. 66–80, 2014.
- [145] A. H. GOMAA, "TIME-COST ESTIMATION SYSTEM FOR MACHINED PARTS USING CONVENTIONAL MACHINES," *JOURNAL OF ENGINEERING AND APPLIED SCIENCE*, vol. 50, no. 6, 2003.
- [146] H. a. Elmaraghy, "Evolution and Future Perspectives of CAPP," *CIRP Annals - Manufacturing Technology*, vol. 42, no. 2, pp. 739–751, 1993.
- [147] L. Wierda, "Linking design, process planning and cost information by feature-based modelling," *Journal of Engineering Design*, vol. 2 (1), no. July 2015, pp. 3–19, 1991.
- [148] P. Xirouchakis, D. Kiritsis, and J.-G. Persson, "A Petrinet Technique for Process Planning Cost Estimation," *CIRP Annals - Manufacturing Technology*, vol. 47, no. 1, pp. 427–430, 1998.
- [149] P. Xirouchakis, D. Kiritsis, C. Gunther, and J.-G. Persson, "A Petri Net Technique for Batch Delivery Time Estimation," *CIRP Annals - Manufacturing Technology*, vol. 48, no. 1, pp. 361–364, 1999.
- [150] D. Ben-Arieh and Q. Li, "Web-based cost estimation of machining rotational parts," *Production Planning & Control*, vol. 14, no. 8, pp. 778–788, 2003.
- [151] C. C. Cheung, "Semi-automated process planning and cost estimation of turned components based on CATIA V5 Machining," 2008.
- [152] F. Elgh and S. Sunnersjö, "AN AUTOMATED COST ESTIMATING SYSTEM FOR VARIANT DESIGN Knowledge , information flow and information transfer," in *INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN ICED 03 STOCKHOLM*, 2003, pp. 1–10.
- [153] M. Germani, M. Mandolini, and P. Cicconi, "Manufacturing cost estimation during early phases of machine design," *ICED 11 - 18th International Conference on Engineering Design - Impacting Society Through Engineering Design*, vol. 5, no. August, 2011.
- [154] B. Babic, N. Nesic, and Z. Miljkovic, "A review of automated feature recognition with rule-based pattern recognition," *Computers in Industry*, vol. 59, no. 4, pp. 321–337, 2008.
- [155] S. Marjudi, M. Amran, K. A. Abdullah, S. Widarto, N. A. A. Majid, and R. Sulaiman, "A Review and Comparison of IGES and STEP," in *World Academy of Science, Engineering and Technology 62 2010*, 2010, pp. 1013–1017.
- [156] *STEP Application Handbook - ISO 10303 - Version 3*. Scra, 2006, no. June. [Online]. Available: <http://www.engen.org.au/index{ }htm{ }files/STEP{ }application{ }hdbk{ }63006{ }BF.pdf>

- [157] *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.* Geneva, Switzerland: International Organization for Standardization (ISO TC 184/SC 4), 1994.
- [158] J. Oussama, E. Abdelilah, and R. Ahmed, “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. Journal of Innovation and Applied Studies*, vol. 8, no. 1, pp. 307–319, 2014.
- [159] M. P. Bhandarkar and R. Nagi, “STEP-based feature extraction from STEP geometry for Agile Manufacturing,” *Computers in Industry*, vol. 41, no. 1, pp. 3–24, 2000.
- [160] N. M. Vangipurapu, “Automatic feature recognition for rotational components from step files,” 2013. [Online]. Available: <http://shodhganga.inflibnet.ac.in/handle/10603/14116>
- [161] S. Reddy, D. V. Manidhar, V. K. Manupati, H. Raj, and A. Ali, “An Intelligent feature based process planning for Rotational parts,” in *Proceedings of the 2016 International Conference on Industrial Engineering and Operations Management*, no. March 8-10, 2016, pp. 2675–2685.
- [162] D. Sreeramulu and 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, pp. 102–115, 2011.
- [163] J. Gao, D. T. Zheng, N. Gindy, and D. Clark, “Extraction/conversion of geometric dimensions and tolerances for machining features,” *International Journal of Advanced Manufacturing Technology*, vol. 26, no. 4, pp. 405–414, 2005.
- [164] J. Boy, P. Rosché, K. C. Circle, E. Paff, B. Fischer, and D. Briggs, *CAX-IF Recommended Practices for the Representation and Presentation of Product Manufacturing Information (PMI) (AP242)*, 4th ed., 2014.
- [165] A. Armillotta, “A method for computer-aided specification of geometric tolerances,” *CAD Computer Aided Design*, vol. 45, no. 12, pp. 1604–1616, 2013. [Online]. Available: <http://dx.doi.org/10.1016/j.cad.2013.08.007>
- [166] A. Venkiteswaran, S. M. Hejazi, D. Biswas, J. J. Shah, and J. K. Davidson, “Semantic Interoperability of GD&T Data Through ISO 10303 Step AP242,” in *ASME 2016 International Design Engineering Technical Conferences*, Charlotte, North Carolina, 2016.
- [167] D. Biswas, A. Venkiteswaran, S. M. Hejazi, J. J. Shah, and J. K. Davidson, “Automated Iterative Tolerance Value Allocation and Analysis,” in *SME 2016 International Design Engineering Technical Conferences*, Charlotte, North Carolina, 2016.
- [168] S. M. Hejazi, D. Biswas, A. Venkiteswaran, J. J. Shah, and J. K. Davidson, “Automated 1st Order Tolerancing: Schema Generation,” in *ASME 2016 International Design Engineering Technical Conferences*, Charlotte, North Carolina, 2016.

# Appendix A

## The STEP file of the cylinder in fig.3.2

ISO-10303-21;

HEADER;

FILE\_DESCRIPTION (( 'STEP AP203' ),  
'1' );

FILE\_NAME ( 'Fig2.STEP',

'2017-11-09T22:52:35',

( 'Brunel University' ),

( 'Brunel University' ),

'SwSTEP 2.0',

'SolidWorks 2017',

");

FILE\_SCHEMA (( 'CONFIG\_CONTROL\_DESIGN' ));

ENDSEC;

DATA;

#1 = SHAPE\_DEFINITION\_REPRESENTATION ( #82, #221 );

#2 = PLANE ( 'NONE', #102 );

#3 = PLANE ( 'NONE', #247 );

#4 = CARTESIAN\_POINT ( 'NONE', ( 30.0000000000000000, 3.673940397442058400E-015,  
45.0000000000000000 ) );

#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', ( 0.000000000000000000, 0.000000000000000000, 1.000000000000000000
) );
#13 = CLOSED_SHELL ( 'NONE', ( #47, #23, #123, #198 ) );
#14 = LOCAL_TIME ( 22, 52, 35.000000000000000000, #72 );
#15 = MANIFOLD_SOLID_BREP ( 'Boss-Extrude1', #13 );
#16 = DIRECTION ( 'NONE', ( 1.000000000000000000, 0.000000000000000000, 0.000000000000000000
) );
#17 = APPROVAL_STATUS ( 'not_yet_approved' );
#18 = PRODUCT ( 'Fig2', 'Fig2', "", ( #54 ) );
#19 = COORDINATED_UNIVERSAL_TIME_OFFSET ( 0, 0, .AHEAD. );
#20 = APPROVAL ( #228, 'UNSPECIFIED' );
#21 = CYLINDRICAL_SURFACE ( 'NONE', #207, 30.000000000000000000 );
#22 = CC_DESIGN_DATE_AND_TIME_ASSIGNMENT ( #131, #150, ( #165 ) );
#23 = ADVANCED_FACE ( 'NONE', ( #6 ), #39, .T. );
#24 = EDGE_CURVE ( 'NONE', #55, #86, #121, .T. ); #25 = APPROVAL_PERSON_ORGANIZATION
( #80, #52, #173 );
#26 = CALENDAR_DATE ( 2017, 9, 11 );
#27 = DATE_AND_TIME ( #245, #49 );
#28 = EDGE_CURVE ( 'NONE', #42, #86, #8, .T. );
#29 = CARTESIAN_POINT ( 'NONE', ( 0.000000000000000000, 0.000000000000000000,
45.000000000000000000 ) );
```

---

```
#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', #98, 30.0000000000000000 );
#40 = PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE ( 'ANY', '', #18,
.NOT_KNOWN. );
#41 = DIRECTION ( 'NONE', ( 0.000000000000000000, 0.000000000000000000, 1.000000000000000000
) );
#42 = VERTEX_POINT ( 'NONE', #110 );
#43 = APPROVAL_PERSON_ORGANIZATION ( #107, #211, #76 );
#44 = PERSON ( 'UNSPECIFIED', 'UNSPECIFIED', 'UNSPECIFIED', ('UNSPECIFIED'), ('UN-
SPECIFIED'), ('UNSPECIFIED') );
#45 = CALENDAR_DATE ( 2017, 9, 11 );
#46 = AXIS2_PLACEMENT_3D ( 'NONE', #229, #101, #16 );
#47 = ADVANCED_FACE ( 'NONE', ( #94 ), #21, .T. );
#48 = CARTESIAN_POINT ( 'NONE', ( 0.000000000000000000, 0.000000000000000000,
45.0000000000000000 ) );
#49 = LOCAL_TIME ( 22, 52, 35.0000000000000000, #103 );
#50 = CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT ( #181, #220, ( #9 ) );
#51 = CARTESIAN_POINT ( 'NONE', ( 30.0000000000000000, 3.673940397442058400E-015,
0.000000000000000000 ) );
#52 = APPROVAL ( #37, 'UNSPECIFIED' );
#53 = DESIGN_CONTEXT ( 'detailed design', #227, 'design' );
```



---

```
#54 = MECHANICAL_CONTEXT ( 'NONE', #224, 'mechanical' );
#55 = VERTEX_POINT ( 'NONE', #217 );
#56 = CALENDAR_DATE ( 2017, 9, 11 );
#57 = PERSON_AND_ORGANIZATION_ROLE ( 'creator' );
#58 = APPLICATION_PROTOCOL_DEFINITION ( 'international standard', 'config_control_design',
1994, #175 );
#59 = EDGE_LOOP ( 'NONE', ( #136, #192, #161, #236 ) );
#60 = LOCAL_TIME ( 22, 52, 35.000000000000000000, #112 );
#61 = PRODUCT ( 'Fig2', 'Fig2', ", ( #151 ) );
#62 = CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT ( #126, #111, ( #9 ) );
#63 = CC_DESIGN_APPROVAL ( #52, ( #40 ) );
#64 = PERSON_AND_ORGANIZATION ( #44, #206 );
#65 = UNCERTAINTY_MEASURE_WITH_UNIT ( LENGTH_MEASURE( 1.000000000000000100E-
005 ), #91, 'distance_accuracy_value', 'NONE' );
#66 = ORIENTED_EDGE ( 'NONE', *, *, #28, .T. );
#67 = DIRECTION ( 'NONE', ( 1.000000000000000000, 0.000000000000000000, -0.000000000000000000
) );
#68 = DIRECTION ( 'NONE', ( 1.000000000000000000, 0.000000000000000000, 0.000000000000000000
) );
#69 = AXIS2_PLACEMENT_3D ( 'NONE', #215, #12, #152 );
#70 = CIRCLE ( 'NONE', #69, 30.000000000000000000 );
#71 = ORIENTED_EDGE ( 'NONE', *, *, #180, .T. );
#72 = COORDINATED_UNIVERSAL_TIME_OFFSET ( 0, 0, .AHEAD. );
#73 = DIRECTION ( 'NONE', ( 0.000000000000000000, 0.000000000000000000, 1.000000000000000000
) );
#74 = PERSON_AND_ORGANIZATION ( #44, #206 );
#75 = SECURITY_CLASSIFICATION ( ", ", #243 );
#76 = APPROVAL_ROLE ( " );
#77 = CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT ( #140, #57, ( #40 ) );
```

---

```
#78 = CARTESIAN_POINT ( 'NONE', ( 0.000000000000000000, 0.000000000000000000,
0.000000000000000000 ) );
#79 = DIRECTION ( 'NONE', ( 1.000000000000000000, 0.000000000000000000, -0.000000000000000000
) );
#80 = PERSON_AND_ORGANIZATION ( #44, #206 );
#81 = LOCAL_TIME ( 22, 52, 35.0000000000000000, #132 );
#82 = PRODUCT_DEFINITION_SHAPE ( 'NONE', 'NONE', #165 );
#83 = AXIS2_PLACEMENT_3D ( 'NONE', #48, #212, #68 );
#84 = ORIENTED_EDGE ( 'NONE', *, *, #174, .F. );
#85 = CALENDAR_DATE ( 2017, 9, 11 );
#86 = VERTEX_POINT ( 'NONE', #51 );
#87 = DATE_AND_TIME ( #139, #193 );
#88 = ( NAMED_UNIT ( * ) PLANE_ANGLE_UNIT ( ) SI_UNIT ( $, .RADIAN. ) );
#89 = PERSON_AND_ORGANIZATION ( #44, #206 );
#90 = CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT ( #74, #129, ( #75 ) );
#91 = ( LENGTH_UNIT ( ) NAMED_UNIT ( * ) SI_UNIT ( .MILLI., .METRE. ) );
#92 = APPROVAL_ROLE ( " );
#93 = CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT ( #89, #231, ( #40 ) );
#94 = FACE_OUTER_BOUND ( 'NONE', #251, .T. );
#95 = ORIENTED_EDGE ( 'NONE', *, *, #180, .F. );
#96 = DIRECTION ( 'NONE', ( 1.000000000000000000, 0.000000000000000000, 0.000000000000000000
) );
#97 = PERSON_AND_ORGANIZATION_ROLE ( 'design_owner' );
#98 = AXIS2_PLACEMENT_3D ( 'NONE', #156, #178, #100 );
#99 = DATE_TIME_ROLE ( 'creation_date' );
#100 = DIRECTION ( 'NONE', ( -1.000000000000000000, 0.000000000000000000, 0.000000000000000000
) );
#101 = DIRECTION ( 'NONE', ( 0.000000000000000000, 0.000000000000000000, 1.000000000000000000
) );
```

---

```
#102 = AXIS2_PLACEMENT_3D ( 'NONE', #155, #41, #79 );
#103 = COORDINATED_UNIVERSAL_TIME_OFFSET ( 0, 0, .AHEAD. );
#104 = CC_DESIGN_DATE_AND_TIME_ASSIGNMENT ( #7, #5, ( #75 ) );
#105 = SECURITY_CLASSIFICATION ( ", ", #122 );
#106 = LOCAL_TIME ( 22, 52, 35.000000000000000000, #186 );
#107 = PERSON_AND_ORGANIZATION ( #44, #206 );
#108 = CARTESIAN_POINT ( 'NONE', ( -30.000000000000000000, 0.000000000000000000,
45.000000000000000000 ) );
#109 = CARTESIAN_POINT ( 'NONE', ( -30.000000000000000000, 0.000000000000000000,
45.000000000000000000 ) );
#110 = CARTESIAN_POINT ( 'NONE', ( 30.000000000000000000, 3.673940397442058400E-015,
45.000000000000000000 ) );
#111 = PERSON_AND_ORGANIZATION_ROLE ( 'design_supplier' );
#112 = COORDINATED_UNIVERSAL_TIME_OFFSET ( 0, 0, .AHEAD. );
#113 =( GEOMETRIC_REPRESENTATION_CONTEXT ( 3 ) GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT
( ( #65 ) ) GLOBAL_UNIT_ASSIGNED_CONTEXT ( ( #91, #88, #241 ) ) REPRESENTA-
TION_CONTEXT ( 'NONE', 'WORKSPACE' ) );
#114 = APPROVAL ( #146, 'UNSPECIFIED' );
#115 = APPLICATION_PROTOCOL_DEFINITION ( 'international standard', 'config_control_design',
1994, #227 );
#116 = VERTEX_POINT ( 'NONE', #109 );
#117 = CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT ( #35, #248, ( #105 ) );
#118 = DESIGN_CONTEXT ( 'detailed design', #175, 'design' );
#119 = APPROVAL_DATE_TIME ( #170, #182 );
#120 = DIRECTION ( 'NONE', ( -0.000000000000000000, -0.000000000000000000, -1.000000000000000000
) );
#121 = CIRCLE ( 'NONE', #201, 30.000000000000000000 );
#122 = SECURITY_CLASSIFICATION_LEVEL ( 'unclassified' );
#123 = ADVANCED_FACE ( 'NONE', ( #219 ), #3, .T. );
```

---

```
#124 = CIRCLE ( 'NONE', #46, 30.0000000000000000 ); #125 = APPROVAL_PERSON_ORGANIZATION
( #177, #182, #133 );
#126 = PERSON_AND_ORGANIZATION ( #44, #206 );
#127 = CC_DESIGN_APPROVAL ( #114, ( #75 ) );
#128 = DIRECTION ( 'NONE', ( 1.0000000000000000, 0.0000000000000000, 0.0000000000000000
));
#129 = PERSON_AND_ORGANIZATION_ROLE ( 'classification_officer' );
#130 = CC_DESIGN_DATE_AND_TIME_ASSIGNMENT ( #32, #138, ( #105 ) );
#131 = DATE_AND_TIME ( #184, #230 );
#132 = COORDINATED_UNIVERSAL_TIME_OFFSET ( 0, 0, .AHEAD. );
#133 = APPROVAL_ROLE ( '' );
#134 = EDGE_CURVE ( 'NONE', #42, #116, #70, .T. );
#135 = CARTESIAN_POINT ( 'NONE', ( 0.0000000000000000, 0.0000000000000000,
45.0000000000000000 ) );
#136 = ORIENTED_EDGE ( 'NONE', *, *, #28, .F. );
#137 = CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT ( #164, #97, ( #18 ) );
#138 = DATE_TIME_ROLE ( 'classification_date' );
#139 = CALENDAR_DATE ( 2017, 9, 11 );
#140 = PERSON_AND_ORGANIZATION ( #44, #206 );
#141 = CC_DESIGN_SECURITY_CLASSIFICATION ( #75, ( #9 ) );
#142 = APPROVAL ( #17, 'UNSPECIFIED' );
#143 = ORIENTED_EDGE ( 'NONE', *, *, #24, .F. );
#144 = DIRECTION ( 'NONE', ( -0.0000000000000000, -0.0000000000000000, -1.0000000000000000
));
#145 = DIRECTION ( 'NONE', ( -1.0000000000000000, 0.0000000000000000, 0.0000000000000000
));
#146 = APPROVAL_STATUS ( 'not_yet_approved' );
#147 = AXIS2_PLACEMENT_3D ( 'NONE', #160, #171, #96 );
#148 = APPROVAL_PERSON_ORGANIZATION ( #200, #20, #92 );
```

---

```
#149 = PRODUCT_RELATED_PRODUCT_CATEGORY ( 'detail', '', ( #61 ) );
#150 = DATE_TIME_ROLE ( 'creation_date' );
#151 = MECHANICAL_CONTEXT ( 'NONE', #244, 'mechanical' );
#152 = DIRECTION ( 'NONE', ( 1.0000000000000000, 0.0000000000000000, 0.0000000000000000
) );
#153 = ORIENTED_EDGE ( 'NONE', *, *, #134, .T. );
#154 = CC_DESIGN_APPROVAL ( #142, ( #105 ) );
#155 = CARTESIAN_POINT ( 'NONE', ( 0.0000000000000000, 0.0000000000000000,
0.0000000000000000 ) );
#156 = CARTESIAN_POINT ( 'NONE', ( 0.0000000000000000, 0.0000000000000000,
45.0000000000000000 ) );
#157 = APPROVAL_DATE_TIME ( #204, #114 );
#158 = APPROVAL_ROLE ( '' );
#159 = PERSON_AND_ORGANIZATION ( #44, #206 );
#160 = CARTESIAN_POINT ( 'NONE', ( 0.0000000000000000, 0.0000000000000000,
0.0000000000000000 ) );
#161 = ORIENTED_EDGE ( 'NONE', *, *, #174, .T. );
#162 = APPROVAL_PERSON_ORGANIZATION ( #209, #114, #30 );
#163 = AXIS2_PLACEMENT_3D ( 'NONE', #78, #73, #205 );
#164 = PERSON_AND_ORGANIZATION ( #44, #206 );
#165 = PRODUCT_DEFINITION ( 'UNKNOWN', '', #40, #53 );
#166 = FACE_OUTER_BOUND ( 'NONE', #167, .T. );
#167 = EDGE_LOOP ( 'NONE', ( #143, #95 ) );
#168 = APPLICATION_PROTOCOL_DEFINITION ( 'international standard', 'config_control_design',
1994, #224 );
#169 = CC_DESIGN_SECURITY_CLASSIFICATION ( #105, ( #40 ) );
#170 = DATE_AND_TIME ( #214, #14 );
#171 = DIRECTION ( 'NONE', ( 0.0000000000000000, 0.0000000000000000, 1.0000000000000000
) );
```

---

```
#172 = CALENDAR_DATE ( 2017, 9, 11 ) ;
#173 = APPROVAL_ROLE ( " ) ;
#174 = EDGE_CURVE ( 'NONE', #116, #55, #176, .T. ) ;
#175 = APPLICATION_CONTEXT ( 'configuration controlled 3d designs of mechanical parts and
assemblies' ) ;
#176 = LINE ( 'NONE', #108, #213 ) ;
#177 = PERSON_AND_ORGANIZATION ( #44, #206 ) ;
#178 = DIRECTION ( 'NONE', ( -0.00000000000000000000, -0.00000000000000000000, -1.00000000000000000000
) ) ;
#179 = LOCAL_TIME ( 22, 52, 35.000000000000000000, #222 ) ;
#180 = EDGE_CURVE ( 'NONE', #86, #55, #124, .T. ) ;
#181 = PERSON_AND_ORGANIZATION ( #44, #206 ) ;
#182 = APPROVAL ( #246, 'UNSPECIFIED' ) ;
#183 = PRODUCT_RELATED_PRODUCT_CATEGORY ( 'detail', "", ( #18 ) ) ;
#184 = CALENDAR_DATE ( 2017, 9, 11 ) ;
#185 = APPROVAL_PERSON_ORGANIZATION ( #232, #142, #158 ) ;
#186 = COORDINATED_UNIVERSAL_TIME_OFFSET ( 0, 0, .AHEAD. ) ;
#187 = LOCAL_TIME ( 22, 52, 35.000000000000000000, #234 ) ;
#188 = CARTESIAN_POINT ( 'NONE', ( 0.00000000000000000000, 0.00000000000000000000,
0.00000000000000000000 ) ) ;
#189 = VECTOR ( 'NONE', #218, 1000.0000000000000000 ) ;
#190 = ORIENTED_EDGE ( 'NONE', *, *, #134, .F. ) ;
#191 = CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT ( #159, #38, ( #61 ) ) ;
#192 = ORIENTED_EDGE ( 'NONE', *, *, #235, .F. ) ;
#193 = LOCAL_TIME ( 22, 52, 35.000000000000000000, #239 ) ;
#194 = APPROVAL_DATE_TIME ( #240, #52 ) ;
#195 = CC_DESIGN_APPROVAL ( #182, ( #165 ) ) ;
#196 = APPROVAL_STATUS ( 'not_yet_approved' ) ;
#197 = PRODUCT_DEFINITION_SHAPE ( 'NONE', 'NONE', #199 ) ;
```

---

```
#198 = ADVANCED_FACE ( 'NONE', ( #166 ), #2, .F. );
#199 = PRODUCT_DEFINITION ( 'UNKNOWN', "", #9, #118 );
#200 = PERSON_AND_ORGANIZATION ( #44, #206 );
#201 = AXIS2_PLACEMENT_3D ( 'NONE', #188, #242, #128 );
#202 = DIRECTION ( 'NONE', ( 0.000000000000000000, 0.000000000000000000, 1.000000000000000000
) );
#203 = CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT ( #64, #10, ( #165 ) );
#204 = DATE_AND_TIME ( #252, #60 );
#205 = DIRECTION ( 'NONE', ( 1.000000000000000000, 0.000000000000000000, 0.000000000000000000
) );
#206 = ORGANIZATION ( 'UNSPECIFIED', 'UNSPECIFIED', "" );
#207 = AXIS2_PLACEMENT_3D ( 'NONE', #135, #120, #145 );
#208 = PERSON_AND_ORGANIZATION_ROLE ( 'creator' );
#209 = PERSON_AND_ORGANIZATION ( #44, #206 );
#210 = APPROVAL_DATE_TIME ( #225, #142 );
#211 = APPROVAL ( #196, 'UNSPECIFIED' );
#212 = DIRECTION ( 'NONE', ( 0.000000000000000000, 0.000000000000000000, 1.000000000000000000
) );
#213 = VECTOR ( 'NONE', #144, 1000.0000000000000000 );
#214 = CALENDAR_DATE ( 2017, 9, 11 );
#215 = CARTESIAN_POINT ( 'NONE', ( 0.000000000000000000, 0.000000000000000000,
45.0000000000000000 ) );
#216 = LOCAL_TIME ( 22, 52, 35.0000000000000000, #19 );
#217 = CARTESIAN_POINT ( 'NONE', ( -30.0000000000000000, 0.000000000000000000,
0.000000000000000000 ) );
#218 = DIRECTION ( 'NONE', ( -0.000000000000000000, -0.000000000000000000, -1.000000000000000000
) ); #219 = FACE_OUTER_BOUND ( 'NONE', #33, .T. );
#220 = PERSON_AND_ORGANIZATION_ROLE ( 'creator' );
#221 = ADVANCED_BREP_SHAPE_REPRESENTATION ( 'Fig2', ( #15, #163 ), #113 );
```

---

```
#222 = COORDINATED_UNIVERSAL_TIME_OFFSET ( 0, 0, .AHEAD. ) ;
#223 = CC_DESIGN_APPROVAL ( #211, ( #199 ) ) ;
#224 = APPLICATION_CONTEXT ( 'configuration controlled 3d designs of mechanical parts and
assemblies' ) ;
#225 = DATE_AND_TIME ( #26, #81 ) ;
#226 = APPROVAL_DATE_TIME ( #27, #20 ) ;
#227 = APPLICATION_CONTEXT ( 'configuration controlled 3d designs of mechanical parts and
assemblies' ) ;
#228 = APPROVAL_STATUS ( 'not_yet_approved' ) ;
#229 = CARTESIAN_POINT ( 'NONE', ( 0.000000000000000000, 0.000000000000000000,
0.000000000000000000 ) ) ;
#230 = LOCAL_TIME ( 22, 52, 35.0000000000000000, #31 ) ;
#231 = PERSON_AND_ORGANIZATION_ROLE ( 'design_supplier' ) ;
#232 = PERSON_AND_ORGANIZATION ( #44, #206 ) ;
#233 = PERSON_AND_ORGANIZATION ( #44, #206 ) ;
#234 = COORDINATED_UNIVERSAL_TIME_OFFSET ( 0, 0, .AHEAD. ) ;
#235 = EDGE_CURVE ( 'NONE', #116, #42, #253, .T. ) ;
#236 = ORIENTED_EDGE ( 'NONE', *, *, #24, .T. ) ;
#237 = CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT ( #233, #208, ( #199 ) ) ;
#238 = ORIENTED_EDGE ( 'NONE', *, *, #235, .T. ) ;
#239 = COORDINATED_UNIVERSAL_TIME_OFFSET ( 0, 0, .AHEAD. ) ;
#240 = DATE_AND_TIME ( #45, #179 ) ;
#241 = ( NAMED_UNIT ( * ) SI_UNIT ( $, .STERADIAN. ) SOLID_ANGLE_UNIT ( ) ) ;
#242 = DIRECTION ( 'NONE', ( 0.000000000000000000, 0.000000000000000000, 1.000000000000000000
) ) ;
#243 = SECURITY_CLASSIFICATION_LEVEL ( 'unclassified' ) ;
#244 = APPLICATION_CONTEXT ( 'configuration controlled 3d designs of mechanical parts and
assemblies' ) ;
#245 = CALENDAR_DATE ( 2017, 9, 11 ) ;
```



---

```
#246 = APPROVAL_STATUS ( 'not_yet_approved' );
#247 = AXIS2_PLACEMENT_3D ( 'NONE', #29, #202, #67 );
#248 = PERSON_AND_ORGANIZATION_ROLE ( 'classification_officer' );
#249 = CC_DESIGN_DATE_AND_TIME_ASSIGNMENT ( #250, #99, ( #199 ) );
#250 = DATE_AND_TIME ( #56, #106 );
#251 = EDGE_LOOP ( 'NONE', ( #190, #66, #71, #84 ) );
#252 = CALENDAR_DATE ( 2017, 9, 11 );
#253 = CIRCLE ( 'NONE', #83, 30.000000000000000000 );
ENDSEC;
END-ISO-10303-21;
```

## Appendix B

### The details of the resultant STEP file AP 203 of the cylinder in fig.3.2after applying the parser

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 CurveRadius = 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 CurveRadius = 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 CurveRadius = 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\_LOOP #136 #192 #161 #236  
#136 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  
#192 ORIENTED\_EDGE #235  
#235 EDGE\_CURVE #116 #42 #253  
#116 VERTEX\_POINT #109  
#109 CARTESIAN\_POINT X=-30 Y=0 Z=45  
#42 VERTEX\_POINT #110  
#110 CARTESIAN\_POINT X=30 Y=0 Z=45  
#253 CIRCLE #83 CurveRadius = 30  
#83 AXIS2\_PLACEMENT\_3D #48 #212 #68  
#48 CARTESIAN\_POINT X=0 Y=0 Z=45  
#212 DIRECTION X=0 Y=0 Z=1  
#68 DIRECTION X=1 Y=0 Z=0  
#161 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  
#236 ORIENTED\_EDGE #24  
#24 EDGE\_CURVE #55 #86 #121  
#55 VERTEX\_POINT #217  
#217 CARTESIAN\_POINT X=-30 Y=0 Z=0  
#86 VERTEX\_POINT #51  
#51 CARTESIAN\_POINT X=30 Y=0 Z=0  
#121 CIRCLE #201 CurveRadius = 30  
#201 AXIS2\_PLACEMENT\_3D #188 #242 #128  
#188 CARTESIAN\_POINT X=0 Y=0 Z=0  
#242 DIRECTION X=0 Y=0 Z=1  
#128 DIRECTION X=1 Y=0 Z=0  
#39 CYLINDRICAL\_SURFACE #98 CurveRadius = 30  
#98 AXIS2\_PLACEMENT\_3D #156 #178 #100  
#156 CARTESIAN\_POINT X=0 Y=0 Z=45  
#178 DIRECTION X=0 Y=0 Z=-1  
#100 DIRECTION X=-1 Y=0 Z=0  
F3=#123  
#123 ADVANCED\_FACE #219 #3  
#219 FACE\_OUTER\_BOUND #33  
#33 EDGE\_LOOP #238 #153  
#238 ORIENTED\_EDGE #235  
#235 EDGE\_CURVE #116 #42 #253  
#116 VERTEX\_POINT #109

---

#109 CARTESIAN\_POINT X=-30 Y=0 Z=45  
#42 VERTEX\_POINT #110  
#110 CARTESIAN\_POINT X=30 Y=0 Z=45  
#253 CIRCLE #83 CurveRadius = 30  
#83 AXIS2\_PLACEMENT\_3D #48 #212 #68  
#48 CARTESIAN\_POINT X=0 Y=0 Z=45  
#212 DIRECTION X=0 Y=0 Z=1  
#68 DIRECTION X=1 Y=0 Z=0  
#153 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 CurveRadius = 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  
#3 PLANE #247  
#247 AXIS2\_PLACEMENT\_3D #29 #202 #67  
#29 CARTESIAN\_POINT X=0 Y=0 Z=45  
#202 DIRECTION X=0 Y=0 Z=1  
#67 DIRECTION X=1 Y=0 Z=0  
F4=#198  
#198 ADVANCED\_FACE #166 #2  
#166 FACE\_OUTER\_BOUND #167  
#167 EDGE\_LOOP #143 #95  
#143 ORIENTED\_EDGE #24

---

#24 EDGE\_CURVE #55 #86 #121  
#55 VERTEX\_POINT #217  
#217 CARTESIAN\_POINT X=-30 Y=0 Z=0  
#86 VERTEX\_POINT #51  
#51 CARTESIAN\_POINT X=30 Y=0 Z=0  
#121 CIRCLE #201 CurveRadius = 30  
#201 AXIS2\_PLACEMENT\_3D #188 #242 #128  
#188 CARTESIAN\_POINT X=0 Y=0 Z=0  
#242 DIRECTION X=0 Y=0 Z=1  
#128 DIRECTION X=1 Y=0 Z=0  
#95 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 CurveRadius = 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  
#2 PLANE #102  
#102 AXIS2\_PLACEMENT\_3D #155 #41 #79  
#155 CARTESIAN\_POINT X=0 Y=0 Z=0  
#41 DIRECTION X=0 Y=0 Z=1  
#79 DIRECTION X=1 Y=0 Z=0

# Appendix C

## The resultant G-code of the model design in fig.4.18

O0001

(Case 1)

(MACHINE EXTERNAL AND INTERNAL)

(STOCK DIA 176 mm)

(SET THE RIGHT SIDE OF STOCK AS 102)

N0 G0 G28 U0.

N5 G0 T0101

N10 G18

N15 G97 S1000 M03

N20 G0 G54 Z103

N25 G0 X178

(Intermediate Feature)

N30 G71 U3.0 R1.0

N35 G71 P40 Q50 U0.5 W0.5 F.1

N40 G1 X160 Z100

N45 G1 X160 Z54

N50 G3 X175 Z46.5 R7.5

N55 G0 Z103

(Intermediate Feature)

N60 G0 X160



---

N65 G71 U3.0 R1.0

N70 G71 P75 Q85 U0.5 W0.5 F.1

N75 G1 X86 Z100

N80 G1 X86 Z58

N85 G1 X160 Z58

N90 G0 Z103

(Intermediate Feature)

N95 G0 X86

N100 G71 U3.0 R1.0

N105 G71 P110 Q120 U0.5 W0.5 F.1

N110 G1 X80 Z100

N115 G1 X80 Z64

N120 G1 X86 Z58

N125 G0 Z103

(Right convex corner)

N130 G0 X80

N135 G71 U3.0 R1.0

N140 G71 P145 Q155 U0.5 W0.5 F.1

N145 G0 X72 Z103

N150 G1 X72 Z100

N155 G3 X80 Z96 R4

N160 G0 Z103

(Facing)

N165 G0 X72

N170 G72 W1.0 R.5

N175 G72 P180 Q190 U0.0 W0.5 F.1

N180 G0 Z100

N185 G1 X-1

N190 G1 Z103

(Radial square groove)

N195 G0 X176

N200 G28 U0.0

N205 M5

N210 M9

N215 M1

N220 G0 G28 U0.0

N225 G0 0202

N230 G18

N235 G97 S1000 M03

N240 G0 G54 Z83

N245 G0 X178

N250 G0 X80

N255 G75 R1.0

N260 G75 X55 Z78 P5 Q5 F.1

N265 G0 Z83

(Intermediate Feature)

N270 G0 X178

N275 G0 Z0

N280 G71 U3.0 R1.0

N285 G71 P290 Q300 U0.5 W0.5 F.1

N290 G1 X160 Z0

N295 G1 X160 Z34

N300 G2 X175 Z41.5 R7.5

N305 G0 Z0

(Radial groove two rounded corners)

N310 G0 X178

N315 G0 X160

N320 G0 Z30

---

N325 G71 U3.0 R1.0

N330 G71 P335 Q355 U0.5 W0.5 F.1

N335 G1 X90 Z30

N340 G2 X80 Z25 R5

N345 G1 X80 Z25

N350 G2 X90 Z20 R5

N355 G1 X160 Z20

N360 G0 Z30

(Two sides tapered groove)

N365 G0 X176

N370 G28 U0.0

N375 M5

N380 M9

N385 M1

N390 G0 G28 U0.0

N395 G0 0303

N400 G18

N405 G97 S1000 M03

N410 G0 G54 Z15.5

N415 G0 X178

N420 G0 X160

N425 G71 U3.0 R1.0

N430 G71 P435 Q445 U0.5 W0.5 F.1

N435 G1 X150 Z12

N440 G1 X150 Z8

N445 G1 X160 Z4.5

N450 G0 Z15.5

(Intermediate Feature)

N455 G0 X176

N460 G28 U0.0

N465 M5

N470 M9

N475 M1

N480 G0 G28 U0.0

N485 G0 0404

N490 G18

N495 G97 S1000 M03

N500 G0 G54 Z103

N505 G0 X0

N510 G1 Z0F0.1

N515 G0 Z103

(Right axial groove)

N525 G0 Z103

N530 G28 U0.0

N535 M5

N540 M9

N545 M1

N550 G0 G28 U0.0

N555 G0 0505

N560 G18

N565 G97 S1000 M03

N570 G0 X0

N575 G0 G54 Z100

N580 G0 X24

N585 G71 U3.0 R1.0

N590 G71 P595 Q605 U0.5 W0.5 F.1

N595 G1 X34 Z100

N600 G1 X34 Z88

N605 G1 X24 Z88

N610 G0 Z100

(Two sides tapered groove)

N620 G0 Z103

N625 G28 U0.0

N630 M5

N635 M9

N640 M1

N645 G0 G28 U0.0

N650 G0 0606

N655 G18

N660 G97 S1000 M03

N665 G0 X0

N670 G0 G54 Z15.5

N675 G0 X24

N680 G71 U3.0 R1.0

N685 G71 P690 Q700 U0.5 W0.5 F.1

N690 G1 X34 Z12

N695 G1 X34 Z8

N700 G1 X24 Z4.5

N705 G0 Z15.5

N710 G0 Z103

(Parting)

N715 G0 X176

N720 G28 U0.0

N725 M5

N730 M9

N735 M1

N740 G0 G28 U0.0

N745 G0 0202

N750 G18 N755 G97 S1000 M03

N760 G0 G54 Z-5

N765 G0 X178

N770 G1 X-1.0

N775 G28 U0.0

N780 M5

N785 M9

N790 M30

%

# Appendix D

## The resultant G-code of the model design used in the practical validation

O0001

(Practical validation)

(MACHINE EXTERNAL AND INTERNAL)

(STOCK DIA 120 mm)

(SET THE RIGHT SIDE OF STOCK AS 102)

N0 G0 G28 U0.

N5 G0 T0101

N10 G97 S1000 M03

N15 G0 Z103

N20 G0 X122

(Intermediate Feature)

N25 G71 U0.6 R0.3

N30 G71 P35 Q45 U0.0 W0.0 F.18

N35 G1 X110 Z100

N40 G1 X110 Z51

N45 G3 X119 Z46.5 R4.5

N50 G0 Z103

(Intermediate Feature)

N55 G0 X110

N60 G71 U0.6 R0.3

N65 G71 P70 Q80 U0.0 W0.0 F.18

N70 G1 X86 Z100

N75 G1 X86 Z58

N80 G1 X110 Z58

N85 G0 Z103

(Intermediate Feature)

N90 G0 X86

N95 G71 U0.6 R0.3

N100 G71 P105 Q115 U0.0 W0.0 F.18

N105 G1 X80 Z100

N110 G1 X80 Z64

N115 G1 X86 Z58

N120 G0 Z103

(Right convex corner)

N125 G0 X80

N130 G71 U0.6 R0.3

N135 G71 P140 Q150 U0.0 W0.0 F.18

N140 G0 X72 Z103

N145 G1 X72 Z100

N150 G3 X80 Z96 R4

N155 G0 Z103

(Facing)

N160 G0 X72

N165 G72 W1.0 R.5

N170 G72 P175 Q185 U0.0 W0.5 F.1

N175 G0 Z100

N180 G1 X-1

N185 G1 Z103

(Radial square groove)



N190 G0 X120

N195 G28 U0.0

N200 M5

N205 M9

N210 M1

N215 G0 G28 U0.0

N220 G0 T0202

N225 G97 S1000 M03

N230 G0 Z86

N235 G0 X122

N240 G0 X80

N245 G75 R0.3

N250 G75 X64 Z82 P1000 Q1000 F.18

N255 G0 Z86

(Intermediate Feature)

N260 G0 X122

N265 G0 Z0

N270 G71 U0.6 R0.3

N275 G71 P280 Q290 U0.0 W0.0 F.18

N280 G1 X110 Z0

N285 G1 X110 Z40

N290 G2 X119 Z44.5 R4.5

N295 G0 Z0

(Radial groove two rounded corners)

N300 G0 X122

N305 G0 X110

N310 G0 Z33

N315 G71 U0.6 R0.3

N320 G71 P325 Q345 U0.0 W0.0 F.18

N325 G1 X96 Z33

N330 G2 X90 Z30 R3

N335 G1 X90 Z28

N340 G2 X96 Z25 R3

N345 G1 X110 Z25

N350 G0 Z33

(Two sides tapered groove)

N355 G0 X120

N360 G28 U0.0

N365 M5

N370 M9

N375 M1

N380 G0 G28 U0.0

N385 G0 T0303

N390 G97 S1000 M03

N395 G0 Z20

N400 G0 X122

N405 G0 X110

N410 G71 U0.6 R0.3

N415 G71 P420 Q430 U0.0 W0.0 F.18

N420 G1 X104 Z16

N425 G1 X104 Z9

N430 G1 X110 Z5

N435 G0 Z20

(Intermediate Feature)

N440 G0 X120

N445 G28 U0.0

N450 M5

N455 M9

---

N460 M1

N465 G0 G28 U0.0

N470 G0 T0404

N475 G97 S1000 M03

N480 G0 Z103

N485 G0 X0

N490 G1 Z0F0.1

N495 G0 Z103

(Right axial groove)

N505 G0 Z103

N510 G28 U0.0

N515 M5

N520 M9

N525 M1

N530 G0 G28 U0.0

N535 G0 T0505

N540 G97 S1000 M03

N545 G0 X0

N550 G0 Z100

N555 G0 X28.5

N560 G71 U0.6 R0.3

N565 G71 P570 Q580 U0.0 W0.0 F.18

N570 G1 X34 Z100

N575 G1 X34 Z95

N580 G1 X28.5 Z95

N585 G0 Z100

(Two sides tapered groove)

N595 G0 Z103

N600 G28 U0.0

N605 M5

N610 M9

N615 M1

N620 G0 G28 U0.0

N625 G0 T0606

N630 G97 S1000 M03

N635 G0 X0

N640 G0 Z90

N645 G0 X28.5

N650 G71 U0.6 R0.3

N655 G71 P660 Q670 U0.0 W0.0 F.18

N660 G1 X34 Z86

N665 G1 X34 Z79

N670 G1 X28.5 Z75

N675 G0 Z90

N680 G0 Z103

(Parting)

N685 G0 X120

N690 G28 U0.0

N695 M5

N700 M9

N705 M1

N710 G0 G28 U0.0

N715 G0 T0202

N720 G97 S1000 M03

N725 G0 Z-2

N730 G0 X122

N735 G1 X-1.0

N740 G28 U0.0

N745 M5

N750 M9

N755 M30 %