

77712

**DEVELOPMENT OF A PROCESS PLANNING SYSTEM
FOR PRISMATIC PARTS**

A Ph. D. THESIS

in

77712

**Mechanical Engineering
University of Gaziantep**

© by

Türkay DERELİ

June 1998

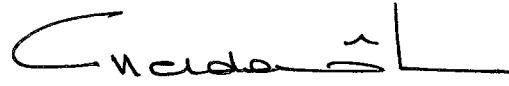
Approval of the Graduate School of Natural and Applied Sciences



Assoc. Prof. Dr. A. Rıza Tekin

Director

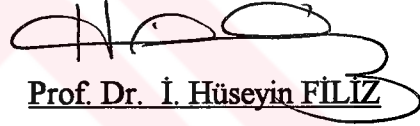
I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Assoc. Prof. Dr. Melda Çarpınlioğlu

Chairman of the Department

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

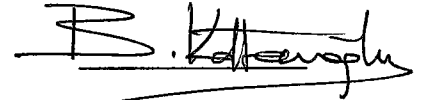


Prof. Dr. İ. Hüseyin FİLİZ

Supervisor

Examining Committee in Charge:

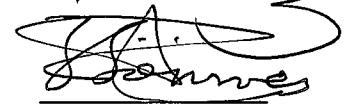
Prof. Dr. Bilgin KAFTANOĞLU (Chairman)



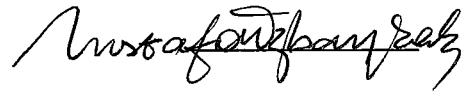
Prof. Dr. İ. Hüseyin FİLİZ



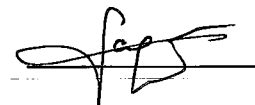
Prof. Dr. A. İhsan SÖNMEZ



Assoc. Prof. Dr. Mustafa ÖZBAYRAK



Assist. Prof. Dr. Cengiz DOĞAN



DECLARATION

No part of the work described in this thesis has been submitted in support of an application for another degree or qualification of University of Gaziantep or any other university or any other institution.

©

by Trkay DERELİ, June 1998

All rights reserved. No part of this thesis may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior *written* permission of the author. The copyright also covers the exclusive rights to reproduce and distribute reproductions of Turkish translation of this thesis.

All texts, drawings, and equations in this manuscript were prepared and typeset electronically by using MS-Word 7.0 on a Windows 95 operated Personal Computer, in Computer Aided Education Laboratory, University of Gaziantep, Gaziantep, Turkey.

ABSTRACT

DEVELOPMENT OF A PROCESS PLANNING SYSTEM FOR PRISMATIC PARTS

DERELİ, Turkey

Ph.D. in Mechanical Engineering

Supervisor: Prof. Dr. İ. Hüseyin FİLİZ

June 1998, 278 pages

In this thesis, a Computer Aided Process Planning (CAPP) package for prismatic parts is developed together with its interfaces to provide a CAD/CAM integration. The developed package is composed of five main sections. The upmost and starting unit includes a *modelling (CAD platform)* and a *feature recognition (CAD/CAPP interface)* based on STEP (STandard for the Exchange of Product model data) data. The second-section modules work under the first-unit modules and involve *usual process planning facilities* such as “workpiece size selection”, “tool selection” and “set-up planning”. The third section is responsible for *optimisation of process planning events* performed in the second section. It makes use of an Artificial Intelligence (AI) tool called Genetic Algorithms (GAs). A *Design for manufacturing (DFM) utility* including elimination of critical regions on the component is presented in the fourth section which has a floating nature. It may not be used in working cycle of the process planning or can be used at any stage of CAD/CAM integration for different purposes. The last and down-most section in the CAD/CAM integration is *CAPP/CAM interface* that consists of the generation, verification and down-loading of CNC codes. Input to this section is a generalised CAD/CAM data base which is prepared at the end of the process planning and optimisation sessions performed in the upstream units.

Primary objective of this research is to develop an intelligent and integrated CAD/CAM system for shop-floor use that can be used by an average operator and produce fully (globally) optimised results (process plans and part programs). For this

purpose, an attempt has been made to include the impact and potential of AI in process planning applications and to optimise all events (parameters) in an integrated CAD/CAM environment. GAs are extensively used together with conventional programming techniques in the development of process planning and optimisation facilities in order to include profits of AI techniques into the system.

The developed package has a modular structure. A computer program is prepared for each of the modules forming the sections of entire system specified above. The programs are written in C language. The modules are combined under a supervisory program. The system is developed on a PC and implemented on a VMC (Vertical Milling Center). The use of developed CAPP system is illustrated with practical examples.

Key Words: Computer Aided Design (CAD), Feature Recognition, Computer Aided Process Planning (CAPP), Computer Aided Manufacturing (CAM), Genetic Algorithms (GAs).

ÖZET

PRİZMATİK İŞ PARÇALARI İÇİN BİR İŞLEM PLANLAMA SİSTEMİ GELİŞTİRİLMESİ

DERELİ, Trkay

Doktora Tezi, Makina Mhendislięi Blm

Tez yneticisi: Prof. Dr. İ. Hseyin FİLİZ

Haziran 1998, 278 sayfa

Bu tezde, BDT/BD (Bilgisayar Destekli Tasarım/Bilgisayar Destekli retim) entegrasyonunu saęlamak amacıyla, prizmatik iř paraları iin arayzleri ile birlikte bir BDİP (Bilgisayar Destekli İřlem Planlama) yazılım paketi geliřtirilmiřtir. Geliřtirilen paket beř ana blmden oluřmaktadır. En stte ve bařlangıta kullanılan blm bir modelleme (BDT platformu) ve STEP (STandard for the Exchange of Product model data) standardına dayalı bir Unsur Algılama (BDT/BDİP arayz) sistemini iermektedir. İkinici blm birinci blmn altında alıřır ve, “iř parası boyut seimi”, “takım seimi” ve “operasyon dzeneęi” planlama gibi genel iřlem planlama uygulamalarını ierir. nc blmdaki modller ikinci nite modllerince bulunan plan ve parametrelerin eniyilenmesi iin hazırlanmıř olup, yapay zeka tekniklerinden biri olan genetik algoritmaları kullanmaktadır. Fonksiyonu para zerindeki kritik blgelerin ortadan kaldırılması olan bir “retime Ynelik Tasarım” modl drdnc blmde yer almaktadır. Bu modl *yzen* bir yapıya sahiptir. Bu yzden iřlem planlama evriminde kullanılmayabilir ya da farklı amalar iin sistemin herhangi bir basamaęında kullanılabilir. En altta alıřan beřinci ve son blm BDİP/BD arayzdr ve CNC kodlarının tretimi, kontrol edilmesi ve tezgaha yklenmesi iin kullanılır. İlk drt blm tarafından hazırlanan genelleřtirilmiř BDT/BD veri tabanı, beřinci blmn girdisidir.

Bu alıřmanın temel amacı, tmyle eniyilenmiř sonular (iřlem planları ve para programları) retebilen, atelyede tezgah bařında ortalama bir operatr tarafından kullanılabilir, akıllı ve entegre bir BDT/BD ortamının geliřtirilmesidir.

Bu amaçla, bu çalışmada, yapay zeka tekniklerinin potansiyel gücünün işlem planlama uygulamalarında kullanılmasına ve entegre bir BDT/BDÜ sistemindeki tüm uygulama ya da parametrelerin eniyilenmesine çalışılmıştır. Konvansiyonel programlama teknikleri ile birlikte genetik algoritmalar kullanılarak yapay zeka tekniklerinin potansiyel faydalarının işlem planlama ve eniyileme uygulamalarına yansıtılmasına çalışılmıştır.

Geliştirilen paket modüler bir yapıya sahiptir. Yukarıda bahsedilen, sistemi meydana getiren bölümleri oluşturan her bir modül için bir bilgisayar programı hazırlanmıştır. Programlar C programlama dili ile yazılmıştır. Bütün modüller bir ana program altında birleştirilmiştir. Sistem kişisel bilgisayar üzerinde ve bir *Dik İşleme Merkezi* için geliştirilmiştir. Geliştirilen sistemin kullanımı örneklerle gösterilmiştir.

Anahtar Kelimeler: Bilgisayar Destekli Tasarım (BDT), Unsur Algılama, Bilgisayar Destekli İşlem Planlama (BDİP), Bilgisayar Destekli Üretim (BDÜ), Genetik Algoritmalar.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, currently the President of University of Gaziantep, Prof. Dr. İ. Hüseyin Filiz, this thesis would never have been completed without his valuable encouragement, support and guidance. Special words of thanks are due to the Research Fund of University of Gaziantep and (TÜBİTAK) - Foundation of Münir Birsal, for their exceptional help to make this work possible by providing partial financial supports.

I am thankful to Assist. Prof. Dr. Nihat Yıldırım, Assist. Prof. Dr. Ömer Eyercioğlu, Assist. Prof. Dr. Günseli Görür, Assist. Prof. Dr. Abdullah Akpolat, Assist. Prof. Dr. Cengiz Doğan and Mr. N. Fazıl Yılmaz for their continuous motivation, support and encouragement. I should declare herein my gratitudes to Prof. Dr. Bilgin Kaftanoğlu, The Founder of CAD/CAM & Robotics Center at METU, who first brought about the CAD/CAM to Turkey and verified its importance for Turkish Industry, for his pioneering work in the CAD/CAM applications. Many sincere thanks go to Prof. Dr. Erdem Koç for his motivation and being an example to how to write papers quickly and how a scientist can be energetic. I also wish to acknowledge with grateful thanks to; Prof. Dr. A. İhsan Sönmez and Assoc. Prof. Dr. Mustafa Özbayrak for their suggestions. Thanks are due to Mr. Adil Baykasoğlu, for his assistance, supplement of new literature on CAD/CAM, CAPP, AI, and his helpful discussions. Thanks are due to hardworking assistants; Mr. A. Tolga Bozdana and Mr. Oğuzhan Yılmaz for their helps in the preparation of this manuscript and for keeping the spirit of CAD/CAM Research Group alive.

I owe to my brothers, Levent Dereli and Mehmet Dereli, much of my appreciation for their supports. However, this work was only finished due to the financial support of my father, Muammer Dereli, and my mother, Fatma Dereli, who still manages me after 30 years. I would like to express finally my appreciation to my wife, Zeliha Dereli and my daughter, Elif Dereli for their unfailing support, understanding, encouragement and patience during many days, weeks and years that I spent in the laboratory on this thesis. Their love sustained me throughout the work. Thanks and apologies to others whose contributions I have overlooked here...



to:

*my wife, Zeliha
my daughter, Elif
my mother, Fatma
my father, Muammer
my brothers, Mehmet and Levent
and all others in my family*

TABLE OF CONTENTS

DECLARATION	<i>Page</i> iii
ABSTRACT	iv
ÖZET	vi
ACKNOWLEDGEMENTS	viii
LIST OF TABLES	xiv
LIST OF FIGURES	xvi
NOMENCLATURE	xviii
1. SCOPE OF THE WORK AND OBJECTIVES	1
1.1. Introduction	1
1.2. Importance of Process Planning in CIM and IMS	1
1.3. Frame of the Work	5
1.4. Objectives of the Work	8
1.5. Methodology and Materials	9
1.6. Organisation of the Thesis	11
2. LITERATURE SURVEY	13
2.1. Introduction	13
2.2. Literature on “Feature Recognition”	13
2.2.1. Historical Background	13
2.2.2. Work on the Feature Recognition	14
2.3. Literature on “Workpiece Size Selection”	17
2.4. Literature on “Operation Sequencing”	18
2.5. Literature on “Tooling”	20
2.6. Literature on “Set-Up Planning”	21
2.7. Literature on “Cutting Conditions”	23
2.8. Literature on “CNC Code Generation”	26
2.9. Review of CAD/CAM Integration from a Different Perspective	29
3. ARTIFICIAL INTELLIGENCE AND ITS IMPACT ON CAD/CAM	33
3.1. Introduction	33
3.2. Artificial Intelligence and Its Techniques	33
3.2.1. Expert Systems	34
3.2.2. Artificial Neural Networks	37
3.2.3. Fuzzy Logic	42
3.2.4. Genetic Algorithms	44
3.4. Current Status of AI Technology and Its Feature	46

4. A FEATURE RECOGNITION SYSTEM FOR PRISMATIC PARTS	49
4.1. Introduction	49
4.2 Computer Aided Design/Drafting (CAD/D)	49
4.3 Geometric Modeling	50
4.3.1. Importance of Solid Modeling	52
4.4 Feature Based Design and Feature Recognition	51
4.4.1. Definition of a Feature	54
4.4.2. Manufacturing Features	54
4.4.3. Part Representation Using Features	55
4.5 Data Exchange Standards and STEP Standard Review	58
4.5.1. The Need for the Data Exchange	58
4.5.2. Functions of Data Exchange Standards	59
4.5.3. DXF (Data eXchange File) Standard	60
4.5.4. IGES (Initial Graphics Exchange Specification) Standard	61
4.5.5. SET (Standard d'echange Et de Transfert) Standard	63
4.5.6. VDAFS (Verband Der Automobilindustrie Flaechen Schnittstelle) Standard	63
4.5.7. CAD*I (Computer Aided Design * Interface) Standard	63
4.5.8. PDDI (Product Definition Data Interface) Standard	63
4.5.9. PDES (Product Data Exchange Specification) Standard	64
4.5.10 STEP (STandard for Exchange of Product data) Standard	65
4.6. 3D Prismatic Part Definition in the STEP Standard	70
4.7. Product Modelling Environment Used in this Thesis: Pro/Engineer	74
4.8. FRS-PP: A Feature Recognizer for Prismatic Parts	76
4.8.1. Methodology Used in the FRS-PP via an Example	77
4.9. Validation of B-Rep Models by Using Euler's Law	82
5. PROCESS PLANNING MODULES	84
5.1. Introduction	84
5.2. Computer Aided Process Planning	84
5.2.1. Approaches to Process Planning	85
5.2.2. Implementation Techniques of CAPP	86
5.3. Workpiece Size Selection Module	88
5.4. Operation Selection and Sequencing Module	91
5.4.1. Computer Aided Determination of Sequence of Operations	91

5.4.2. Selection of Machining Operations	92
5.5. Tool Selection Module	97
5.5.1. Specification of Workpiece Material	100
5.5.2. Determination of Cutter-Type Domain for Features	100
5.5.3. Determination of Cutter Material	102
5.5.4. Determination of Characteristic Angles	102
5.5.5. Determination of Cutter Diameter	104
5.5.6. Determination of Cutter Length and Number of Inserts	105
5.6. Tool Holder Selection Module	106
5.7. Set-up Planning and Sequencing Module	107
5.7.1. Motivation and Methodology	108
5.7.2. An Example to Set-up Planning	113
6. DESIGN FOR MANUFACTURING	116
6.1. Introduction	116
6.2. Design for Manufacturing /Concurrent Engineering	116
6.3. Elimination of Critical Regions on the Component	117
6.3.1. Methodology of the Developed System for Critical Regions	119
6.3.2. An Example	122
7. OPTIMISATION OF PROCESS PLANNING EVENTS	125
7.1. Introduction	125
7.2. Optimization of Sequence of Machining Operations by Using a Genetic Algorithm	125
7.2.1. The Proposed Approach	126
7.3. An Alternative Approach to the Optimization of Sequence of Operations: RTSS - Right Tool Selection System	137
7.3.1. Methodology	138
7.4. Optimization of Turret Indexing Positions by Using a Genetic Algorithm	141
7.4.1. An Example	146
7.5. The Optimisation Module for Cutting Parameters: CPOS - Cutting Parameters Optimisation System	148
7.5.1. Optimisation Model	150
7.5.2. Derivation of the Objective Functions	150

7.5.3. Constraints Used in the Optimisation	154
7.5.4. Optimisation Methodology	157
7.5.5. Case Study	161
8. CAPP/CAM INTERFACE: CNC CODE GENERATION	165
8.1. Introduction	165
8.2. Computer Aided Manufacturing (CAM)	165
8.2.1. Generation and Verification of Part Programs	166
8.3. Machine Tool Specifications	167
8.4. CNC Code Generation Module	168
8.4.2. Simulation Tool Used for Verification of Part Programs	170
8.5. CNC Code Generator/Simulator for Cavity Machining	171
8.5.1. Background on Pocket Machining	172
8.5.2. Methodology of Pocket Machining System	173
9. APPLICATIONS	182
9.1. Introduction	182
9.2. OPPS-PRI: An Optimised Process Planning System for Prismatic Parts	182
9.3. Execution of the OPPS-PRI	184
9.4. Case Study	191
10. DISCUSSION AND CONCLUSIONS	210
10.1 Introduction	210
10.2 Discussion	210
10.2.1 Initiation of the Work	211
10.2.2 The Need for This Work	212
10.2.3 Specification of the Tools Employed in the Thesis	213
10.3 Contributions and Concluding Remarks	218
10.4 Recommendations for Further Study	221
LIST OF REFERENCES	223
APPENDICES	241
VITA	278

LIST OF TABLES

<i>Table</i>	<i>Label</i>	<i>Page</i>
1.1.	Direct Effects of CIM modules	4
4.1.	A Section of DXF File Generated for Circle Shown in Figure 4.7	61
4.2.	Some of the Data Types in IGES Standard	62
4.3.	The Internal Structure of a Typical CAD*I File	64
4.4.	A Section of B-Rep Data Base of the Example Part	79
4.5.	The Key Words for Determining the Type and Orientation of Faces	79
4.6.	Face Attributes	79
4.7.	Edge Attributes	80
4.8.	Relation Matrix Derived for the Example Part	81
4.9.	The List of Recognised Features and Their Codes	81
4.10.	Features and Their Acronyms Used in This Thesis	82
4.11.	Output From the Feature Recogniser	82
5.1.	A Portion of Raw Stock (Workpiece) Data Base	89
5.2.	The Input Data for the Sample Part	93
5.3.	Priority Values for Each Characteristic Factors and Their Totals	95
5.4.	The Order of Features Based on Priority Values	95
5.5.	Sequence of Operations Derived From Sequence of Features	96
5.6.	Rake Angle Combinations with Hertel Material Groups	104
5.7.	Precedence Attributes of the Example Part in Fig. 5.20.	113
5.8.	Tool Approach Directions of Example Part in Fig. 5.20.	113
5.9.	All Feasible Set-ups for the Example Part in Fig. 5.20.	114
5.10.	Optimal Set-up Plans Obtained in the First Stage for the Example Part	114
5.11.	The Converted Tolerance Table for Example Part in Fig 5.20	115
5.12.	Set-up Sequences Based on Tolerance Factors	115
5.13.	A Converted Tolerance Table II for the Example Part in Fig. 5.20	115
5.14.	Set-up Sequences Based on Tolerance Factors (Second Case)	115
6.1.	Geometric Database of the Example Part	123
6.2.	Minimum Distances (mm) Between Interacting Features for Example Part	123
6.3.	Tool Diameter and Speed for Machining Features on Example Part	123
7.1.	CAD Data Base of the Sample Part Shown in Fig. 7.3	131
7.2.	Reward/Penalty Matrix (REP.MAX) of the Sample Part shown in Fig. 7.3	132
7.3.	The Sequences from TSP Algorithm for Sample Part Shown in Fig. 7.3	132
7.4.	Randomly Generated Initial Population of Sequences (Parents)	133
7.5.	Sequences Derived by the Genetic Operators (Children)	133
7.6.	Reward/Penalty Matrix Prepared for a Highly Difficult Part	135
7.7.	The Sequences from TSP Algorithm Based on Table 7.6	136
7.8.	Priority Matrix	138
7.9.	Tool(s) versus Operations	139
7.10.	Tool versus Operations	139

7.11.	Optimal Block Structure of Sequence of Operations with Their Assigned Tools	140
7.12.	Cutting Tools Assigned to Machining Operations	146
7.13.	Proposed Solutions by GA	147
7.14.	Nomenclature Used in the Formulation of Objective Functions	152
7.15.	The Alternative Cutting Strategies	160
7.16.	A Typical Chromosome for Cutting Parameters Optimisation	161
7.17.	Specifications and Constants	162
7.18.	Machining Operation and Its Parameters	163
7.19.	Tooling Data	163
7.20.	Optimum Machining Parameters Found by the Optimisation Program	163
7.21.	Objective Function Values	163
7.22.	Cutting Strategies for Machining with Their Ratings	163
7.23.	Optimum Machining Parameters for 5mm depth	164
7.24.	Ratings of Parameters Shown in Table 7.23.	164
8.1.	CAD/CAM Data Base for the Example Component given in Fig. 8.1.	168
8.2.	The Part Program Generated for Example Part	170
8.3.	Pocket Boundary - Data Base for the Pocket Shown in Fig. 8.6.	177
8.4.	Island Boundary - Data Base for the Pocket Shown in Fig. 8.6.	177
8.5.	Codes Generated for the Example Pocket shown in Fig. 8.6	180
9.1.	B-Rep Data Base of Example Component Shown in Fig. 9.11.	192
9.2.	Relation Matrix Derived for the Example Component	195
9.3.	The List of Extracted Feature Types and Their Codes	196
9.4.	Output from the Feature Recogniser	197
9.5.	Raw Stock Information	197
9.6.	Optimal Set-up Plans for the Example part	198
9.7.	The Sequence of Operations Determined for the Sample Part	198
9.8.	Modified Sequence of Operations for the Second Set-up	198
9.9.	Cutting Tool(s) versus Operations	199
9.10.	Tooling Information	199
9.11.	Specifications of Cutters	199
9.12.	Reward/Penalty Matrix of the Sample Part (First Set-up Operations)	200
9.13.	Optimum Sequence of Operations in the First Set-up Based on "Safety"	200
9.14.	Optimum Sequence of Operations in the Second Set-up	201
9.15.	Priority Matrix (for the First Set-up)	201
9.16.	Tool(s) versus Operations (the First Set-up)	201
9.17.	Tool versus Operations (the First Set-up)	201
9.18.	Optimal Block Structure of Sequence of Operations with Their Assigned Tools (the First Set-up)	202
9.19.	Tool versus Operations (the Second Set-up)	202
9.20.	Cutting Tools Assigned to Operations (for Both Set-ups)	202
9.21.	Proposed Position Sets	203
9.22.	Machining Operations and Their Requirements	204
9.23.	Cutting Parameters for the Example Part	204
9.24.	Modification of the Critical Distance Between the Features	205
9.25.	Absolute Part Program Generated for the Example Part	206

LIST OF FIGURES

<i>Figure</i>	<i>Label</i>	<i>Page</i>
1.1.	The Elements of Computer Integrated Manufacturing	3
1.2.	Main Keywords in This Thesis	8
3.1.	General structure of an expert system	35
3.2.	A Typical ANN Structure	38
3.3.	An Example of an Artificial Neurone Unit	38
4.1.	Wire-frame Modeling	50
4.2.	Surface Modeling	51
4.3.	Solid Modeling	51
4.4.	The Function of Interfaces in CAD/CAM Integration	59
4.5.	Data Exchange Between CAD/CAM Systems	60
4.6.	Data Exchange Between CAD Systems with Their Pre/Post Processors	60
4.7.	A Circle Drawn Using AutoCAD and its Geometrical Information	61
4.8.	The IGES file of the Circle Shown in Figure 4.7	62
4.9.	The structure of STEP Standard	67
4.10.	Typical Structure of a STEP file	69
4.11.	An Example Part	71
4.12.	Example Part and Its Faces	78
4.13.	The View of Recognised Features	81
5.1.	CAPP Approaches and Implementation Techniques	85
5.2.	Typical Material Flow in Process Planning	89
5.3.	Enveloping Faces, Blank and Workpiece Size for an Example Part	90
5.4.	Corner Stones in a Typical CAD/CAM Integration	91
5.5.	Features and Corresponding Milling Operations	92
5.6.	Sample Part	93
5.7.	Convention for Approach Directions	94
5.8.	Elementary Machined Volumes	94
5.9.	The Three Basic Elements of Milling Processes	97
5.10.	Block diagram of CTSS	99
5.11.	Three Different Orientations of a <i>Step</i> Feature	101
5.12.	Rake Angles and Approach Angle	102
5.13.	Rake Angle Designs	103
5.14.	Combination Arbor	106
5.15.	Arbor with Internal Centering	107
5.16.	Centering Mandrel with Steep Taper Shank	107
5.17.	Interaction Between the Slots	109
5.18.	Relevant Lengths of a Tolerance	110
5.19.	The Maximum Admissible Rotation Within the Tolerance Field	110
5.20.	An Example Part for Set-up Planning	112
6.1.	Thin Sections Between Part Features	118
6.2.	Representative Critical Area Between Two Rectangular Pockets	120
6.3.	Representative Critical Area Between Two Holes	120
6.4.	An Example Part	122
6.5.	Top View of the Example Part	122
7.1.	Flowchart of the GA Based Optimisation System	127

7.2.	Interaction Between a Thru Hole and a Face Pocket	128
7.3.	Isometric View of the Sample Part	130
7.4.	Top View of the Sample Part shown in Fig. 7.3	131
7.5.	Distances Between Features' Datum and the Main Datum	131
7.6.	A Part with Two Features: a step and a side-slot	137
7.7.	Turret Indexing Time (From Tool to Tool), t_i	142
7.8.	A Typical Tools Arrangement on a 12-Station ATC	142
7.9.	ATC Arrangement; # Tools = # Indexes (Phase 1)	144
7.10.	ATC Arrangement; # Tools < # Indexes (Phase 2.a)	144
7.11.	ATC Arrangement; # Tools < # Indexes (Phase 2.b)	145
7.12.	ATC Arrangement; # Tools > # Indexes (Phase 3)	145
7.13.	Schematic Representation of Sectioning Strategy	159
7.14.	A Slot	162
8.1.	An Example Component	168
8.2.	Pocket Machining Methods: a) Stair-casing b) Contouring	172
8.3.	Geometrical Constraint (Throat) and Unmachinable Regions	174
8.4.	Calculation of the Distance of a Point to an Arc Segment	175
8.5.	Positioning of a Cutter	176
8.6.	An Example Pocket with Two Islands	177
8.7.	Minimum Distance Between Two Islands and Boundaries	178
8.8.	Segments of Tool Paths	179
9.1.	The Overall Structure of the OPPS-PRI	183
9.2.	Main Menu of the OPPS-PRI on the Opening Screen	184
9.3.	FILE Pull-down Menu	185
9.4.	CAD-INTERFACE Pull-down Menu	185
9.5.	PLAN Pull-down Menu	186
9.6.	OPTIMISATION Pull-down Menu	186
9.7.	DFM-TOOL Pull-down Menu	187
9.8.	CAM-INTERFACE Pull-down Menu (Code Generation Item Is Selected)	187
9.9.	CAM-INTERFACE Pull-down Menu (Code Transfer Item Is Selected)	188
9.10.	Effect of ABOUT Pull-down Menu on the Screen	188
9.11.	The Example Component for Case Study	191
9.12.	The Example Component with Approximated Holes	195
9.13.	Extracted Features on the Component	196
9.14.	Tool Arrangement on the 16-Station ATC of TAKSAN TMC 700 VMC	203
9.15.	Workpiece Zeros for the Example Part	206
A1.	A Small Section on a Hollow Cylindrical Member	248
A2.	Forces Exerted on the Small Section (Element)	248
A3.	Shearing Stress at Any Point of a Transverse Section	249
A4.	A Small Element of the Wall Section	250
A5.	Triangular Area	250
A6.	Area @	251

NOMENCLATURE

<i>Abbreviation</i>	<i>Meaning</i>
AAG	Attribute Adjacency Graph
AFD	Automatic Fixture Design
AGM	Advanced Geometric Modelling
AI	Artificial Intelligence
ANN	Artificial Neural Network
APT	Automatically Programmed Tools
ASCII	American Standard Coding for Information Interchange
ASV	Alternating Sum of Volumes
ATC	Automatic Tool Changer
B-Rep	Boundary Representation
CAD	Computer Aided Design
CAD*I	Computer Aided Design*Interface
CAD/D	Computer Aided Design / Drafting
CAI	Computer Aided Inspection
CAM	Computer Aided Manufacturing
CAPP	Computer Aided Process Planning
CAQQ	Computer Aided Quality Control
CAT	Computer Aided Testing
CBN	Cubic Boron Nitride
CE	Concurrent Engineering
CIM	Computer Integrated Manufacturing
CNC	Computer Numerical Control
CPOS	Cutting Parameters Optimisation System
CSG	Constructive Solid Geometry
CTSS	Cutting Tool Selection System
DFM	Design For Manufacturing
DIS	Draft International Standard
DNC	Direct Numerical Control
DXF	Data eXchange File
EDM	Electro Discharge Machine
EIA	Electronic Industries Association
EOB	End Of Block
ES	Expert System
FBD	Feature Based Design
FEM	Finite Element Method
FIXES	A System for Automatic Generation of Setups and Design of Fixture for Prismatic Parts
FL	Fuzzy Logic
FMS	Flexible Manufacturing System
FRS-PP	Feature Recognition System for Prismatic Parts
GA	Genetic Algorithm
HMC	Horizontal Machining Center
HSM	High Speed Machining
HSS	High Speed Steel
IGES	Initial Graphics Exchange Specification
IMS	Intelligent Manufacturing System
IPIM	Integrated Product Information Model

IS	International Standard
ISO	International Standardisation Organisation
LISP	LISt Processing
LMS	Least Mean Square
LOX	Linear Order Crossover
MCU	Machine Controller Unit
MDI	Manual Data Input
MEB	Minimum Enclosing Box
MTC	Minimum Tool Change
MTTD	Minimum Tool Travelling Distance
NC	Numerical Control
OPPS-PRI	Optimised Process Planning System for Prismatic Parts
OPPS-ROT	Optimised Process Planning System for Rotational Parts
PC	Personal Computer
PDDI	Product Definition Data Interface
PDES	Product Data Exchange Specification
PMX	Partially Matched Crossover
POB	Principal Orientation Box
PP&C	Production Planning and Control
PROLOG	PROgramming in LOGic
REPMAX	Reward & Penalty Matrix
RM	Right Movement
RTSS	Right Tool Selection System
SET	Standard d'echange Et de Transfert
SME	Society of Manufacturing Engineers
STEP	STandard for the Exchange of Product model data
TF	Tolerance Factor
TRP	Tool Reference Point
TSP	Travelling Sales Person
UM	Up Movement
VDAFS	Verband Der Automobilindustrie FlaechenSchnittstelle)
VDU	Visual Display Unit
VMC	Vertical Milling Center
WSN	Wirth Syntax Notation

CHAPTER I

SCOPE OF THE WORK AND OBJECTIVES

1.1 INTRODUCTION

This chapter gives a brief introduction to the context of the research and points out the importance of process planning for integrated design and manufacturing systems. It also presents the objectives of the research. The main reasons behind the selection of prismatic parts (milling operations) for the application area of the thesis and other issues associated to frame of the work are described.

1.2 IMPORTANCE OF PROCESS PLANNING IN CIM AND IMS

Recent researches and developments in manufacturing area have the objectives of increased productivity and cost effectiveness by integrating many activities in a global *CIM (Computer Integrated Manufacturing)* system which is heading towards the *Factory of the Future*. For this purpose, an advanced production system called *IMS (Intelligent, or sometimes Integrated, Manufacturing System)* is oriented towards the needs of the 21st Century and designed to maintain and improve the vitality of manufacturing sector keeping it as the cornerstone of all economic activities and ensuring that manufacturing remains an attractive industrial area. IMS takes intellectual activities in manufacturing and improves productivity by systematising, optimising and flexibly integrating those corporate activities [1]. The need for this

orientation has emerged due to the market requirements, forcing the manufacturing industries to adapt to the rapid developments in science and technology, and to transform the results of the research into marketable new products. Only those industries capable of high market adaptability, competitive planning, productivity and quality with quick responses to these requirements would withstand international competition.

The acronym CIM for the expression Computer Integrated Manufacturing was first used by Harrington [2] to mean *the integration of business, engineering, manufacturing and management information that spans company functions from marketing to product distribution*. However, this is the wide meaning of CIM. Many companies concentrated on the term “integrated manufacturing” as the main priority to be achieved, and they accepted others as the elements to be linked to this “integrated manufacturing”. Therefore, the manufacturing and computers are the two key elements constituting the core of the CIM. Computers have had a substantial impact on almost all activities of a factory since the 1980s. CIM can be considered as the philosophy in which the computer plays a central role for planning and controlling the manufacturing process. The operation of a CIM system in a factory may provide invaluable benefits compared with the conventional systems. Typical benefits are [3];

- Increase of productivity by 40-70%
- Reduction of design costs by 15-30%
- Reduction of the in-shop time by 30-60%
- Better product quality
- Reduction of scrap by 20-50%
- Improved product design

The most important elements of CIM are illustrated in Figure 1.1. One of the greatest challenges of CIM is the integration of CAD (Computer Aided Design), CAPP (Computer Aided Process Planning), PP&C (Production Planning and Control), and CAM (Computer Aided Manufacturing) [3].

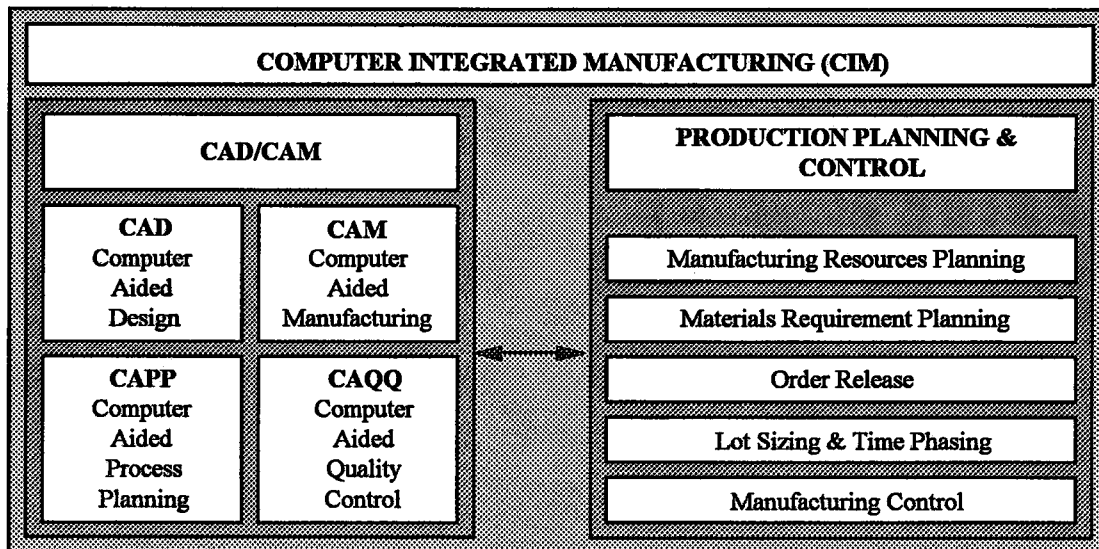


Figure 1.1. The Elements of Computer Integrated Manufacturing [3]

CAD and CAM are two of the most important activities to be amalgamated in a typical CIM environment. Due to developments in computer technology, numerically controlled equipment, robots and computer controlled automation in CAD and CAM systems, many manual skills have been automated, thus resulting in reduction of lead times, improvements in production and increase in manufacturing accuracy. However, the full integration of CAD and CAM systems in industry has not yet been achieved. One of the most important obstacles to be overcome is the automation of process planning activities that bridges CAD to CAM: CAPP [4]. CAPP is very inclusive and basically consists of the determination of processes and parameters required to convert a blank into a finished product [5,6]. The key elements of process planning include; material selection, operation selection, machine tool selection, operation sequencing, cutting tool selection, set-up selection, etc. Automated process planning brings about following advantages [7,8];

- reduction in process planning time
- reduction in the required skill of the process planner
- reduction in costs due to efficient use of resources
- increased productivity and process rationalisation
- production of accurate and consistent process plans
- maximisation of Just-In-Time performance

Two approaches have been used to automate the process planning; variant and generative approach. The variant approach uses the classification and coding of parts to initiate the process planning activity. When a plan is to be generated for a new component, a standard plan for a similar product is retrieved and modified for the new component. The variant approach is useful in cases where there is a lot of similarities between the components. The generative approach generates a new plan for each component without retrieving the stored plans as in the case of variant approach. This approach is more difficult to implement, because a large amount of process planning knowledge is handled. However, they are more flexible and can provide automation with little manual intervention. A typical generative process planning system including its interfaces has at least the following modules [5];

- CAD/CAPP interface: feature recognition
- Determination and sequencing of machining operations
- Determination or selection of cutting parameters
- Selection of cutting tools and auxiliary tooling
- CAPP/CAM interface: CNC code generation.

As shown in Table 1.1 in which a comparison of direct and indirect benefits of CIM modules are given, CAPP has the greatest direct effect (90%) among all the modules. Therefore, an effective implementation of CAPP in a CIM system has the primary importance. However, this may only be justifiable, when it interacts with the other modules of the system.

Table 1.1 Direct Effects of CIM modules [3].

CIM Module	Effects %
CAPP	90
Industrial Robots	90
Automated Assembly	85
Automatically Guided Vehicles	77
NC/CNC/DNC Systems	75
CAD	70
Office Automation	70
FMS	69
CAQQ	66
Warehouse Automation	65
Production Planning and Control	60
Factory Data Acquisition	40

One of the problem with the design of CIM modules (especially with process planning) is the software. There are many programming and system design languages being used or that are still under development. The impact of Artificial Intelligence (AI) tools (like Expert Systems, Neural Networks, Genetic Algorithms, Fuzzy Logic) on the planning of manufacturing processes has been proven by recent research projects and actual implementations. There are numerous packages being developed for almost any manufacturing activity. At the present time, however, it is not known how many of these systems are practicable and are being used. For example, most of the software for operations planning has led to the development of large and complex application packages which are too rigid when manufacturing variables are included. A conservative estimate is that only 5% of all research endeavours have found their place in the factory. This may be a very discouraging reality; but there are numerous reasons for this problem. The tools for building intelligent systems are not sufficiently developed and are difficult to apply. The methods for acquiring knowledge from experts to develop expert systems are not very well understood. They usually contain several thousand rules and are huge software systems which are difficult to use on conventional computer systems. There are only a few qualified people available who really know how to apply AI tools. These problems can be solved by using or developing right tools, methods and environments. However, the potential and power of AI is very great and it is believed that with the exploitation of AI methods, it will only be possible to build well conceived and intelligent CIM systems in which many routine jobs (which may become very repetitive and boring, after the skill has been acquired) are taken out of the responsibility of the experienced manager so that his creativity can be devoted to solving more complex problems in factory. In this thesis, an attempt has been made on the use of AI techniques for the enhancement of performance and capability of process planning facilities.

1.3. FRAME OF THE WORK

Milling is one of the most common metal removing processes in manufacturing. With the introduction of High Speed Machining (HSM) and

improvement on the milling equipment and tools that can produce superior milled components, the application of milling process in manufacturing is increasing.

Milling is becoming an increasingly universal machining method with growing variations in machinery, controls and cutting tools. Machining centres are today large users of milling and required a number of different type of tools. Milling is performed on different types of machines, from those with basic designs from the turn of the century to those with sophisticated multi-axis CNC. Milling has great variation in not only the types of operations that can be performed but also the machines, tools and workpieces. The advantages of milling are high machining efficiency, good surface finish and accuracy and flexibility in generating forms. Most frequently, milling entails the generation of flat faces, shoulders and slots but efficiency in contour-milling is growing with the use of CNC [9].

A recent survey of die/mould industry (in Japan, Germany, and USA) which makes extensive use of milling processes illustrated the state of the technology in the die/mould making business [10]. It has been reported that about 5% of the total lead time is spent in the process planning step. With the introduction of high speed machining (HSM), application of the milling process is continuously expanding. The surveyed companies were asked about the milling equipment they use. Although Vertical Machining Centers (VMCs) have disadvantages concerning chip disposal, they are less expensive and, therefore, more popular than Horizontal Machining Centers (HMCs). Observations made in an earlier survey had indicated that copy milling was the most popular machining method [11]. The last survey revealed that 3-axis CNC machining is more common now. 4-axis CNC milling offers the possibility to tilt the milling cutter to improve the cutting conditions. It is currently in use in almost 35% of the German companies. 5-axis CNC machining exceptionally offers advantages when machining convex surfaces. It is still not widely accepted in Japan and USA. Tough brazed cutting inserts offer a higher accuracy during machining, they need regrinding when they are worn. Thus, indexable tools are preferred over tools with brazed inserts. In regard to cutting tools, the most important improvement that the die/mould producers expect is a longer tool life. The majority of the

die/mould shops are unsatisfied with the life they obtain from presently available cutting tool materials.

The above discussion is more or less true for global Turkish industry and local Gaziantep industry, although they are not well-developed as compared with those highly industrialised countries. The use of new technology and CNC machine tools has increased in Turkey, while “CAD/CAM” is still seen as an unattainable meta. Small and medium sized companies recently have become an important part of Turkey’s manufacturing industry. They are increasing in number and would probably play significant roles in the acceptance of Turkey to the European Union. *Productivity* of this industry has a vital importance and should be increased to provide the qualitative success as well as the quantitative success, and competitiveness. *Process planning* is the key of production and increasing productivity. Small to medium sized machining industry is very sensitive about the diffusion of process planning know-how. Therefore, the research on the automation of *process planning* linking CAD to CAM is invaluable.

In the light of state of the art of the machining industry described above, in this work, milling is selected as the application area for process planning system with its two interfaces to CAD and CAM. In all modules of the developed system, the latest developments on CAD/CAM technology are considered. For example, in tool selection module, tools are selected from an indexable tools library. The selected machine tool was a CNC VMC. This is due to availability of a TAKSAN TMC 700 VMC in the department, popularity of VMC among machining centers and wide variety of drilling operations that can be performed on a VMC. However, the system developed in this work can easily be modified so that it can produce part programmes for HMCs with small changes.

New products have been generally designed to be produced on 3-axis CNC machining centers from cubical billets [12]. The current work focuses on the manufacture of prismatic parts using a 3-axis machining center. *Prismatic parts* can be defined as the components which have planar and cylindrical faces as orthogonal sets such that all features are a product of a 2-D profile swept through a linear

distance along an axis normal to their plane. They cannot have sculptured or convex surfaces requiring 5-axis machining. The system handles a wide spectrum of finished part geometry with little modifications.

The frame of the work can be best described by following figure and keywords;

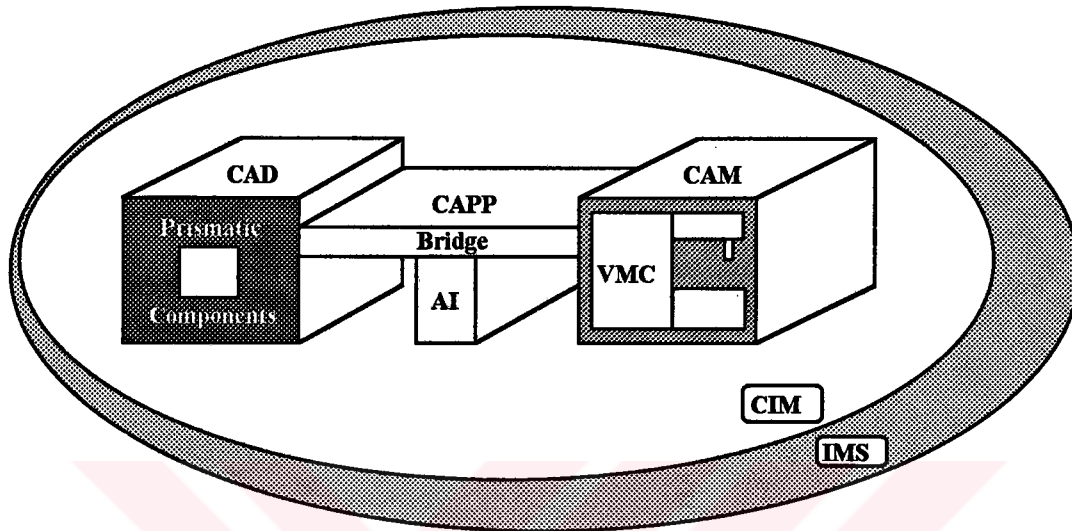


Figure 1.2. Main Keywords in This Thesis

- CAPP system bridging CAD and CAM with respective interfaces
- CNC 3-axis machining
- Prismatic Parts

1.4. OBJECTIVES OF THE WORK

The main objective of this work is to make a contribution to the research related to integrated CAD and CAM activities for prismatic parts. The present work concentrates on the following points;

- trying to systematise, optimise and integrate the each corporate activities in CAD/CAM integration in order to improve productivity for the realisation of CIM and IMS

- developing a complete CAD/CAM package (software) as a means of integrating elements between design and manufacturing activities via CAPP
- integrating design activities with process planning by a STEP standard based feature recogniser with the use of features
- allowing automatic generation of CNC codes by integrating process planning activities with manufacturing tool with a CAPP/CAM interface
- investigating the use of AI techniques in process planning activities in order to accelerate the performance and intelligence of CAD/CAM integration
- providing a PC based process planning aid for prismatic parts which can be used on the shop floor of the small/medium sized manufacturing industry in Gaziantep
- implementing the developed software on a CNC machine tool
- presenting an expert CAD/CAM system which may physically not be an expert system, but it can be replaced with an expert human process planner
- including a 'Design for Manufacturing' utility to enhance the degree of integration between CAD and CAM

1.5. METHODOLOGY AND MATERIALS

In this study, a Computer Aided Process Planning (CAPP) package for prismatic parts is developed together with its two interfaces (CAD/CAPP interface and CAPP/CAM interface) to achieve the CAD/CAM integration. The developed package is mainly composed of the modules for the following functions;

- CAD facilities
- interfacing CAD to CAPP: feature recognition based on STEP standard
- determination and elimination of critical regions on the component
- selection of an workpiece size
- determination of machining operations
- set-up planning
- operations sequencing and/or optimisation of sequence of operations
- optimisation of cutting parameters (conditions)

- selection of cutting tools and auxiliary tooling
- optimisation of turret index positions
- interfacing CAPP to CAM: CNC code generation and verification including automatic pocket machining

These modules has been grouped into five main sections;

- Section 1: consists of the modelling platform (CAD tool) and feature recogniser
- Section 2: includes the process planning works
- Section 3: involves optimisation of process planning events
- Section 4: offers a utility on “Design for Manufacturing”
- Section 5: is devoted to the CAPP/CAM interface

A typical session performed on the developed package is as follows. After the component is modelled on a CAD platform and the STEP data of the component is obtained, the machining features on the component are recognised. An optimum workpiece size is going to be selected from standard workpiece database. Type of machining operations for each feature of the part is determined correspondingly. Then, machining operations are sequenced based on the rules constructed in the system, while considering set-ups. The cutting tools as well as the other auxiliary tooling are selected from respective tool libraries. The machining parameters are then determined. Features on the component can be checked for the critical regions in between whether they are machinable or not, under the determined machining conditions. The most usable result of the system is a part program executable on a VMC which is generated with the use of the CAD/CAM database prepared by up-stream modules of the system.

The developed package has a modular structure. For each of the module specified above, a program has been written by C programming language on a Pentium 133 PC. The modules have been combined under a supervisory program. The package can work either as a stand-alone system or be integrated with other process planning systems developed for both prismatic and rotational parts in a CIM environment.

The most outstanding characteristics of the developed package can be summarised by the following key words or sentences;

- flexibility
- modular structure
- generative approach
- the use of features
- STEP based feature recognition
- handled multi set-ups
- enhanced with AI techniques
- implemented “Design for Manufacturing” tool
- designed for prismatic parts
- optimised process plans

1.6. ORGANISATION OF THE THESIS

Following this chapter which introduces the objectives, frame and context of this research, the first three chapters present an overview of the main theories and subjects related to this work.

- Chapter 2 is devoted to related research work review. Current status of CAD/CAM integration is highlighted and conclusions from the literature survey are drawn.
- Chapter 3 provides a discussion on the essence of Artificial Intelligence and review of Artificial Intelligence methods.

From Chapter 4 onwards, the modules forming the CAD/CAM integration system developed in this thesis are discussed together with their methodologies.

- Chapter 4 is the first of four application chapters and gives a general review of STEP standard and presents the product modelling scheme as well as modelling

environment used in the development of CAD stage of the work. The Feature Recognition system, FRS-PP is also introduced.

- Chapter 5 concentrates on the modules which form the developed process planning system presented in this thesis.
- In Chapter 6, the 'Design for Manufacturing' tool implemented in the developed CAD/CAM integration system are presented.
- Chapter 7 involves the optimisation procedures of process planning events.
- Chapter 8 illustrates the main components of CAM interface of the developed process planning system and manufacturing environment.
- Chapter 9 is devoted to demonstrations of the functions of all modules of OPPS-PRI through an example.
- Chapter 10 discusses the main issues arising from this research work, capability of the developed CAPP package and the results obtained from this study. The recommendations for further work are also presented.



CHAPTER II

LITERATURE SURVEY

2.1. INTRODUCTION

A survey of the most relevant literature related to the study reported in the thesis is presented in this chapter. Since process planning has a divergent nature, it embraces many sub-topics like *workpiece size selection*, *operation sequencing*, *tool selection*, *holder selection*, *set-up planning* and so on. Apart from these, there are two main interfaces to process planning; *feature recognition* and *CNC code generation*. Moreover, *Design For Manufacturing* utilities and optimisation of process planning events increase the divergence of the system. Literature on each of these activities is investigated one by one in the following sections. The CAD/CAM integration efforts in the literature are also investigated as a whole phenomenon. The current status of the CAD/CAM, problems and conclusions drawn from extensive literature survey are given at the end of the chapter.

2.2. LITERATURE ON “FEATURE RECOGNITION”

2.2.1. Historical Background

The earliest work related to features goes back to Grayer’s Ph.D. dissertation at Cambridge in 1976 [13]. The objective of Grayer’s work was to automate the link

between CAD and NC part programming. Several other researchers in the CAD group at Cambridge contributed to solid modeling and feature modeling. Some well known solid modeling systems; namely *Romulus*, *Parasolid* and *Acis2* are the examples of contributions of the CAD group. Kyprianou's Ph.D. work established the foundations for feature recognition from boundary models. Feature recognition research then appears to have spread to several universities in the U.S.A, particularly to Purdue and Illinois universities. In the mid-1980s CAM-I proposed the *design by feature* approach and several prototype systems began appearing. By the late 1980s commercial CAD systems started supporting parametric and feature based approaches. Throughout this period there was also considerable research in automating process planning and NC part programming [14].

2.2.2. Work on the Feature Recognition

There have been considerable researches on the feature recognition systems. First part description method on the basis of the fundamental concept of converting the part drawing information into computer-oriented information for data structure was developed by Kakino [15]. The method of *syntactic pattern recognition*, in which a set of basic primitives and a part representation technique are used, has been generally used for two dimensional patterns. Many researchers used a syntactic pattern to describe 2-D profile information on machined parts [16-21]. In *state transition diagram approach*, a component is described using the sweep operators, and union of the swept volumes. The part geometry is decomposed into several volumes (like manufacturing features) for manufacturing considerations in *volume decomposition approach*. Woo [22] developed an algorithm that works on the decomposition of 3D objects depends upon the designer's sequences of volume operations. Woo [23] developed also a novel technique known as "*Alternating Sum of Volumes*" (ASV) expressions, for extracting features from the boundary representation of 2D or 3D parts. The fundamental importance of the ASV procedure is that it enables the conversion from B-rep to a CSG-like tree structure. Since B-Rep representations of solid are more favourable in extracting the features, a few research has been made into *set theoretic approach* in which primitives (design features) are used in creating the component. In *graph-based approach*, boundary representation of the part is

converted into a graph which involves a set of nodes and their attributes. Joshi and Chang [24] developed a graph named the *Attribute Adjacency Graph (AAG)* to represent features in which each face of the part is represented as a node, and each edge or face adjacency is represented as an arc. Sakurai and Gossard [25] have used feature graphs by identifying faces from a 3D B-rep model. de Floriani [26] converted B-rep information of the part to a face-oriented relational model called *generalised edge face graph*. Gindy, et.al. [27], have used *external excess directions* technique which is one of the more prominent methods for feature classification and characterisation.

Apart from above approaches, many researches have used *expert system techniques* in which features are recognised using human logic and based on the rules constructed in the form of IF <conditions> THEN <actions>. Henderson and Musti [28] utilised expert system approach to extract knowledge useful to manufacturing in the form of part features from 3D CAD data. Henderson and Anderson [29] used the rules that were constructed based on the geometry and topology of the features.

Neural-net based techniques having the ability to learn and dynamically improve their behaviour are one of the most promising method for feature recognition. They have been applied to the extraction of features since 1990. The systems developed by Prabhakar and Henderson [30] and Chen and LeClair [31] are the two good examples of neural network based feature recognition.

The automatic feature recognition is actually a difficult task, since it essentially depends on the kind of representation technique used in solid modeler with which the part is designed. Moreover, the algorithms written for feature recognition is very complex and depend upon the specific solid modelers, and they can only recognise the classes of features which are previously described [32]. It is also well worth pointing out that there is a few research on the recognition of features for prismatic parts *having a wide spectrum of features* and capable of recognising highly complex features in the literature as compared to those performed for rotational ones. This is mainly due to difficulty of representation of a generic object on a solid modeler, and due to increased complexity of features on the prismatic components. However, the

application of many systems developed for the feature recognition of prismatic parts has not been considered thoroughly enough to verify its suitability for process planning [33]. On the applications side, “*features*” have not been fully exploited, as yet. This is notably true in the case of manufacturing features for prismatic parts.

The participants of a features symposium held in U.S.A in August 1990, under CAM-I’s sponsorship identified the following “obstacles” to widespread use of features and the development of better feature recognition systems [34];

- non-availability of a standard feature description language,
- lack of techniques for transforming design features to manufacturing features,
- no standardised communications between feature modelers and geometric modelers,
- inflexibility in feature definitions leading to many specialised features.

The frontiers of “features technology” were identified by the researchers participating the *Features Workshop* held in UK in August 1991 at Bristol University as follows [35];

- Unification of *feature recognition* and *design by features* has not been achieved,
- Feature interactions make recognition algorithms unreliable,
- Feature mapping does not utilise much feature level information relying instead on just the geometric model,
- Current support for feature data exchange via STEP is at a very low level, since relational information is not fully supported. It is the feature relations that are most needed in manufacturing applications. Thus, there is a strong need for a geometric constraint model,
- The high computational cost of feature based modeling was considered to be a problem.

In spite of all of these problems, significant progress has been made in the feature technology over the years and feature technology is accepted as the main key to CAD/CAM integration. “*Feature Recognition*” as well as Rapid Product Modeling

was also announced as a possible future project topics in Intelligent Manufacturing System (IMS) area, in the 1st Symposium on IMSs which was held in Sapanca, in 1996, at the Sakarya University, Turkey. It is worth here noting that the works on the feature technology to automate the CAD/CAM integration have been shifted recently on a mixed methodology called *hybrid method* which includes *Feature Based Design* (FBD) in conjunction with the *feature recognition* and design by features which are extensively discussed in the following chapters.

2.3. LITERATURE ON “WORKPIECE SIZE SELECTION”

The literature on the workpiece size selection is not so rich. Some example work which are commonly an integral part of a process planning system are reported in the following paragraphs.

Mayer [36] has described an idea by which the raw stock is chosen for 3D polyhedral parts. The size of the raw stock is determined by the *Minimum Enclosing Box (MEB)* that is the smallest rectangular enclosing box based on the component convex hull. Such a box has at least one face that lies on the same plane as one of the convex hull faces, and the smallest possible volume.

Yue et.al [37] has used an enclosing object paradigm as a mechanism for performing raw stock determination. The generation of convex hulls and enclosing boxes are a central part of this mechanism. A number of enclosing box called *Principal Orientation Box (POB)* has been created from the convex hull of the silhouette such that each POB has at least one side co-linear with one of that of the convex hull. POB can also be defined as the minimum enclosing box whose principal moment is parallel to that of the part. After the possible POBs of a part are found, the smallest possible workpiece is selected based on the created POBs from the raw stocks available in the data base.

EPPSU which has been developed recently for prismatic parts has provided a “*stock selection and blank part modeling*” module integrated into the process

planning system [38]. In that work, maximum and minimum dimensions of the component is found. A machining allowance is added to each face to determine the blank size. The stock for the blank is then selected by searching the raw stock data base. The amount of minimum allowance in this work is 2.5 mm. However, this value can be increased, if the blank dimensions can not be found in the data base. Machining allowances are increased accordingly to match the raw stock size dimensions in the data base.

Champati, et.al. [39] et.al. have used the “*raw stock representation*” as a deterministic criterion in their automated operation sequencing system. They proposed that a raw stock has six faces which can be classified according to the mechanical processes revealed. For example, a face can be saw cut or cold rolled.

2.4. LITERATURE ON “OPERATION SEQUENCING”

There have been considerable researches on *operation sequencing*, since it is one of the modules of process planning and more than two hundred prototype process planning system have been reported in the last two decades. Each system has its own method to determine the operations and their sequence. *Decision Tree Method* [40,41] and *Sequential Operation Planning* (SOP) method [42] are the two widely used approaches for the determination of sequence of machining operations. OPPS-ROT [43] has used the SOP method based on the assumption that a single feature on a component is usually considered as a single operation. The sequence of operations is determined by using a technique developed in this work called RIL-Z method which seemed to be more advantageous over some of the studies which makes use of form primitives. ICAPP system which is one of the earlier work for non-rotational prismatic parts determined the machining operations necessary to produce a feature on the prismatic components by using the information on type of feature together with its dimensions and tolerances [44]. Joshi, et. al. [45], have developed an expert process planning system for prismatic parts in which the sequence of operations are determined by establishing a precedence among the set of features comprising the part. Kusiak [33] has used decomposition and sequencing of machinable volumes to

find the sequence of operations by using the precedence constraints generated within the system. The operation sequencing module of a more recent expert process planning system called EPPSU have sequenced operations based on ordering of features for 3D prismatic parts [46].

Determination of an *optimal sequence* for machining operations which depends basically on feature sequencing (so indirectly on the operation sequencing) is one of the main activities in process planning systems to satisfy the cost requirements. The function of the optimisation of sequence of operations is defined as the arrangement of the processes to produce a component in an optimal order by applying a standard set of rules [47]. There have been extensive works on optimal operation sequencing, since the performance of a process planning system mainly depends on the efficiency of operation sequencing. Ssemakula's pioneering work is one of the first systematic approaches to optimal process sequencing [48]. Rho, et.al. and Kayacan et. al. have analysed the technical and economical aspects of operation sequencing and implemented a heuristic matrix method to obtain optimum sequence of operations [43,49]. Machining operations have been sequenced in an effort to minimise the total machining and tooling cost by Lin and Wang [50] and Koulomas [51].

Recent researches have been performed in optimal process sequencing by using artificial intelligence methods like expert systems and genetic algorithms. Vosniakos and Davis developed a knowledge-based system in which hole-making operations are selected and sequenced producing a single plan with minimum tool changes [52]. Kamhawi et.al., have proposed an approach based on a genetic algorithm for feature sequencing in a rapid design system [53]. The proposed system provides a machinist with safe and near-optimal feature sequence while considering "*safety*" and "*tool change*" as conflicting criteria.

2.5. LITERATURE ON “TOOLING”

Selection of cutting tools and holders is another important module of a generative CAPP system. As indicated in Reference [43], there have been a lot of efforts to automate the tool selection facility. Maropoulos and Hinduja [54] prepared a program to select cutting tools for rough turning operations performed on a CNC machining centre. Tools are selected by using the cutting data for first and last passes in a multi-pass situation. Hinduja and Barrow developed [55], a semi-intelligent tool selection system for turning components. The system is user friendly and highly interactive. It is capable of dealing with all the operations normally associated with the manufacturing of turning components. The automated tool selection by the *production rule matrix method* was developed by Domazet [56] for external rough turning operations. The system is designed by a production rule matrix method using a table formed rule base to replace “If <antecedent.> Then <consequent>” type of rules. The system is heavily dependent on information provided by the manufacturer.

Although extensive works have been reported on the *cutting tool selection* in the literature such as the works reported in [56,57], no one has considered a parallel approach in which optimisation procedures were directly amalgamated with the tool selection and operation sequencing, while some of them have attempted to optimise the cutting conditions after selecting a tool for each operation. Moreover, only a few of them have considered the optimisation of sequence of operations in relation with the cutting tools. Another conclusion which can be drawn from the literature survey is that although milling is one of the most common metal removal processes, there have not been so many studies on the tool selection for prismatic components as compared with those for the rotational parts. This is mainly due to problems in geometrical representation of the prismatic parts and wide spectrum of candidate cutting tools as with the lack of standardisation of tool holders. It has been also recognised from the literature that the developed systems on the selection of cutting tools are not able to suggest alternative tools for the part features (so, machining operations) which is one of the basic requirements for suggesting *alternative process plans*.

2.6. LITERATURE ON “SET-UP PLANNING”

Many attempts have been done to automate the set-up planning over the last two decades. Most of them were considered as an integral part of the CAPP systems. One important conclusion that can be drawn from the literature survey is that the existing systems were generally based on single set-up, and set-up sequencing and its optimisation was rarely addressed. Those systems have generally limited their domain to a vertical machining center. Clustering methods with constraints, precedence relationships, and heuristics were widely used for implementing set-up planning. Below are the some other promising works referred to the literature on the set-up planning, clamping and fixturing.

Hayes developed a system in which feature interactions were used in constraining the sequence of set-ups [58]. Chang suggested clustering of features based on commonality of approach directions [59]. Sakurai provided a heuristics based system to group features into set-ups [60]. The works done on set-up planning by Zhang et.al. [61] and, Boerma and Kals [62] might be considered as the most attracting approaches. The performances of their systems are quite promising. Zhang et.al. evaluated various constraints on set-up planning of prismatic components. In their study, rules and heuristics have been created based on those constraints to determine precedence relationships between the features on the part. An optimisation approach was developed to find the optimal plan(s) from all feasible set-up candidates [61]. Boerma and Kals developed a computer aided planning system called FIXES for the selection of set-ups and the design of fixtures in part manufacturing [62]. The automatic selection of set-ups was based on the comparison of tolerances of the relations between different shape elements of a component. A tolerance factor has been developed to be able to compare the different type of tolerances.

Wiendahl and Fu [63] have developed a method which enables to analyse the set-up process and derive measures for set-up time reduction upon the basis of ascertained weakness. The necessity of the planning of the set-up process in analogy to manufacturing process results from the existing problems in the set-up organisation. A prototype of such planning system has been developed to equip the set-up

personnel with aids, which make a structured and disturbance free set-up process possible.

Delbressine, et.al. [64] have developed a design methodology which takes manufacturing restrictions into account in the geometric design phase. This has been accomplished by defining *manufacturable design transformations*. A manufacturable design transformation is a design operation which has one or more manufacturing operations as counterpart. The applied design operations are recorded in a *design tree* which is an integral part of the product model. Process planning converts this *design tree*, given the available machines, fixturing tools and machining tools, into a *manufacturing tree*. The *manufacturing tree* consists of one or more set-ups with per set-up a collection of basic manufacturing operations. Their study deals with the automatic generation of the set-ups given a design representation based on manufacturable design transformations. The main advantage of this approach is a faster generation of set-ups. Furthermore, the generated set-ups and the collection of basic manufacturing operations per set-up can be used for automatic generation of NC code.

Murray and Yue [65] have prepared a study on the validation, workpiece selection and clamping of complex 2.5D components. The work in their study is based on the ability to extract geometrical data and features from solid models of the components. The work describes algorithms which validate solid models as 2.5 D components and select suitable workpiece dimensions. Vice, machine clamping and frame bolting workpiece holding methods have been examined. Algorithms capable of determining the best clamping approach have been presented.

A method for selecting and positioning locating points in fixture design has been proposed by Manessa and De Viers [66]. The types of fixture applications considered are for prismatic workpieces that use the *3-2-1 locating principle* as the general structure of the fixture. The design synthesis used six rules based on machining practise to select the secondary and tertiary locating datum for a workpiece stored in a boundary representation solid modeler. Once these edges are selected,

kinematic rules are used to determine the position of the locating points so as to allow accessibility when loading and unloading a workpiece.

Roy and Sun [67] have focused on the development of heuristic algorithms for selecting the locating and clamping positions on an automatic fixture configuration for a given workpiece for an *Automatic Fixture Design* (AFD) system. The motivation of the work is to develop a fixturing subsystem as an integral part of the complete CIM environment. A prototype AFD system has been completed in the object oriented programming platform of *Wisdom's Concept Modeler*. It has used several geometric reasoning mechanisms based on the traditional fixture design principles, and on the principles of collision-free assembly. The system exploits the power of *Wisdom's Advanced Geometric Modelling* (AGM), and it implements the basic architecture of the proposed knowledge based fixture design system.

Young and Bell [68] have developed a method of integrating fixturing strategies with technological and geometric information, within a product modelling environment, to automate set-up planning for machining. The method used machine capability representations and product model analysis techniques to generate set-up plans. The software implementation concentrates on the machining of 2.5D prismatic parts from solid stocks on a 3-axis vertical machining center.

2.7. LITERATURE ON “CUTTING CONDITIONS”

Determination of optimum cutting conditions (parameters) is also one of the most critical stages of process planning strategies. There have been a few attempts to determine the optimum cutting conditions in the literature. It is worth noticing that the literature in the area of the optimisation of multipass machining operations has been quite limited. It has been also recognised that the progress in developing constrained optimisation systems for milling operations has been even slower than for turning operations, since the milling has a more complex cutting mechanism than that of turning. Today, there are only a few works on the optimisation of multipass milling operations cited in the literature. There are generally two approaches to solve the

problem; computer aided mathematical programming techniques and numerical search techniques. In the following paragraphs, the evolution of the works on the optimisation of cutting parameters found in the literature are briefly summarised.

Kals et.al. [69] have first studied the optimisation of external turning operations by considering constraints such as power of machine tool, work holding and workpiece rigidity. Their technique, in a multi-pass operation, resulted in passes with the same depth of cut. Kals, et.al. [69], Crookall and Venkataramani [70] and Lambert and Walvekar [71] have all analysed the optimisation of multipass turning operations. However, they have all selected the number of passes arbitrarily, thus their results could not be considered as the optimal values. Hitomi [72-74] has optimised the optimum cutting speeds and the cycle time without considering the other cutting parameters in his first three attempt. In his later publication, an approximation was applied in order to find the optimum values of the cutting speed and feed [75]. Rao and Hati [76] as well as Iwata, et.al., [77] has also developed an optimisation model for the turning based on either minimum production time or minimum production cost.

Somlo and Nagy [78] stated that geometric programming method is more powerful than other methods in determining the optimum cutting conditions when the solution is restricted by one or two constraints. But they also pointed out that as the number of constraints increases, other optimisation methods should be employed in conjunction with the geometric programming.

Approach introduced by Groover was the *Monte-Carlo simulation technique*. This technique is mainly based on a mathematical model which is described by some assumed probability distribution. The actual machining process is replaced by its mathematical model. The variables in the mathematical model are then sampled by means of a random number generator [79].

El Hakim et.al. [80] prepared an algorithm which makes feasibility checks for the constraints. In this algorithm, the depth of cut is equated to the maximum depth of cut value for the tool and checked for the constraints if it violates any one of them,

depth of cut is then reduced in steps until it satisfies the constraints. Cutting speed calculated by using the selected criteria is then checked for the constraints. The procedure continues until total depth of cut is removed.

Brewer [81] attempted to optimise the speed and feed for minimum cost per component produced considering the maximum available power. He presented graphs of feed versus speed with minimum cost loci to select the optimum feed and speed. He also included curves of constant power so that it is possible to check whether there is sufficient power available to realise the optimum feed and speed.

Van Houten [82] further improved the treatments of constraints. To predict the deflection of the workpiece, he used equations and considered limitations due to chucking and this enabled him to take into consideration axial and rotational slip. For torque, he used power/speed characteristics of a DC servo-motor, which gives the actual power available within the speed range.

Devor, et.al., [83] and Ermer and Kromodihardjo [84] have investigated the optimisation of a two-pass operation using the "*minimum production time*" criterion. However, their methods attempted to optimise the cutting conditions through a trial and error method, some non-flexible assumptions like using equal passes for the machining, although optimal values can be achieved by using unequal cutting conditions.

Friedman and Tipnis [85,86] have introduced a new concept called R-T characteristic functions which opens a new field in economic optimisation of cutting conditions. The concept views all metal removal processes in terms of two basic parameters; the cutting rate (material removal rate) R and the tool life T. The advantage of this method is that, even in the absence of cost data, if the R-T characteristic curve is known, it is always possible to approach the economically optimum point by merely moving the operation points on the R-T characteristics curve within the practical working region.

Hinduja et.al. [87] developed an optimisation procedure for turning operations. Additional constraints such as component instability and workpiece deflection were considered and optimum depths of cut were determined.

Arsecularatne et.al. [88] developed an optimisation procedure for turning, drilling, grooving, threading and parting-off operations by using the same constraints as in Hinduja's work. A similar approach is used for the optimisation of rough turning, boring, facing, threading, drilling, grooving and parting-off operations by introducing additional constraints like cutting tool temperatures, tool wear, motor power of feed-drive motors.

Agapiou [89] has developed a multipass optimisation system for turning processes in which the number of passes have been found by dynamic programming technique, and then the cutting conditions are optimised through the *Nelder-Mead Simplex Search* method by using a combined objective function.

Wang [90] has recently developed an optimisation software for multipass peripheral and end milling operations which used a combination of above two considerations based on the objective function of "*maximum production rate*". He has also verified the superiority of multipass over the single-pass by carrying out some simulation tests. In this paper, the development of a constrained optimisation system for multipass face milling operations is outlined. The optimum number of passes is first determined via dynamic programming, and then the optimal values of the cutting parameters are found based on the objective function; "*maximum production rate*" and using a non-linear programming technique; "*geometric programming*".

2.8. LITERATURE ON "CNC CODE GENERATION"

The innovation of numerically controlled machine tools has been the most significant development in manufacturing industry during the past 40 years. Before the advent of Numerical Control, machine tools were controlled by the skill of operators. With the developments of Numerical Control (NC), the optimisation of machining

process has been introduced and inaccuracies have been largely eliminated. NC machines became Computer Numerical Control (CNC) machines through the development of inexpensive minicomputers and microprocessors with large memories. CNC became one of the most important production devices in the manufacturing industry. The preparation or development of the program of instructions for a particular part to be machined on these CNC machines is actually the task of the part programmers. However, due to variation in proficiency, the work done by the part programmers is sometimes not satisfactory especially for complex part geometries. Therefore, the automatic generation of CNC part programs without any human intervention has long been a challenge to both academia and industry. The automatic generation of NC codes is currently one of the major elements in the CAPP. It also plays a key role being an interface between CAPP and CAM.

It is very difficult to separate the work in the area of automatic generation of CNC codes from that in a CAPP system. Actually, one area in which generative process planning techniques have been more successful is CNC part programming as compared with the other modules. Most of the CAPP systems have converted a cutter location file which contains the tool path generated on a computer assisted part programming language into a part program for a particular machine tool. Some of the promising works on the automated CNC code generation found in the literature are given in the following paragraphs.

Turbo-CAPP [91] which is an intelligent process planning system is able to generate a CNC part program for each planned workpiece based on its geometric characteristics extracted from AutoCAD and the machining parameters stored in the knowledge base. The CNC part programs are generated specifically to run on a variety of CNC turning machines and centres.

TECHTURN [92] (A Technically Oriented System for Turned Parts) generated a CNC part program with little manual intervention by using the outputs from EXCAP (Expert Computer Aided Process Planning). The generated codes have been tested on a Turning Center.

Wang and Lin [93] generated linear and circular tool paths after offsetting the part profile by using the boundary representation of the part from a feature recognition system that is an interface between CAD and CAPP.

Chin [94] developed a feature based knowledge system for CAD/CAM integration in which a cutter location file is generated and stored firstly and then it is converted into an NC code file. The G codes created in this system are for EMCO CNC lathe. However, only external shapes of a cylindrical part are defined and processed in his research.

On the basis of the geometric data and process route generated by processing planning module, ICAPP [95] can generate part programs in higher level languages like APT (Automatically Programmed Tools) and COMPACT II. The programs are then post-processed for a particular controller unit of a machine tool and controller combination.

Shyu and Chen [96] have developed a mini CIM for turning operations in which both cutting paths and CNC part programs are generated. The tool paths are calculated by using the macro instructions prepared previously in the system. CNC instructions are obtained from cutter location file through post-processors as in the case of previous reference.

Boer, et.al. [97] developed a CAPP/CAM expert system in which the tool paths translating the information contained in the synchronised solution in an APT-like cutter location file. This cutter location file is then read from a post-processor that generates a part program file in standard G codes used by the numerical control of a specific machine tool. Recently, Opas and Mäntylä [98] and Dereli [6,99,100] described an approach for automatically generating a part program from a given process plan.

2.9. REVIEW OF CAD/CAM INTEGRATION FROM A DIFFERENT PERSPECTIVE

“Fully Automated Factory” or in other words *“unmanned factory”* has been a major and long-term goal of the people both from industry and academia for a long time, although some authorities claim that it is impossible to achieve [5]. Today, it has been recognised that *Computer Aided Process Planning* is the key system promoting the integration of CAD and CAM systems. It has been seen as the only tool for removing the slash between CAD/CAM.

Process planning forming the link between CAD and CAM, actually translates the design language into the manufacturing language. There have been a great number of efforts and studies to achieve the major goal via CAPP; to effectively integrate the design and manufacturing, providing the automation and standardisation of the process planning function. However, it is worth here noticing at once that the industrial use and applications of the developed systems are quite limited. One of the main reasons behind this lack of industrial use is that the process planning is a complicated and divergent task including many sub-activities from the interpretation of the product data to the tool path planning and CNC part program generation, and between these sub-activities there are many highly complex and ill-defined relationships. Another reason may be the differences between the assumed conditions and actual conditions on the shop floor, or unexpected disturbances, or continuously changing demand, or improved technology. The performance and applicability of the CAPP systems can be improved by including a higher number of the actual manufacturing variables and by providing a adaptive structure to the system.

The CAPP was first proposed by Niebel [101]. Schenck supported the CAPP idea of Niebel by his Ph.D dissertation at Purdue University in 1966 [102]. Due to the limitation of computer hardware and software, the CAPP had not been feasible until the early 1980s.

Two approaches namely; *variant* and *generative approaches* have emerged over the years. Initially, the variant systems were favourable. Most of the commercial

process planning systems make use of variant approach in the planning function. However, the level of integration of CAD/CAM systems is relatively low obtained from the variant approach based systems, although significant improvements can be done over the manual process planning.

The second generation process planning systems were used the generative approach in their planning mechanism. In addition to advantages of provided in the variant approach, the generative systems are capable of producing plans for components not belonging to any existing part family, without any human intervention. The first party generative systems were developed by pure conventional programming techniques. The degree of automation (integration) of CAD/CAM systems by these first party systems was at medium level due to lack of CAD interface capability. Those systems focused on the planning of a limited part spectrum such that the number of different components that a system can handle is quite limited. Decision logic, formulae, manufacturing rules were used to generate process plans in generative systems.

Second party generative process planning systems employed artificial intelligence techniques like expert systems to enhance their capabilities and to improve the level of automation (integration) between CAD and CAM. The use of “*features*” was first taken into consideration in the design stage.

After the features were extensively used in the design stage, the “*feature based systems*” on the CAD/CAM integration have appeared in the literature. The need for the “*feature recognition*” was also conceived. Third party generative process planning systems (also called *Automated Process Planning systems*) generates process plans directly from a CAD model (data). The main difference between an automated process planning system and other approaches is the use of an automated CAD/CAPP interface capability [40] as well as the CAPP/CAM interface capability. Therefore, no human intervention is required for those systems. However, there is no fully automated process planning system commercially available, at present [103].

Due to the rapid diminishing of experienced process planners current industry and the importance of concurrent CAD/CAM, the research on CAPP has gained more attention than ever before. Automated Process Planning received the highest priority among the top ten areas which are to be improved in the manufacturing environment [104].

A large number of process planning systems have been found in the CAD/CAM literature. They are primarily developed to integrate CAD and CAM using the CAPP and its interfaces as key technologies. Each CAPP system has its own method to generate process plans. Reviews of the systems reported in the literature can be found in one of the numerous survey articles [8,14,105-107]. Among these, for example, Altung and Zhang [105] reviewed 156 CAPP systems and discussed 14 of them in detail. They summarised all the papers in a table based on the information such as computer programming language, approaches, input and output styles, data base methodologies used in the development of the system. The interface possibilities and developers of the systems are also declared. Some of the well known process planning systems or in other words; the systems that are considered as the mile-stones of the process planning, are summarised below.

CAPP (CAM-I's Automated Process Planning) [108] system which is developed by *McDonnell Douglas Company* is one of the good examples of variant process planning approach. MIPLAN [109] and MULTIPLAN [110] are also two powerful members of the variant approach.

GENPLAN (GENerative process PLANning) [111] system is a generative system which uses a group technology based coding scheme. It covers both part geometry and process variables to generate comprehensive operations.

AUTAP (Automatistic Arbeits Planerstellung) [112] is developed at the laboratory of Machine Tools and Production Engineering at Aachen Technical University in Germany. It is one of the most complete generative process planning systems integrated with a CAD system and in use today.

CMPP (Computer Managed Process Planning) [113] is developed by United Technologies Research Centre (UTRC) in conjunction with the U.S.A Army Missile Command. It is a comprehensive and generative process planning system aimed at high technology, machine cylindrical parts, tight tolerances and complex manufacturing process.

First AI based process planning system is GARI [114] which is a rule-based CAPP system including 50 rules. The rules are applied by *If-Then* statements. Turbo-CAPP [91], SIPP (Semi-Intelligent Process Planner) [115] system and EXCAP (EXpert Computer Aided Process Planning) [116] are the three powerful examples to the knowledge based process planning systems. SIPP used the frame based knowledge representation whereas EXCAP was a rule-based system and used the backward planning mechanism. TOM [117], and XPLAN [118] also applied AI technologies to the process planning.

KAPPS (Know-how and knowledge Assisted Production Planning System) [119] was one of the first serious attempts to recognise the machined surface and initial shape of the part for which a process plan is to be generated. It was designed for both rotational and prismatic parts.

From the literature survey, following conclusions can be drawn. It has been recognised that some of the process planning modules are less emphasised than the others. The most of the emphasis is given to the feature recognition. However, little work has been reported on the automated selection of multi set-ups for the prismatic parts. There is no work that includes DFM utilities in the process planning. Machining of complex pockets and optimisation of turret indexing positions were not addressed. The use of STEP standard for interfacing CAD to CAPP is limited. The utilisation of AI tools to enhance the capabilities of process planning systems is rare. This work attempts to fill those gaps with a compact package called *Optimised Process Planning System for Prismatic Parts: OPPS-PRI*.

CHAPTER III

ARTIFICIAL INTELLIGENCE AND ITS IMPACT ON CAD/CAM

3.1. INTRODUCTION

In this chapter, the meaning of AI and some of the most effective AI tools are introduced. The applications of AI tools in design and manufacturing are also discussed.

3.2. ARTIFICIAL INTELLIGENCE AND ITS TECHNIQUES

Artificial Intelligence (AI) is a part of computer science concerned with systems that exhibit the characteristics usually associated with *intelligence in human behaviour*, such as learning, reasoning, problem solving, understanding language, and so on. The main goal of AI is to simulate human behaviour on the computer. The knowledge and the use of knowledge are the key characteristics. The art of bringing relevant principles and tools of AI together for solving difficult application problems is therefore sometimes referred to knowledge engineering.

Manufacturing systems in industrialised countries have dramatically changed as a result of advanced manufacturing technologies employed in today's factory. Factories are now striving to attend and maintain a world-class status through automation that is made possible by sophisticated computer programs. The

development of CAD/CAM systems is evolving towards the phase of *Intelligent Manufacturing Systems (IMS)*. The systems belonging to this phase may be characterised by their ability to solve problems without either a detailed, explicit algorithm available for each solution procedure, or all the facts, mathematical relationships and models available in perfect arrangement and complete form for a deterministic and unique answer to be found. A tremendous amount of manufacturing knowledge is needed in an intelligent manufacturing system. AI based techniques are designed for capturing, representing, organising, and utilising knowledge by computers, and hence play an important role in intelligent manufacturing. AI has provided several techniques with applications in manufacturing like; *Expert Systems (ESs)*, *Artificial Neural Networks (ANNs)*, *Genetic Algorithms (GAs)* and *Fuzzy Logic (FL)*.

3.2.1. Expert Systems

An Expert System (ES) also called a *knowledge-based system* is an intelligent computer program that uses knowledge and reasoning techniques to solve problems that are difficult enough requiring significant human expertise for their solution [120]. An expert system may emulate the external behaviour of an expert, or it may attempt to closely model the internal mental processes of the expert as well. ESs have the ability to justify or explain the rationale behind a specific problem solution [121]. ESs are particularly useful for problems based on a limited knowledge domain. Like conventional programs, ESs usually perform relatively well-defined tasks. Unlike conventional programs, expert systems can explain their actions, justify their conclusions, and provide end users with details of the knowledge they contain [121]. One of the main difference between the conventional and ESs is that the knowledge is separated from the algorithms and is readily accessible at a run time in the ESs, while data and algorithm are executed together in conventional programming. An expert system generally consists of the components as shown in Figure 3.1. They are; a user interface, a knowledge base, an inference engine, a working memory, a knowledge acquisition mechanism and an explainer.

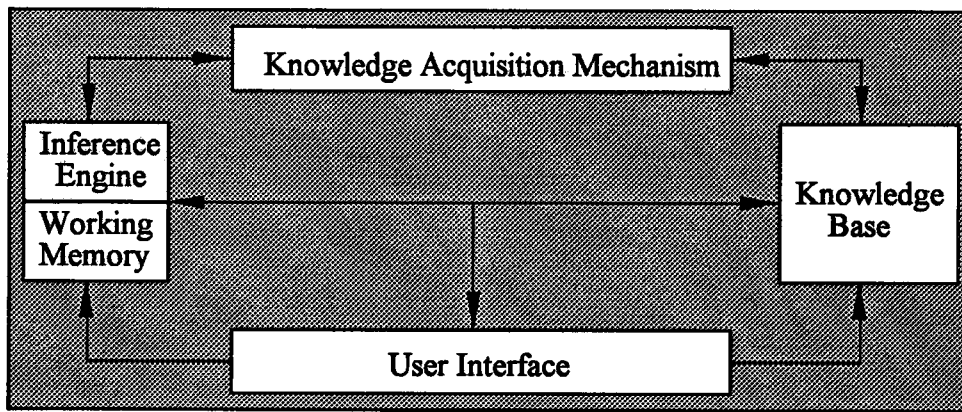


Figure 3.1 General Structure of an Expert System [122]

The *user interface* is designed to provide a convenient means of two-way communication between the user and the inference engine. It enables a tutor to set up and maintain the expert system, and prepare the knowledge to be entered into the knowledge database. An end user who tries to find a solution to a problem can describe the context of his problem to the system by means of the user interface.

An important subset of the general area of ESs concentrates on explicitly representing an expert's knowledge about a class of problems and providing a separate reasoning mechanism called *inference engine* that operates on this knowledge to produce a solution. These kinds of systems are called *knowledge based expert systems*. The *knowledge base* is a file that contains the facts and heuristic that make up an expert's knowledge. A knowledge base is different from a typical data file or data base. In a data base, knowledge about the problem domain may be implicitly represented by the structure of the data base. The actual contents of a data base are the facts, data or information rather than knowledge. On the other hand, in the ESs, knowledge about the problem is explicitly represented in the knowledge base. The knowledge base is expressed in computer codes, usually in the form of "*if-then*" rules (in backward or forward chaining) or frames, with a series of questions. The problems are solved by using these rules through inference engine. The knowledge required to solve a specific problem is called the *domain* of an ES. A *knowledge acquisition mechanism* is used to acquire human expertise and transform into the knowledge base. This module processes the data entered by the expert and transforms it into a data presentation understood by the system. A knowledge base can involve all of the

methods that the expert uses to perform a task. These methods may include computer programs, rules of thumb, theories, logic, etc. When the explicitly represented knowledge is combined with the fact of a specific problem, the ES is able to compute a solution to the problem.

The *inference engine* is the knowledge processor which looks at the problem description and tries to find a solution with the help of factual and meta-knowledge. It can be considered as a program that applies domain knowledge to known facts to draw conclusions. Inference engines are domain independent such that they apply domain knowledge to case-specific application.

The explainer is used to find out how a solution was obtained from an expert system and which individual steps were taken. The user can communicate with the explainer to obtain a report about the operation of the expert system. If the user desires, he can obtain intermediate data and information on how the knowledge was used.

ESs represent a revolutionary transition from the traditional data processing to knowledge processing. They offer an environment for incorporating the capabilities of humans and the power of computers. Some of the advantages of ESs are summarised as follows;

- ESs can accommodate new expertise whenever new knowledge is identified
- ESs are able to explain their recommendations
- ESs can apply heuristics to reduce the complexity of search
- ESs reduce the company's reliance on human experts by capturing expert knowledge and store the knowledge in computers

There are significant implementation challenges with the ESs, known as the knowledge acquisition bottleneck. Some of their disadvantages are listed below;

- Debugging and maintenance of a large (or complex) ES is very difficult
- The human expert must be available in order to build an ES

- The human expert must be able to articulate the rules that define the solution and not lapse into vagueness or incoherence
- The development of an ES can be a lengthy-process depending on the size of the problem domain
- An ES's performance drops off sharply if the problem deviates even slightly from the expected problem domain
- In a large and complex ES, the execution time of the system can also be a problem

There are numerous ESs being developed for almost any manufacturing activity [123]. Many major applications of ESs can be found in the manufacturing area. ES is widely used in design, process planning, scheduling, material handling, quality control, machine diagnosis, machine layout and other operations.

3.2.2. Artificial Neural Networks

Artificial Neural Networks (ANNs) are the second generation AI techniques. The study of neural networks is an attempt to understand the functionality of human brain. Artificial Neural Network (ANN) technology imitates the brain's own problem solving process. Neural networks are systems composed of many simple processing elements operating in parallel and whose functions are determined primarily by the pattern of connectivity. These systems are capable of high-level functions, such as adaptation or learning, and lower level functions such as data pre-processing for different kinds of inputs. Neural networks have been inspired both by biological nervous systems and mathematical theories of learning, information processing and control.

A neural network may be considered as a black box that can accept a series of input data and produce from these one or more outputs [124]. A neural network consists of many non-linear computational elements operating in parallel and arranged in patterns resembling those of biological neurones. The neurones are interconnected into networks via weights which are usually adapted in a learning or training process in order to improve performance. Inputs are supplied to the input nodes. From these nodes, the input values are propagated through the links to the other region of the

network. As they propagate, they are combined at common nodes and changed according to the computational rules of the links at nodes through which they pass. Outputs from the output nodes form the numeric outputs of the network. In summary, a neural network takes an input numeric pattern and produces an output numeric pattern. A typical ANN structure is illustrated in Figure 3.2.

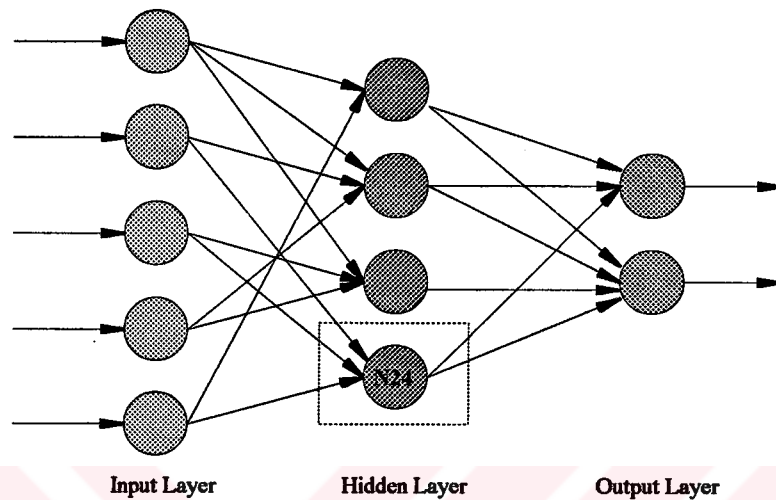


Figure 3.2 A Typical ANN Structure

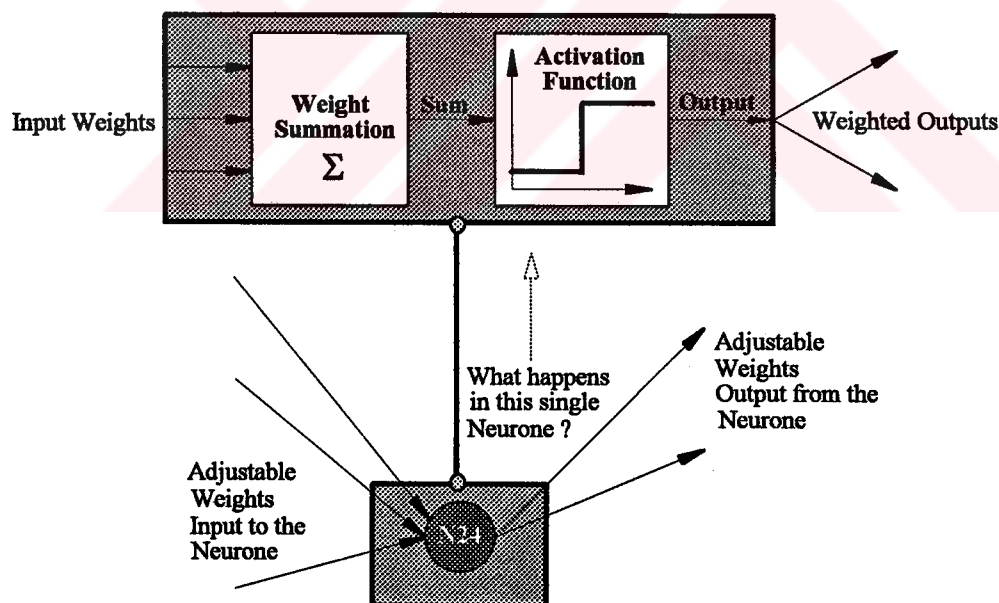


Figure 3.3 An Example of an Artificial Neurone Unit

As depicted in Figure 3.2, the basic components of an ANN are neurones. A group of neurones is called a slab. Neurones are also grouped into layers by their connection to the outside world. If a neurone receives data from outside world, it is

considered to be in the input layer. If a neurone contains network's predictions or classifications, it is in the output layer. Neurones between the input and output layers are in the hidden layer(s). A layer may contain one or more slabs of neurones. What happens in a single neurone (e.g., Neurone: N24) is shown in Figure 3.3.

The neurones are usually characterised by an internal threshold and by the type of their activation (transfer) function. The most common transfer functions are hard-limiter, threshold-logic and sigmoidal function being the most widely used transfer function. The neurones (nodes) in the input layer are fed with input data. Each node sums all the inputs, with one input per node in the input layer but many inputs per node in the next layer. Each node transfers the data, according to a transfer function, to all the elements in the next layer. However, each node receives a different signal due to different connection weights between the nodes. The output of each node in the output layer is compared to the desired output. The difference between the desired output and the calculated one (the error) is fed back to the network so that the weights of the connections between the neurones are adjusted in a way that minimises this error. After repeating this procedure which is called *training* several times, the network is "trained" and can be used with input data not shown previously to the system.

In training of a network, the network is forced to take on particular output form by means of weighting selection for a given input pattern. The biggest secret to building successful neural networks is knowing when to stop training. The choice of learning examples is very important. If there are not enough examples, the network realises a rote learning and then it is unable to answer to new inputs. On the other hand, if the set of examples is too big, the neural network is overtrained and cannot converge to a solution. A good rule of thumb is that the number of training pattern should equal 10 times the number of inputs [125].

Most of the ANN models can be trained adaptively although Hopfield Network and the Hamming Network are usually used with fixed weights [126]. Adaptive neural networks can be trained using two types of training procedures, *supervised training* and *unsupervised training*. In supervised learning, the training

process involves providing the network with known sets of input and the corresponding outputs. A network is said to have learned if it can produce the desired outputs when a sequence of inputs are provided. In unsupervised learning, the learning process is based on clustering. Unsupervised networks can classify a set of training patterns into a specified number of categories without being shown in advance how to categorise.

For a class of multi-layered neural networks through its simplicity, *Back propagation learning algorithm* made the neural networks approach very popular and resolved the doubts regarding the viability of the neural network approach once and for all. The back propagation procedure, the most frequently used learning procedure for ANNs, is a generalisation of the LMS (Least Mean Square) procedure for feed forward, multi-layered networks with “hidden layers”. It uses a *gradient descent procedure* which changes the weights in its original and simplest form by an amount proportional to the partial derivative of the error function (ϵ) with respect to the given weight. In a system that has no hidden units, the computation of these derivatives is relatively easy, but for multi-layered networks it is a harder task. The central idea of back propagation is that these derivatives can be computed efficiently by starting with the output layer and working backwards through the layers.

A potential important advantage of neural networks is their high degree of error resistivity. A normal computer may completely fail in its operation if only a single bit of stored information or a single program statement is incorrect. This leads to a high risk of failure, especially if the system cannot be continuously supervised. By contrast, the operation of a neural network often remains almost unaffected if a single neurone fails, or if a few synaptic connections break down. The advantages of ANNs over conventional computing are listed below;

- ANNs have the ability to generalise. From past experiences, they can give a solution to a new problem.
- ANNs do not need an expert to represent knowledge and does not require much programming knowledge

- ANNs have a simpler structure, since only the input and the output of the system are simulated
- ANNs have a good ability to tolerate faults and missing data, making it useful where all the rules and data are not known
- ANNs are self-organising and can learn. These characteristics make it suitable for those tasks that were originally done by humans
- The massive number of processing elements (neurones) makes neuro-computing faster than conventional computing
- When fully developed, neural hardware is expected to be 20 to 50 times smaller in size than conventional computers

Currently, there is an explosion of interest in ANNs since they have several advantages that are desired in manufacturing practice. However, some of the scientific and economic expectations on ANNs are unreasonable. This is because ANNs have the following drawbacks;

- The configuration of an ANN is usually time consuming, as one needs to use a trial-and-error method to find the proper neural network architecture for a given problem
- The knowledge representation of an ANN is imprecise and not easily understood
- ANNs cannot explain its results explicitly, which implies that the user interface of a neural network may not be as friendly or productive as that of an expert system
- The current learning algorithms for ANNs are not efficient enough and cannot guarantee network convergence
- How to derive some type of optimal training set for a neural network application still remains a question

Note that ANNs should be considered as complementary tools to traditional AI techniques, rather than a competitive one. However, there is no doubt that ANNs will play an important role in the development of IMS, keeping in mind that ANNs constitute an enhancing but not a replacing technology. Having reacted to the attractive results gained during the renaissance of connections research, some attempts were in the last 2-3 years, to apply AI techniques in the Computer Integrated

Manufacturing (CIM), in order to shorten the reaction time of the system, to increase the product quality, to make the systems more reliable and to enhance their intelligence by the knowledge representation and learning abilities of AI methods.

The application area of neural networks in CIM is quite broad. It covers nearly all of the fields spreading from the design phase through simulation, control, monitoring and quality assurance to the maintenance. The majority of approaches relates to sensor processing, especially multi-sensor integration, owing the biological inspiration of the neural network research. Robotics applications form a very large field of neural network research and application. ANNs can generally be used in the following applications; Quality Control, Pattern Recognition, Resource Allocation, Constraints satisfaction (Optimisation), Scheduling, Maintenance and Repairing, Process Control and Planning, Database Management, Simulation, and Robotics Control.

3.2.3. Fuzzy Logic

Fuzzy Logic (FL) is another tool of AI that is gaining popularity in recent years. It is based on the observation that people make decisions based on imprecise and non-numerical information, fuzzy models or sets are mathematical means of representing vagueness and imprecise information, hence the term fuzzy. These models have the capability of recognising, representing, manipulating, interpreting, and utilising data and information that are vague and lack certainty. Linguistic examples are: *a few, almost all, more or less, very important, good, poor, appropriate, acceptable*, etc. [127].

FL is a technique for dealing with sources of imprecision and uncertainty that are non-statistical in nature. It is an innovative technology that enhances conventional system design with engineering expertise. FL is based on a complex mathematical body. However, its practical use is easy to learn. FL is a true extension of conventional logic. Therefore, anything that was built using conventional techniques can be built with the FL. However, in a number of cases, conventional solutions are

simpler, faster, and more efficient. The key to successful use of FL is clever combination with conventional techniques [128].

The basic idea behind the FL is quite simple. An exact rule or law cannot be defined for each possible problem. For these cases, humans approximate well solutions to the problems. This approximation is possibly due to the flexibility in the definition of the *words* that constitute the rules. Hence, human logic is very flexible. FL simulates this human logic using a mathematical model. However, the nature of this simulation is different from that of ANNs. ANNs simulate the humans' nervous system and deals with the learning ability, whereas FL mainly deals with the decision making processes performed by humans based on linguistic terms.

Fuzzy logic is one of the powerful design techniques as with the other techniques like ANNs, FL systems have their own strengths and weaknesses. Fuzzy logic has been mainly suggested to handle fuzzy concepts, inexact information, and approximate reasoning in expert systems [129]. In fuzzy logic, system behaviour can simply be described with a simple rule base containing many "if-then" rules. Therefore in FL, the knowledge representation is explicit, the verification is easy and optimisation of the system performance is possible. On the other hand, FL has no training ability. Everything should be defined explicitly to the system. Trainability is the most important function of ANNs. Therefore, if the explicit knowledge representation capability of fuzzy logic is combined with the learning power of ANNs, a more powerful technology can be obtained. Such systems are called NeuroFuzzy Systems.

The FL was first introduced by Zadeh [130]. From then on many theoretical developments in FL took place in US, Europe and Far East countries, especially in Japan. Although FL technology has been first proposed in US, the first industrial applications are performed in Europe. Japanese researches have been a primary force in advancing the practical implementation of the fuzzy logic. The main reason for this is that FL has not found wide acceptance in US and Europe at the beginning; since the term "*fuzzy*" carries negative connotations in English and the word "*fuzzy*" did

not imply disorganisation and imprecision in Japanese, as it does in English. Japanese firms have now over 2000 patents in the fuzzy logic area [131].

Fisher and Sanyo have produced cameras and camcorders that use FL to adjust autofocus mechanisms and also to cancel the jitter caused by a shaking hand. Nissan has patents on fuzzy logic based braking systems, transmission controls and fuel injectors. AEG and Matsuhita has built a fuzzy washing machine that combines smart sensors with fuzzy logic. The sensors detect the colour and kind of clothes present and the quantity of grit, and a fuzzy microprocessor selects the most appropriate combination from 600 available combinations of water temperature, detergent amount and, wash and spin cycle times. It is very easy to increase the number of examples of FL applications, especially for electronic home appliances. Apart from these industrial applications, fuzzy logic has also attracted much attention in academia. Zhang and Huang have recently developed a fuzzy approach for process plan selection [132]. In their approach, each process plan is evaluated and its contribution to shop floor performance is calculated using the fuzzy set theory.

3.2.4. Genetic Algorithms (GAs)

Genetic Algorithms (GAs) are the most popular approach among combinatorial algorithms like *Tabu (taboo) Search (TS)*, *Simulated Annealing (SA)*. GAs are robust search algorithms based on the mechanics of natural selection and genetics. They are used to abstract and explain the adaptive processes of natural systems to design artificial systems software that retains the important mechanisms of natural systems. Many papers and dissertations establish the validity of GA technique in function optimisation. Having been established as a valid approach to problems requiring efficient and effective search, GAs are finding more widespread applications in CAD/CAM area recently. These algorithms are computationally simple and powerful in their search for improvement and are limited by restrictive assumptions about the search space [133]. GAs are different from other optimisation and search procedures in many ways. GAs work with a coding of the parameter set and search from a rich population of strings. They use only an objective function (fitness) information, not derivatives or any other auxiliary knowledge. GAs use probabilistic

transition rules, not deterministic rules. A genetic algorithm is composed of several genetic operators such as crossover, inversion and mutation. There are also other types of operators that yield good results like partially matched crossover (PMX).

Generation of the initial population of strings is done randomly. The number generator is initialised with the seed supplied in the randseed parameter. The randseed is initialised in turn to the system time. The random number generator is used to select the mutation points that are used to generate the initial population from the input. New chromosomes can then be generated using the genetic operators.

In GA terminology, a candidate solution is represented by a sequence of numbers and/or characters known as a *chromosome* or string. Each element in the string is called a *gene* and represents a process variable. A selected number of strings is called a *population* and the population at a given time is a *generation*. In the following examples, a gene is abbreviated by G in the chromosomes. Genetic operators operate on the *genes* to replace their place within the chromosome.

Simple crossover involves two parents and crossover points are selected randomly. If two parents to be used for generating new chromosomes are; [Parent1:G1-G2-G3-G4-G5] and [Parent2:G5-G3-G1-G4-G2] and a crossover point was chosen randomly as 2; this produces the following children: [Child1:G1-G2-|G1-G4-G2] and [Child2:G5-G3-|G3-G4-G5]. From the example above, it is obvious that using simple one-point crossover produces undesirable results, and therefore, a modified crossover operator was used, referred to as *PMX* (partially matched crossover) [133] or sometimes *LOX* (linear order crossover) [134]. PMX is a method of reproduction that arose to deal with the travelling salesman problem. Under PMX two parents are randomly picked from the population, and two crossover points are randomly chosen. These two points define where the crossover is to take place. The genes between the crossover points are replaced between the parents and the children are generated. The example below illustrates how PMX operator works. If the same parents (Parent1 and Parent 2) are used for generating new chromosomes with PMX or LOX and two crossover points were chosen randomly as 2 and 4; this produces the following children: [Child3:G1-G2-|G1-G4-|G5] and [Child4:G5-G3-|G3-G4-|G2].

These intermediate children are not valid, since some of the genes appears more than once and others do not appear at all. To eliminate this problem, the children go through a validation process that produces valid chromosomes from the invalid children, making sure that the genes between the crossover points are not changed and each gene appears once and only once in a chromosome. The final result is: [Child3:G3-G2-[G1-G4-G5] and [Child4:G5-G1-[G3-G4-G2].

Inversion operates on a single parent. It inverts the order of the element between two randomly chosen points in the parent: [Parent5: G1-[G2-G3-[G4-G5]. Assume that the two random inversion points are 1 and 3 the child generated from applying the inversion operator on the parent using 1 and 3 as the inversion points produces: [Child: G3-G2-G1-G4-G5].

Mutation operation involves a single parent as opposed to crossover. An index into the parent is randomly picked, and the gene at that position becomes the first gene in the new chromosome. From there on, we wrap around the parent to produce the child. This operation preserves some of the parent characteristics. If the parent is: [Parent: G1-G2-[G3-G4-G5], pick position is 2; this operator produces: [Child: G3-G4-G5-G1-G2].

3.3. CURRENT STATUS OF AI TECHNOLOGY AND ITS FUTURE

There have been numerous applications of AI for CAD/CAM for almost all design and manufacturing activities; from feature recognition to optimisation. ES is widely used in design, process planning, scheduling, material handling, quality control, machine diagnosis, machine layout and other operations. ANNs can be used for; quality control, pattern recognition, resource allocation, optimisation, scheduling, maintenance and repairing, process control and planning, database management, simulation, and robotics control. FL has been preferred for those problems in which there are conflicting process parameters, while GAs have been generally used for the optimisation issues such as optimisation of cutting parameters and operation sequences.

One of the problem with the design of CIM modules (especially with process planning) is the software. There are many programming and system design languages being used or that are still under development. On one hand, the impact of Artificial Intelligence (AI) tools (like Expert Systems, Neural Networks, Genetic Algorithms, Fuzzy Logic) on the planning of manufacturing processes has been proven by recent research projects and actual implementations. There are numerous packages being developed for almost any manufacturing activity. At the present time, however, it is not known how many of these systems are practicable and are being used. For example, most of the software for operations planning has led to the development of large and complex application packages which are too rigid when manufacturing variables are included. A conservative estimate is that only 5% of all research endeavours have found their place in the factory. This may be a very discouraging reality; but there are numerous reasons for this problem. The tools for building intelligent systems are not sufficiently developed and are difficult to apply. The methods for acquiring knowledge from experts to develop expert systems are not very well understood. They usually contain several thousand rules and are huge software systems which are difficult to use on conventional computer systems. There are too few qualified people available who really know how to apply AI tools. These problems can be solved by using or developing right tools, methods and environments. However, the potential and power of AI is very great and it is believed that with the exploitation of AI methods it will only be possible to build well conceived and intelligent CIM systems in which many routine jobs (which may become very repetitive and boring, after the skill has been acquired) are taken out of the responsibility of the experienced manager so that his creativity can be devoted to solving more complex problems in factory [3].

The development of powerful, intelligent, optimised and flexible CAD/CAPP/CAM systems in CIM (or IMS) concept will only be possible with the extensive and true use of Artificial Intelligence (AI). AI tools like ESs, ANNs, GAs, FL, SA offer promising solutions in the areas of product definition, layout design, process planning, optimisation and so on. The next generation of intelligent manufacturing systems will hopefully integrate the computational paradigms of expert or knowledge based systems and artificial neural networks, as well as other promising

methodologies like fuzzy logic and genetic algorithms [135]. To design IMSs, one or several AI technique(s) have been utilised. It should be noticed that each AI implementation paradigm has its own advantages or disadvantages. None of them may completely solve “intelligence” problem alone. Therefore, different techniques or approaches related to AI must be used in amalgamation to eliminate and to take the disadvantages and advantages of individual methodologies, respectively. Thus, it will be possible to realise the goals of IMS.

In this work, an attempt has been made to enhance the capabilities of several modules in the process planning system with the use of AI techniques, like optimisation and sequencing issues.



CHAPTER IV

A FEATURE RECOGNITION SYSTEM FOR PRISMATIC PARTS

4.1. INTRODUCTION

In the first sections of this chapter, the basics and the importance of CAD/D (Computer Aided Design / Drafting) and implementation techniques currently in use are introduced. The importance of data exchange between CAD/CAM systems and data exchange standards are also discussed. Feature recognition system developed in this study is then presented through an example.

4.2. COMPUTER AIDED DESIGN/DRAFTING (CAD/D)

Computer Aided Design/Drafting (CAD/D) is generally defined as the process by which computer drawings are produced for a variety of engineering functions including manufacturing and design. It is a term which means many things to many people. To some, it means computer aided drafting or drawing. To others, it means computer aided analysis. And still to others, it suggests totally automated design. CAD is all of the above, aiming mostly for the last description through techniques in artificial intelligence. CAD/D, however, still exists separately in each of the stages mentioned. Probably most common and simplest CAD/D systems are limited to automated drafting capabilities. Modern CAD systems (also called CAD/CAM system) are based on

interactive computer graphics. Interactive computer graphics denotes a user oriented system in which the computer is employed to create, transform, and display data in the form of picture or symbols, and data of the object is stored in the memory for CAD/CAM applications.

4.3. GEOMETRIC MODELING

Geometric modelling is the process in which a geometric model is created to represent the size and shape of a component in computer memory. It is the starting point of the product design and manufacturing process. Another crucial reason for its importance is that the accuracy of the model and the way in which the model is structured and stored in computer memory will have far-reaching effects on other CAD/CAM functions such as finite element analysis, draughting and NC part programming. Geometric modelling is often performed with the assistance of software programs like AutoCAD, Pro/ENGINEER, CATIA, etc., since interactive graphics enables users to enter, manipulate and modify the data easily for the construction of geometric models. A geometric model representing a component in computer memory may be a 2D or 3D depending on the capabilities of the CAD/CAM system and the requirements of the users.

There are generally three types of modelling scheme in common use to represent a physical object in CAD/CAM systems. They are *wire-frame modeling*, *surface modeling* and *solid modeling* schemes. An example for each system is shown in Figure 4.1, Figure 4.2 and Figure 4.3, respectively.

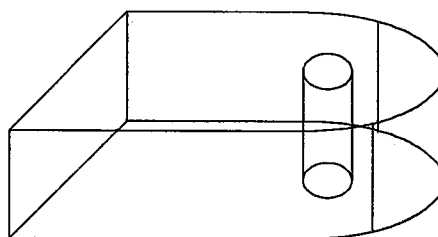


Figure 4.1. Wire-frame Modelling

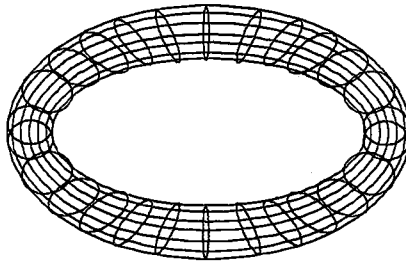


Figure 4.2. Surface Modeling

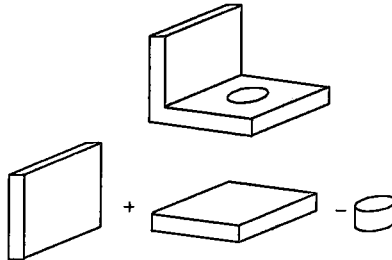


Figure 4.3. Solid Modelling

Wire-frame modeling is one of the simplest and most basic methods of geometric modelling. It is most commonly used to define computer models of parts especially in computer aided drafting systems. Wire-frame modeling uses points, lines, circles and arcs to define objects. The wire-frame models are simple and easy to create, and they require little computer time and memory, even inexpensive computers can cope with the processing needed. However, it has a number of limitations. For example, although the wireframe models provide accurate information about the location of surface discontinuities on the part, they do not give a complete description of the part. Wireframe models cannot be used in the downstream applications in the CAD/CAM integration like NC part program generation since they contain low level information. The representation of real and complex objects is ambiguous. It is not easy to represent typical complex components of cars, planes, etc.

Surface modeling is a higher level of sophistication in geometric modelling which is used to overcome many of the ambiguities of wireframe modeling. It includes the definitions of surfaces, edges and vertices. Surface models actually involve the wireframe modeling. They contain extra information like connectivity of the surfaces as well as the information of a wireframe model. The surface models may be constructed using a large variety of surface features often provided by many

CAD/CAM systems. The plane is the most basic feature to represent a surface element. More complex shapes can be defined by tabulated cylinders, ruled surfaces of revolution, sweep surfaces and fillet surfaces. Surface modeling can be used for downstream applications, for example to generate the tool path, as opposed to wireframe modeling. It is especially useful for industrial design applications and when primary importance is given to “*aesthetic*”. However, surface modeling has also its own drawbacks. Surface models are more difficult to create than the wireframe models. They do not actually represent the solid nature of parts, because they contain no information describing what lies within the part interior or exterior. They cannot be used as a basis for engineering analysis programs such as finite element and model analysis for stress and strain predictions.

Solid modeling is one of the latest developments in geometric modeling. In a solid modeling system, objects are described by primitives. With the solid modeling, a complete and unambiguous representation of a part can be obtained. Solid models are recorded mathematically in the computer as volumes bounded by surfaces rather than as stick-figures. As a result, it is possible to calculate mass properties of the parts which are often required for engineering analysis such as finite element methods and kinematics studies for interference checking. Despite the benefits of using solid models, they have their own disadvantages. The one disadvantage of solid modeling is that it uses much more memory to represent a model than wireframe and surface modeling. Another considerable disadvantage is that, they require extensive processing for their manipulations owing to the more complicated data structure and associated mathematics.

4.3.1.Importance of Solid Modeling

The complete definition of a part shape including its geometry and topology through solid models has been called a functional key to CIM. A solid modeler actually allows user to design a workpiece on the monitor and store the model in a 3D data base. There exists commonly three type of formats for storing solid model data, namely [136];

- Constructive Solid Geometry (CSG) models
- B-Rep (Boundary Representation) models
- Hybrid models

CSG and B-Rep are the most widely used representation schemes for solid modeling. A CSG model is based on the topological notion that a physical object can be divided into a set of primitives (simplest solid objects) which can be combined in a certain order following a set of rules (Boolean operations) to form the objects. CSG stores a model in an *unevaluated (implicit)* form such that the edges and vertices should further be determined from the data by *Boolean operations*.

B-Rep describes the geometry of an object in terms of its boundaries with faces, edges and vertices. B-Rep involves an *evaluated (explicit) model* including all relational information about faces, edges, and vertices on the component to be modelled.

Both CSG and B-Rep have their advantages and disadvantages. For example, B-Rep is more suitable for representing complex designs, while CSG is preferred for representing relatively simple parts. However, CSG models are easier to create than B-Rep models. The structure of the CSG models is concise. CSG models require less storage capacity. However, they are very slow in displaying the objects on the monitor as compared with the B-Rep models. For this reason, they are usually converted into B-Rep models to display the model.

It is however not obvious whether the CSG or B-Rep type of modeling scheme is globally advantageous over the other. One may be more suitable than the other in different applications. Since both have some advantages and disadvantages over each other, CSG and B-Rep systems are unified in a *hybrid modelling system* [137]. Today, many solid modeling environments provide both the B-Rep and CSG systems.

4.4. FEATURE BASED DESIGN AND FEATURE RECOGNITION

4.4.1. Definition of a Feature

Features have been claimed to be the enabling technology for many CAD/CAM applications. There are many published definitions of the concept of a feature. Even though these definitions seem to be dissimilar, they all consider features as entities which are of semantically higher level than the pure geometric elements typically used in solid modeling systems. The geometric aspects of features are defined as volumes enveloped by a set of real and imaginary faces. Geometric elements are solid primitives in CSG type solid models (blocks, cylinders spheres, tori) or boundary elements used in B-Rep type solid models (faces, edges, vertices). Almost universally, the concept of generic feature classes is used, and models are built from instances of generic features. The generic types may be organised into a feature taxonomy, often realised as a collection of classes with inheritance of information according to the principles of object-oriented programming.

In the type-instance approach, feature instances are represented in terms of various feature attributes. Common attributes include the intrinsic geometric attributes of the shape corresponding to the feature (length, width, depth, radius), the position and orientation of the feature with respect to some global coordinate frame, geometric tolerances, material properties, references to adjacent and other features. Types contain information shared by all instances of the type. In the object-oriented approach to features, this information often is in the form of procedures for computing interesting properties of the instances like volume.

4.4.2. Manufacturing Features

Features can also be defined from different viewpoints, such as design, analysis, assembly, and function. Because of this, there may be several co-existing feature models of the same product design. This gives rise to the problem of feature mapping, i.e. conversion among the various viewpoints. A manufacturing feature is commonly defined as a collection of related geometric elements which correspond to a particular

manufacturing method or process, or which can be used to reason about the suitable manufacturing methods or processes for creating that geometry.

The link between manufacturing features and manufacturing knowledge is typically realised through manufacturing process models. For machining, process models can be organised into a taxonomy containing elementary processes such as milling, drilling facing, and turning. Process models are expressed in terms of the manufacturing resources which can be used to realise the process (machines, tools, fixtures, auxiliary materials), process parameters related to the use of the resource (for machining, feed and speed), and attribute information which can be used to guide the choice of a particular process (such as time and cost). Often, procedural knowledge is included, such as procedures for computing the process parameters on the basis of feature attribute information. An important aspect of a process model is representation of the tool kinematics, such as access direction, and possible technological constraints.

To implement the link, manufacturing feature types refer to a collection of possible process models which can be used to generate instances of the feature type. Thus, for instance, a hole feature would be linked with the process models related to drilling and milling processes.

4.4.3. Part Representation Using Features

Research into the use of features as a means of part representation has allowed work to be done on three aspects of representation generation, namely;

- human-assisted feature definition
- automatic feature recognition
- design by features (feature oriented design)

In the human assisted feature definition approach, the user interacts with the CAD model to define a feature by picking up entities from the part drawing that

constitute a particular feature. These systems cannot validate the user's actions, thus employing this approach is inefficient and inconvenient.

Automatic feature recognisers conceive features after the part is modelled on a solid modelling system. Recognition of features is performed using the geometric and topological information of the CAD data [138]. A specific geometry/topology configuration is searched for in the part model to detect the presence of a particular type of feature. However, the algorithms devised for recognising the features in these systems are fairly complex in nature.

In *design by features* (feature oriented design) method, the feature information is stored during the design phase of the part model. The designer can create the component model using Boolean operations and by instantiating the generic features at a desired place. This eliminates the need for feature recognition. However, *Design by features* approach has its own drawbacks. First of all, since the number of features is infinite, all possible features cannot be accommodated in the feature library. Secondly, feature validation needs to be performed every time a new feature is added. Thirdly, the designer is forced to decide upon the type of the primitive to be used. Design by features method is well suited to *Design for Manufacturing* applications, since it allows the designer to be advised some manufacturing considerations without actually being constrained by them.

Both *design by features* and *automatic feature recognition* approaches have their advantages and disadvantages. Automatic feature recognition systems are well suited for dealing with interacting and complex non-primitive features such as virtual or composite features. However, in *design by features* methodology, the designer is limited with a previously specified feature library. The recent and widely shared notion by many authorities in the field of feature technology is that a hybrid architecture incorporating both approaches called *feature based system* can solve the problems associated with the feature technology. The feature based system should be a blend of *design by features approach* and automatic feature recognition systems. It is an incubating phenomena and needs further research.

Some CAD systems describes a component by low level information such as points, curves, surfaces and primitive solids. However, in downstream applications like CAPP and CAM, form features such as slot, hole, pocket, etc. and technological information, namely tolerance, surface roughness, hardness, material, etc. are required. In these circumstances, in order to interface CAD and CAM, *Feature Recognition* system which analyses a CAD model and identifies form features in it can effectively be used. In this case, only geometric and topological information about a form feature can be extracted.

Some other systems based on feature instancing (design by features) enables the designer to specify the characteristics of a part and the information required for design and manufacturing at the design stage and represented by a set of interrelated part features, with the use of the part feature concept. The meaning of *design by feature* method is generally confused with the Feature Based Design. A Feature Based Design system is also a design by feature system. Vice versa is not always true. If a CAD system allows the extraction of technological information together with the geometric and topological information about the component, it can be called as *Feature Based Design* system in which CAD data can be shared by design, process planning and manufacturing systems. This sharing operation can generally be realised using a common data base in a specific data exchange formats like STEP. However, *Feature Based Design* systems currently suffers from apparent drawbacks. For example, only those features (simple features), which are a product of a 2D profile swept through a linear distance along an axis normal to their plane, have been supported. Especially, when extensive feature relations and interactions are encountered on a typical component, the available features in the feature library have not been enough. Hence, in such cases, there is a strong need for the feature recognition. Feature validation requirement reinforces the notion that automatic feature recognition is closely linked to it and would play a dominant role in the feature based design systems of the near future. Moreover, some technological information forces a model to be a lower level model. For instance, most of the geometrical tolerancing on the component is performed using the relation between the surfaces, not the features. So, it is extremely important to obtain a B-Rep data including information on surfaces, edges, vertex and points for a component. Such data might be best processed by automatic feature

recognition systems, rather than the *design by features* methodology or feature based design system itself. It is worth to point out that feature based design systems is still in its infancy and needs further research to exploit its full potential [138].

4.5. DATA EXCHANGE STANDARDS AND STEP STANDARD REVIEW

4.5.1. The Need for Data Exchange

Most initiatives for data exchange standards came from CAD users, particularly the aerospace and automotive industries. Aerospace and automobile companies were among the first serious users of CAD. Several of them were sufficiently advanced in their need for CAD that they developed their own CAD systems [139]. They also bought commercial CAD/CAM systems. It was soon apparent that they needed to exchange data between different systems [140].

There are two important reasons for CAD data exchange. The first one is the need for enabling parts of companies with different CAD systems to transfer data to each other. The second one is due to the facilitation of concurrent engineering within a company or between companies. Many companies share out their works to subcontractors. Both leader companies and their subcontractors have a lot of different CAD systems. For example, Volkswagen and their subcontractors have had more than 60 different CAD systems [140]. If CAD data had not been exchanged or every drawing exchanged had to be recreated or modified on a different CAD system, the consequences for productivity would be disastrous and the probability of error be very high. All these problems have been recognised and, as a result, works have been undertaken to develop standard formats that would provide available means of communication between CAD systems since 1980s. At present, it is not so difficult for CAD systems from different vendors to exchange geometric information. Several formats such as IGES, SET, VDAFS, PDDI, CAD*I, DXF have been developed. However, none of them solved the problems, especially related to the exchange of full product data including technological information as well. To solve this problem, more recently, available standards and their backgrounds have been drawn together by the

ISO (International Standardisation Organisation) into a unified standard called the STEP (Standard for Exchange of Product Data).

4.5.2. Functions of Data Exchange Standards

The data exchange standards are mainly defined as the structures (neutral formats) that make the transfer of the part information or product models possible between CAD and between CAD/CAM systems. Therefore, data exchange standards have a great importance for the realisation of process planning [141].

The function of the data exchange standards is illustrated in Figure 4.4. Another function of the data exchange standards can also be defined as the mechanisms providing data exchange between different CAD and CAM platforms. The use of the standards for this motivation is shown in Figure 4.5. Data exchange mechanism between two dissimilar CAD systems is depicted in Figure 4.6. As shown in this figure, each CAD system requires two programs, i.e. a pre-processor and post-processor. Once a part is designed on a CAD system, its own internal representation format is converted into the standard form through the use of the pre-processor to send the part information to another system. Likewise, the post-processor converts the data coming from other CAD systems in the standard format into the system format, then this part can be seen on the CAD system and modified if necessary.

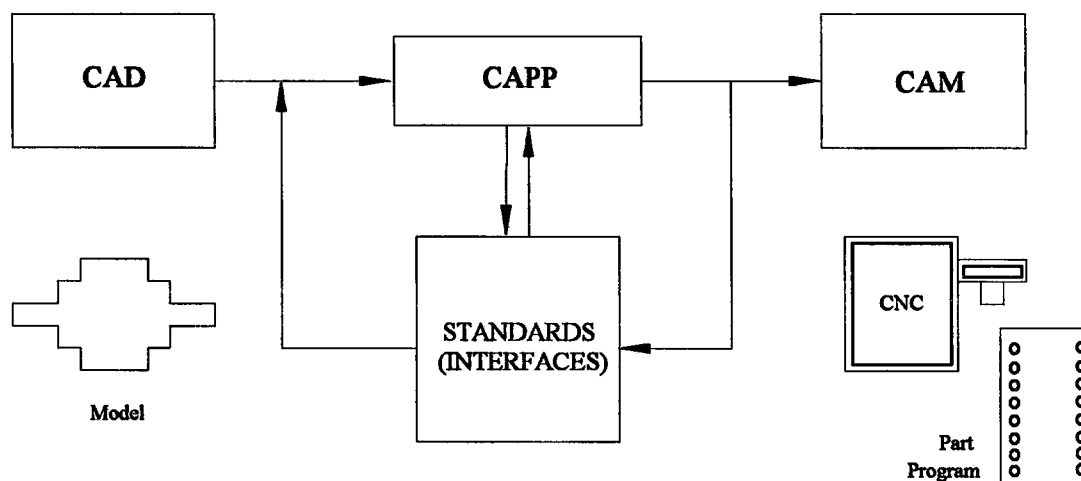


Figure 4.4. The Function of Interfaces in CAD/CAM Integration [141]

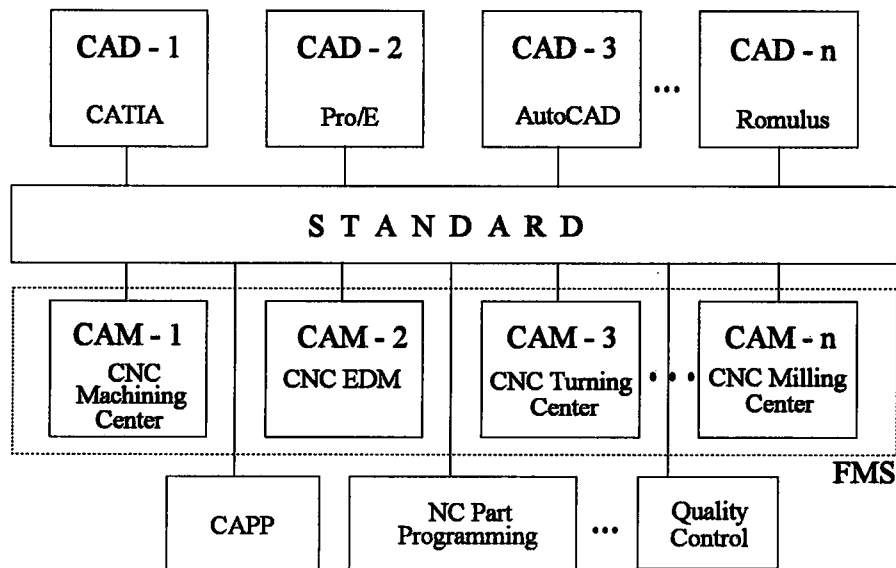


Figure 4.5. Data Exchange Between CAD/CAM Systems [141]

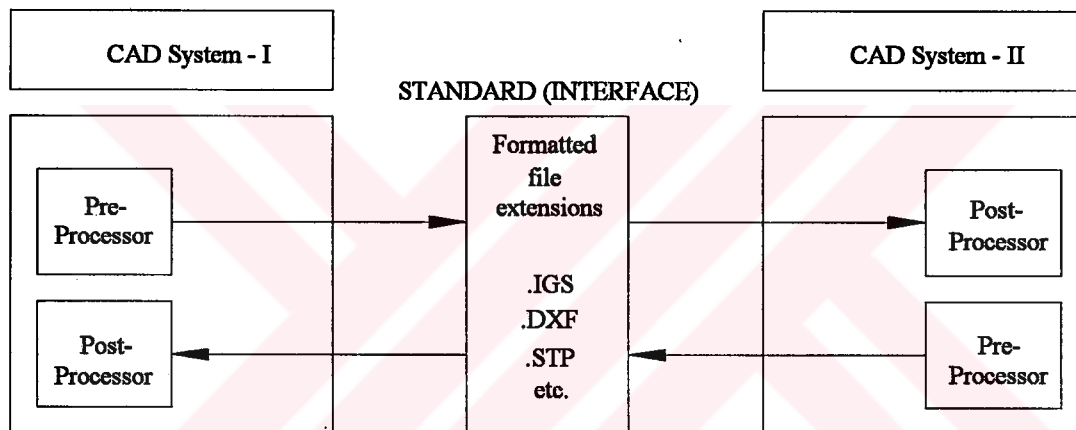


Figure 4.6. Data Exchange Between CAD Systems with Their Pre/Post Processors

4.5.3. DXF (Data eXchange File) Standard

Beginning of the early 1990s, PC based CAD systems have come to dominate the market in terms of the number of copies of software sold. Among those PC based CAD systems, AutoCAD by Autodesk, Inc., has had an extensive number of end user. The main reason for its large market share is due to the adaptation of the approach of making AutoCAD relatively straightforward to work with the AutoCAD files. For this purpose, they used format in a readable form using ASCII for storage of files [142]. This format is known as DXF (Data Exchange File) format and has found a widespread usage around the world. Today, many software packages support the

DXF format. DXF format is quite wordy. It uses a single line for each data item. Therefore, the length of the DXF files are long. It stores all necessary information for external use about the geometric entities which are used to draw a part. The format of this file is open to the user. A section of DXF file for the circle shown in Figure 4.7 is given in Table 4.1.

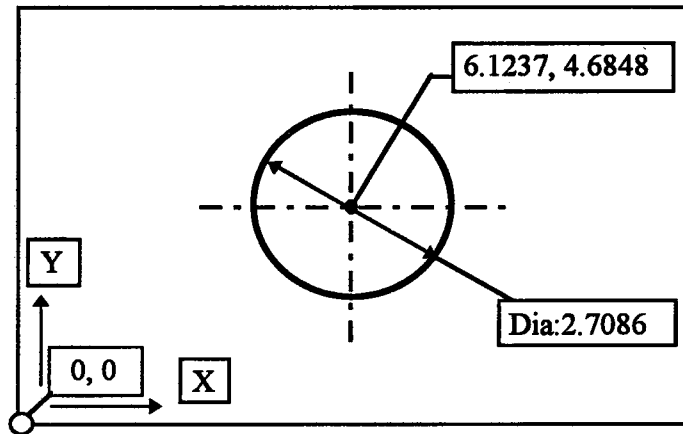


Figure 4.7. A Circle Drawn Using AutoCAD and its Geometrical Information

Table 4.1. A Section of DXF File Generated for the Circle Shown in Figure 4.7

0	AC1009		8	0.0
SECTION	9		0	40
2			10	1.390313
HEADER			6.123695	0
9		ENTITIES	20	ENDSEC
\$ACADVER		0	4.684783	0
1		CIRCLE	30	EOF

4.5.4. IGES (Initial Graphics Exchange Specification) Standard

IGES was primarily introduced to transfer part definition data among CAD systems and is probably the most widely used standard by CAD systems. It was developed to cover a wide range of application areas. The product is represented in terms of geometric and non-geometric information. The geometric information includes points, curves, surfaces, solids and relationships, which define the physical shape. The non-geometric information is divided into annotations, definitions and organisational data. IGES supports the wireframe, CSG and B-Rep models in addition to Finite Element Modelling (FEM).

IGES addresses the problem of exchanging CAD data between one CAD system and another. It is not concerned with communications between CAD systems and other computerised functions in a factory. IGES does not support complete product definition including features, tolerances, surface roughness, material, etc., which is essential for data transferring between CAD and CAPP/CAM systems. It assumes that a person is available at the receiving end to interpret the meaning of product model data. For instance, a person is required to determine how many holes are in the part because a hole itself is not an entity in the IGES standard [143]

In IGES, an entity is assigned a number. It reserves numbers 100 through 199 to define geometric entities. Some of the geometric entities and their codes are given in Table 4.2. An IGES file is an ASCII text file and has six sections; flag section, start section, global section, directory entry section, parameter data section and terminate section. The lengths of the IGES files are quite long as in the case of DXF files, since the numbers are expressed in terms of characters. An IGES file which is created for the circle shown in Figure 4.7, is illustrated in Figure 4.8. Although it was the most widely used standard, there are three common problems with IGES files, like the loss of functionality of information in the data exchange and file expansion.

Table 4.2. Some of the Data Types in IGES Standard

Code	Geometric Entities	Code	Geometric Entities	Code	Geometric Entities
100	Circular Arc	112	Parametric Spline Curve	142	Curved on a Surface
102	Composite Curve	126	B-Spline Curve	143	Surrounded Surface
114	Spline Surface	120	Revolved Surface	144	Trimmed Surface
128	B-Spline Surface	122	Cylinder		
110	Line	141	Boundary Surface		

IGES file generated from an AutoCAD drawing by the IGES	S0000001
translator from AutoDesk, Inc., translator version IGESOUT-3.04.	S0000002
„9HC:\CIRCLE,13HC:\CIRCLE.IGS,10HAutoCAD-12,12HIGESOUT-3.04,32,38,6,99,	G0000001
15,9HC:\CIRCLE,1.0,1,4HINCH,32767,3.2767D1,13H961017.104438,	G0000002
7.5140082228766D-9,7.5140082228766D0,14HMzzybek,21HG.unk.,6,	G0000003
0;	G0000004
100 1 1 1 0	00000000 D0000001
100 2	D0000002
100,0.0, 6.1236954970275D0, 4.6847826074201D0, 7.5140082228766D0,	1P0000001
4.6847826074201D0, 7.5140082228766D0, 4.6847826074201D0;	1P0000002
S0000002G0000004D0000002P0000002	T0000001

Figure 4.8. The IGES file of the Circle Shown in Figure 4.7.

4.5.5. SET (Standard d'echange Et de Transfert) Standard

SET is a French national standard. SET allows the data exchange between dissimilar CAD/CAM systems. It was developed to exchange data in a more compact structure than that of IGES. It has a more general data structure than IGES and the data is compressed. Hence, the neutral data file size is smaller than that of IGES. Wire-frame models, surface models, B-Rep models, FEM models and technical drawings can all be transferred via SET standard. It has found many applications in the European aircraft industry.

4.5.6. VDAFS (Verband Der Automobilindustrie FlachenSchnittstelle) Standard

VDAFS was developed by the Association of German Automobile Industry for the transfer of 3D curves and surfaces for which IGES does not offer a satisfactory solution. VDAFS has been accepted as a standard by DIN. VDAFS has a file format for free-form surface exchange. It is possible to transfer any surface including the orientation of cutting tool with the VDAFS. The data structure of this standard is ill-defined. Moreover, the transfer of the text, dimensions and cross-hatching are not possible. The extension of a VDAFS file is VDA.

4.5.7. CAD*I (Computer Aided Design * Interface) Standard

The main goal of the CAD*I standard was to improve the deficiencies of IGES, SET and VDAFS standards. The CAD*I interface was also an attempt to develop solutions which could be directly implemented in STEP. The main emphasis of the CAD*I is given to the exchange of CAD and FEM models, to the data exchange between various CAD systems and to the conception of a neutral data bank for CAD data. The internal structure of a typical CAD*I file is given in Table 4.3.

4.5.8. PDDI (Product Definition Data Interface) Standard

PDDI is different from the other standards in that it was one of the first attempts to realise the exchange of complete product information between CAD and CAM.

Table 4.3. The Internal Structure of a Typical CAD*I File

```

CAD*I_FORMAT_BEGIN_29851012 CAD*I_FORMAT_BEGIN_29860612
HEADER ('Unix 32','1996.10.08','3c','0','0','0',3,3, 2);
WORLD (OPEN); WORLD_HEADER(+1.000E-03, +1.000E+00, 19.182E+04, 19.182E-02);
SCOPE; B_REP (#1:OPEN);
POINT_CONSTANT (.....); DIRECTION (....); LINE (.....); CIRCLE (.....);
BOUNDED_CURVE (.....); PLANE (.....); CYLINDER (.....); VERTEX
(.....); EDGE (.....); LOOP (.....); FACE (.....); SHELL (.....);
END_SCOPE; B_REP(CLOSE);
END_SCOPE; WORLD (CLOSE);
CAD*I_FORMAT_END_29851012CAD*I_FORMAT_END_29860612

```

It was designed to provide a formal specification of the product model including 3D geometry, topology, tolerance, features and other non-geometric information. The results and experience gained from the PDDI efforts were used for the development of PDES activities and hence utilised extensively in the STEP.

4.5.9. PDES (Product Data Exchange Specification) Standard

Before the development of PDES standard, data exchange standards could not solve the communication problem between CAD, CAPP and CAM systems since they were originally developed to transfer geometric data between dissimilar CAD systems rather than between CAD and CAM systems. They do not comprise feature definitions and other technological information and cannot be used for a complete product definition. After the requirement for a new standard that can be used for complete product definition has been recognised, PDES was designed by the IGES/PDES Organisation in 1984. Although, it has been presented by the IGES organisation, it is currently being developed within the ISO work-group TC 184/SC4. The work-group named the project STEP (STandard for the Exchange of Product data). *Therefore, PDES is actually considered to be the initiation stage of the STEP and, PDES and STEP are considered to be virtually the same.* In USA, the PDES acronym has been still retained for various reasons in the term **Product Data Exchange** using STEP, and is used instead of STEP. It is intended that STEP will replace all existing standards into a unified and powerful standard. PDES, PDDI, CAD*I, IGES, VDAFS, and SET have contributed to develop the STEP standard. A more detailed information about the STEP standard is given in the next section.

4.5.10. STEP (STandard for the Exchange of Product data) Standard

STEP is the international standard for geometric and non-geometric data transfer between heterogeneous CAD, CAE and CAM systems. STEP, sometimes called *Standard for External Representation of Product Model Data*, does not attempt to replace all the existing standards (except PDDI and PDES which are already amalgamated with STEP) that provides communication between CAD systems only using geometric entities. Instead, STEP has been specified both to incorporate other standards and to provide a framework within which other standards can be developed [140]. The STEP which is also referred as ISO 10303 can be defined as; *a series of standards providing a platform or a mechanism that is designed for the standardisation and definition of all the processes and their parameters required to convert a CAD model to a product, independent of any of the design, application or manufacturing hardware and software or system* [3,144]. STEP is an attempt for global standardisation of information exchange related to automated manufacturing. Therefore, it is a more comprehensive and complex standard and designed to support various applications in industry.

The development of the STEP standard is being undertaken by a number of committees and working groups from many countries that deal with different aspects of the standard. A working draft of STEP Version 1.0 was published in 1988 and accepted by ISO as a first draft proposal of an international graphics data standard. STEP comprises *both geometric and technological information* including geometry, topology, features, tolerances, surface roughness and material necessary to completely define a component.

STEP is divided into a number of separate standards called *Parts* [142]. For example, Part 1 is the overview, Part 11 is devoted to the *Express Language* which is the core of the STEP standard. From Part 40 to Part 49 and from Part 100 to Part 199, integrated resources are covered. For example, Part 45 is for “materials” and Part 103 is concerned with “FEM”. From Part 200 onwards, *application protocols* for particular applications are described. For instance, Part 207 of the STEP deals with “sheet metal die design and planning”, while Part 204 applies to “mechanical design

using B-Rep technique”. Among the all *parts* of the STEP standard, Part 41 and Part 42 have the particular importance since these parts are related to foundation of CAD. Part 41 is concerned with “fundamentals of product description and support”. Part 42 covers “geometrical and topological representation in modelling”. Similarly, for process planning and CAM applications, Part 213 which is related to NC process plans for machined parts has a primary importance. STEP has been currently adopted by many major CAD/CAM system vendors, so it is commercially available. Pro/Engineer is one of the professional software which has supported STEP standards since 1994. CADD5, CATIA, Ansys, Cimatron have recently adopted its STEP interface for exchanging product information. It has been recognised that a complete exchange of product data can only be achieved using STEP standard.

STEP has a three phase (layer) architecture; *application layer*, *logical layer* and *physical layer* as shown in Figure 4.9, and based on the use of reference models and formal languages. In the first phase which is an interface between the standard and the user, the application model (information structure) is expressed. The models for each specific area of interest are developed at the application layer. These models are called topical models. Topical models are then integrated into a complete information model called logical layer (or sometimes called the Integrated Product Information Model (IPIM)). The second layer is actually used to provide a consistent system independent description of application models with the use of EXPRESS language. Output of the logical layer includes the information, objects, entities, attributes, relations, associations and conditions. The last layer is the physical layer in which data formats and files are described in the *Bachus-Naur-Form* of the *Wirth Syntax Notation* (WSN) which is a context-free formal grammar. The output of physical layer is a physical file (STEP file). A detailed content about the STEP development layers are given in the following paragraphs.

The *application layer* is composed of application area, shape models, geometry, topology, form features, dimensions and tolerances. The first thing to do is to decide about the application areas, like FEM, drafting, electrical and electronic applications, etc., of the product. Then, a shape model which is specific to the chosen application area is to be selected. Wireframe, surface or solid modeling associated with CSG or

B-Rep schemes might be selected as the shape model for the particular applications. A common scheme is used for the definition of 2D and 3D geometry in the STEP. The geometric entities which are used to define the geometry of a component involves; coordinate system, axis, point, vector, curve, surface, and so on. Topology is another element in the application layer. The basic topological entities are *vertex*, *edge*, *face*, *loop* and *shell*. Vertex is the primitive entity among the hierarchy of topological entities. All other topological entities are defined in terms of vertices.

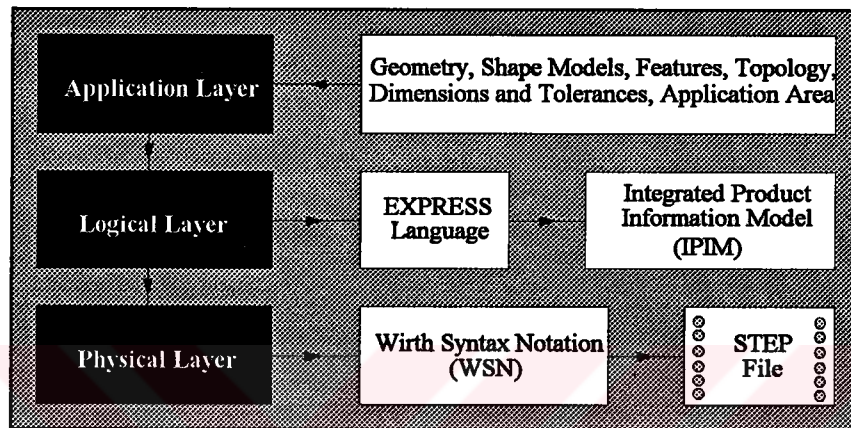


Figure 4.9. The Structure of STEP Standard.

Many of the topological entities have optional geometric attributes which associate them with the appropriate geometric data. The geometric associations for vertex, edge and face are point, curve and surface, respectively. The simple definitions of the topological entities are as follows [143];

- A vertex is the topological entity corresponding to a point.
- An edge is the connection between two vertices.
- A face is a portion of a surface bounded by loops. It is represented by its bounding loops and has a topological normal.
- A loop is combined by stringing together vertices and edges beginning and ending at the same vertex. It is typically used to bound a face lying on a surface.
- A shell is used to bound a region. It is constructed by joining faces along edges. The shell is represented by a collection of faces.

Form feature data is applied in the context of a geometric model (e.g., B-Rep, CSG) of the shape which includes the feature. There are two types of form feature representations in the application layer; explicit and implicit form feature representations. In the explicit representation of form features, the shape elements of a feature are given explicitly. For instance, an explicit face pocket might be the identification of five constituent faces. However, in the implicit representation of the form features, information is given parametrically rather than geometrically. For example, a hole might be described by giving its diameter and the centre line rather than a cylindrical surface. Finally, dimensions, dimensional and geometric tolerances can also be defined in the application layer of the STEP.

The specific models prepared at the application layer are combined into a complete information model; logical layer in which *Integrated Product Information Model (IPIM)* is prepared through the use of EXPRESS language. It is a lexical language designed for the STEP standard. It is syntactically similar to Pascal programming language and enables object-oriented programming. It has declarative constructs based on entity, attribute and object paradigms for specifying, structuring and partitioning data [140]. EXPRESS also provides the fundamentals of software tools for the definition of product information and description of data files. The main elements of EXPRESS are as follows; *Schema, Type, Entity, Algorithm, Function, Procedure* and *Rule*. The *schema* is a description for an area of interest and is composed of declarations of types, entities, functions and so on. A *type* describes the characteristics of the data which is used to represent entities. The structure of the EXPRESS is based on the definition of entities. An entity has attributes which either constitute an entity or can derive from it. An entity can also have restrictions which define the range of values for the attributes. A schema is actually defined as the collection of the definitions of entities, data types and associated constraints. *Rules* define the constraints on entities and attributes. *Functions, procedures* and *rules* are used to describe the behaviour of entities. Entities, procedures and functions begin with '*' and end with '('. EXPRESS has also mechanisms for capturing the hierarchical nature of some data types, so builds a conceptual tree of information. It offers a powerful facility for modelling information in terms of entities and their

relationship to each other [142]. An example of entity definition for a right circular cylinder is given in the following schema [3];

```

*)
ENTITY right_circular_cylinder;
    SUBTYPE OF (primitive_with_one_axis);
    radius      : real;
    position    : axis1_placement;
    height      : real;
    WHERE
    radius > 0.0;
    height > 0.0;
END_ENTITY;
(*)

```

Here, radius, position and height are the attribute definitions. Constraints (prepositions) associated with the attributes are given under the heading of WHERE. SUBTYPE is a kind or variety of entity that its SUPERTYPE. It is declared that, the entity (right_circular_cylinder) is as subtype of the class of primitives with one axis.

Integrated product data entities formulated in EXPRESS at the second layer are then mapped on a physical file. The STEP exchange file is a sequential ASCII file, and has a modular and free format. The file consists of two sections; *header* section and *data* section as shown in Figure 4.10. A STEP file is initiated by “ISO-10303-21;” and is finalised with “END-ISO-10303-21;”. Between these two labels, there are *header* and *data* sections. The header section includes the file identification, implementation level, classification, etc. The data section is used to define the product data to be exchanged.

```

ISO-10303-21;
HEADER;
FILE_DESCRIPTION('', '1');
FILE_NAME('CUBE', '1997-02-06T10:56:15', ('mutude'), (''),
'PRO/ENGINEER BY PARAMETRIC TECHNOLOGY CORPORATION, 9537', '');
FILE_SCHEMA(('CONFIG_CONTROL_DESIGN'));
ENDSEC;
DATA;
#1=DIRECTION('', (0.E0, -1.E0, 0.E0));
#2=VECTOR('', #1, 2.5E2);
#3=CARTESIAN_POINT('', (0.E0, 0.E0, 0.E0));
.....
#155=ADVANCED_BREP_SHAPE_REPRESENTATION('', (#150), #154);
#162=PRODUCT_DEFINITION('design', '', #161, #158);
#163=PRODUCT_DEFINITION_SHAPE('', 'SHAPE FOR CUBE.', #162);
#164=SHAPE_DEFINITION_REPRESENTATION(#163, #155);
ENDSEC;
END-ISO-10303-21;

```

Figure 4.10. Typical Structure of a STEP file

The following schema and its instances in a physical file (i.e. a STEP file created by Pro/ENGINEER) shows how a geometric entity can be defined in EXPRESS and how instances might be included in the physical file;

```
*)  
ENTITY point;  
    x_coordinate: real;  
    y_coordinate: real;  
    z_coordinate: real;  
END_ENTITY;  
(*
```

Actual instance of this schema on the physical file could appear as follows:

```
#11 = POINT(1.0,1.0,1.0);
```

The following example shows a portion of a STEP file to exchange a rectangle data composed of four lines. In that example, the entity RECTANGLE is defined by four line entities. Since the LINE entities are embedded in the RECTANGLE, they are dependent on the RECTANGLE.

```
#11 = RECTANGLE  
SCOPE  
#12 = POINT(0,0,0);  
#13 = POINT(20,0,0);  
#14 = POINT(20,10,0);  
#15 = POINT(0,10,0);  
ENDSCOPE  
(LINE(#12, #13), LINE(#13,#14), LINE(#14, #15), LINE(#15, #12));
```

4.6. 3D PRISMATIC PART DEFINITION IN THE STEP STANDARD

This section is devoted to definition of a simple 3D prismatic part (i.e. a cube) in the STEP through a simple example part illustrated in Figure 4.11. The cube has been designed on the Pro/ENGINEER software. A STEP file is then generated based on B-Rep modeling scheme and this is declared at the end of the data section in the physical STEP file (whose extension is STP) as a statement;

“#155=ADVANCED_BREP_SHAPE_REPRESENTATION('',(#150),#154);”, as shown in Figure 4.11. Similarly, a STEP file based on wireframe modeling can be recognised by;

“#...=GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION('',(#...),#);”,

and, a STEP file based on surface modeling is conceived by;

“#...=GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION('',(#...),#);”.

Actually, both a solid model STEP file and surface model STEP file includes the wireframe STEP file.

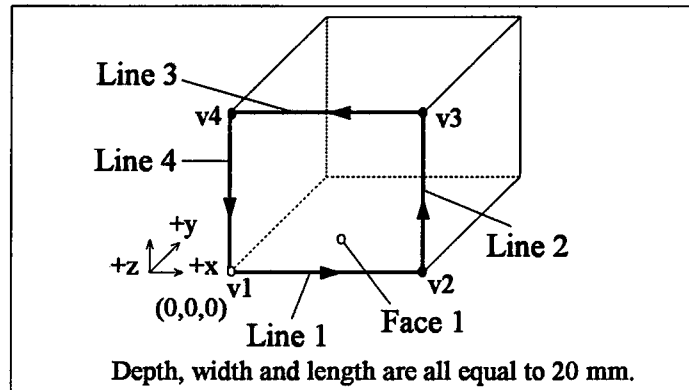


Figure 4.11. An Example Part

The example part has 6 faces, 12 edges and 8 vertices, and a global coordinate system for the part defined and located at the left bottom corner. In *B-Rep modeling scheme*, a face can be represented by edges, vertices (points) and a surface normal. The topology of the face is represented by an edge loop. Each edge consists of two vertices and each vertex corresponds to a point. Edge loops are directed in counter-clockwise direction. The complete STEP files, even for a simple cube in the example, is too long. Therefore, here only the sections related to definition of *Face 1* (front face) of the cube will be shown. At the beginning of data section, in all of the three modeling cases, the part contour is described as if it is wireframe by the use of Direction, Vector and Cartesian_Point statements. For example, Line 1 of Face 1 on the cube shown in Figure 4.11 is described as follows;

```
#1=DIRECTION('',(1,0,0));
#2=VECTOR('',#1,20);
#3=CARTESIAN_POINT('',(0,0,0));
#4=LINE('',#3,#2);
```

#1 points out that direction of the vector to be defined lies in + x direction. #2 states that the vector has a magnitude of 20 mm and its direction is already defined in #1. A

cartesian point is described with respect to global coordinate system by #3. Then, a line constructed using the cartesian point being the starting point of the vector and the vector specified in #3 and #2, respectively. Similarly, Line 2, Line 3 and Line 4 of the Face 1 is represented by following blocks;

```
#5=DIRECTION('', (0,0,1));
#6=VECTOR('', #5,20);
#7=CARTESIAN_POINT('', (20,0,0));
#8=LINE('', #7, #6);
#9 =DIRECTION('', (-1,0,0));
#10=VECTOR('', #9,20);
#11=CARTESIAN_POINT('', (20,0,20));
#12=LINE('', #11, #10);
#13 =DIRECTION('', (0,0,-1));
#14=VECTOR('', #13,20);
#15=CARTESIAN_POINT('', (0,0,20));
#16=LINE('', #15, #14);
```

These blocks including four statements are repeated for all lines (note that these line definitions should not be confused with that of edge definitions) on the part whatever the part representation scheme is. From this point onwards, the three types of representations differ from each other. For example, for the wireframe modeling this is enough to represent the part. In the surface modeling; planes, curves and bounded surfaces with these curves are described in the STEP file. However, in the case of B-Rep modeling scheme; vertices, edges, edge loops and faces are defined in the next sections of the STEP file. Since, B-Rep modeling scheme is used in this thesis, here only the examples related to B-Rep based STEP file is included.

Since the cartesian points have been already described in the STEP file, vertices can reference them by giving the identifiers (unique entity numbers) of points. For example, the corresponding vertices on the Face 1 are defined by;

```
#62= VERTEX_POINT(#3); /* v1 */
#64= VERTEX_POINT(#7); /* v2 */
#66= VERTEX_POINT(#11); /* v3 */
#68= VERTEX_POINT(#15); /* v4 */
```

Next comes the edge definition. Before defining an edge, associated geometric curves are required to be defined in order to locate the edge in a coordinate space. In

this case, the curve for all edges is line type. A line is an unbounded curve defined by a point and a direction. The positive direction of the line is in the direction of the direction vector. Since a line uses a direction vector it should be defined before the associated line. The edge curves are then defined referencing to the associated vertices and lines as shown in the following statements;

```
#134=EDGE_CURVE(#62,#64,#4); /*Edge1 referencing vertices & line*/
#183=EDGE_CURVE(#64,#66,#8); /*Edge2 referencing vertices & line*/
#158=EDGE_CURVE(#66,#68,#12); /*Edge3 referencing vertices & line*/
#186=EDGE_CURVE(#68,#62,#16); /*Edge4 referencing vertices & line*/
```

Next stage is the definition of a geometric surface which may be associated with a face followed by the definition of edge loops and faces. A plane surface is defined by a point on the surface and the normal direction to the surface. A point and two orthogonal axes with the normalised Z axis directed away from the surface and X axis lying on the surface are defined to locate and orientate the plane surface in the space. A plane face entity is represented by a point and an axis, **AXIS2_PLACEMENT** entity. The surface definition in the physical STEP file for the *Face 1* is defined as;

```
#177=CARTESIAN_POINT('',(0,0,0));
#178=DIRECTION('',(1,0,0));
#179=DIRECTION('',(0,0,1));
#180=AXIS2_PLACEMENT_3D('',#177,#178,#179);
#181=PLANE('',#180);
```

A plane face is represented by its bounding edge loops. Each loop runs counter-clockwise around the face. Each face is defined by outer bound, bounds and face surface. Outer bound is outside of the face. Face 1 on the cube is defined as follows in the physical STEP file;

```
#182=ORIENTED_EDGE('',#134);
#184=ORIENTED_EDGE('',#183);
#185=ORIENTED_EDGE('',#158);
#187=ORIENTED_EDGE('',#186);
#188=EDGE_LOOP('',#182,#184,#185,#187);
#189=FACE_OUTER_BOUND('',#188);
#190=ADVANCED_FACE('',#189,#181);
```

Some simple features such as holes, steps, slots, grooves, etc. can also be defined in STEP standard using feature volume definition tools which includes *feature sweep*. The feature sweep is a procedural definition of a 2^{1/2} dimensional shape consisting of a planar profile and a path along which the profile is to be swept and shape definitions for either or both ends.

4.7. PRODUCT MODELLING ENVIRONMENT USED IN THIS THESIS:

Pro/ENGINEER

Pro/ENGINEER marketed by PTC (Parametric Technology Corp.) is a professional software providing engineers with a unique approach to the automation of mechanical design based on a parametric and feature-based modeling technology. The parametric and feature-based capabilities of Pro/ENGINEER makes it easy and flexible to use. In addition to this, the unique data structure of the software provides full associativity among all engineering disciplines. These capabilities allow companies to develop their products and manufacturing processes concurrently and enable them to evaluate design alternatives easily. All these result in better designed products, faster production and lower cost. The basic functionality of Pro/ENGINEER provides the following capabilities; part design, assembly design, design documentation and so on. This basic functionality of Pro/ENGINEER can be expanded by adding specialised modules that allow users to tailor a Pro/ENGINEER configuration to suit their requirements. Pro/ENGINEER has currently offered more than 40 additional modules that may be purchased in application packages or individually. Some of these additional modules and their functions are as follows;

- Pro/ASSEMBLY which is a parametric assembly management system,
- Pro/SHEETMETAL which is used to develop sheet model,
- Pro/FEM-POST which provides FEM post-processing capabilities

Pro/ENGINEER runs on all of the major UNIX and Windows NT platforms (workstations) as well as PCs. The cost of a complete package can be described with hundred-thousand dollars. Therefore, individual purchase of the modules according to

customer's need is more common. Pro/ENGINEER can also be purchased with educational purposes at a reduced cost for certain time intervals like one-year.

Pro/INTERFACE module of the Pro/ENGINEER provides an extensive set of industry standard data exchange formats for the seamless transfer of data between Pro/ENGINEER and other CAD/CAM software programs. The user can use the translators prepared for the different standards found in this module to import and export geometric and non-geometric information of product including solid models. It can import and export data from the following standards; IGES, SET, DXF, VDAFS and STEP. There are other specialised formats like SLA supported in this module, for use in the *stereolithography* applications.

Since STEP standard is the latest and most comprehensive communication standard in the CAD/CAM world, Pro/ENGINEER is equipped with Pro/STEP-INTERFACE module enabling a designer to import product data into, and export product data from Pro/ENGINEER environment via ISO 10303, commonly called STEP standard. This module provides the end users a new way to exchange product model data. The STEP allows designers to exchange more complete data sets than they could with traditional data transfer modes like IGES, DXF, SET and VDAFS which only allow the transfer of geometric data.

The transfer of data between dissimilar CAD, CAE, CAPP and CAM systems is particularly important in today's concurrent engineering work environments where widespread knowledge of design and manufacturing changes are critical. With Pro/STEP-INTERFACE module, product data can be imported from and exported to any CAD/CAM system that supports STEP. This allows companies and their suppliers to share data as needed, regardless of their location or in-house CAD/CAM system [145].

Pro/STEP-INTERFACE supports both the DIS (Draft International Standard) and IS (International Standard) formats for the Level One (file exchange) implementation of *Application Protocol (AP) 203*, the protocol that addresses configuration controlled 3D designs of mechanical parts and assemblies. It will

recognise whichever standard is used immediately upon importing the product data. Similarly, users can choose between exporting product data in either the DIS or IS format. The following conformance classes (cc) of the AP 203 standards are supported by the Pro/STEP-INTERFACE [145];

- Advanced boundary representation solid (cc 6)
- Manifold surfaces with topology (cc 4)
- Surface-and-wireframe models (cc 2)

4.8. FRS-PP: A FEATURE RECOGNISER FOR PRISMATIC PARTS

In CAD/CAM integration, Computer Aided Process Planning (CAPP) system plays a key role between design and manufacturing functions. In a typical CAD/CAPP/CAM integration, there are two basic interfaces to be established; CAD/CAPP interface and CAPP/CAM interface. The former is accomplished by a “*Feature Recogniser*”, whereas the latter is achieved by an “NC Part Programme Generator”. A *feature* is sometimes described as *any geometric form or entity that is used in one or more design/manufacturing activities*. Webster's Ninth New Collegiate Dictionary defines a feature of an object as a prominent part or characteristic of the object [146]. More specifically, a feature is defined as *a set of adjacent faces of a part that correspond to shapes which commonly occur on the surface of the machined parts, like pockets, holes and protrusions, etc.* Features can generally be classified into three categories; design features, machining features and geometric features. A *design feature* (e.g., a cylinder) is a direct representation of the way in which a part is designed. The *machining features* (e.g., a hole) can be derived from the design features. They can be identified from the volume of the material that is to be removed to obtain the final product. *Geometric features* (geometric entities) include surface, curves and points. Feature recognition is defined as grouping of the set of faces on the surface of a part such that each set of face corresponds to a feature. The recognition of features involves identification of higher level features (pockets, holes, etc.) from a set of lower level features such as surfaces, edges and vertices. In other words, the function of a feature recogniser is to translate CAD information into a CAM language.

The common methods that have been applied to recognise the features are [15]; Syntactic pattern recognition, state transition diagrams, volume decomposition approach, set theoretic (Constructive Solid Geometry) approach, graph based approach and external access directions approach.

In this work, a graph-based feature recognition system called FRS-PP (Feature Recognition System for Prismatic Parts) is developed for recognising the features that can be found on prismatic parts and can be machined on a VMC [147]. The input to the system is a B-Rep data file which is obtained by post-processing the STEP file created for a component modeled on a solid modeling system. The post-processor was prepared within a MSc. study recently finished at the University of Gaziantep [148]. The B-Rep file includes all information (like coordinates, face numbers, vertex numbers, etc.) of the prismatic component. Using this file, the system then determines the orientation of each face. The relation between adjacent faces are also found, and these relations are stored in a relation matrix. By tracing the elements of this matrix, features are searched first and then by comparing these with the ones existing in data base, feature types are determined.

4.8.1. Methodology Used in the FRS-PP via an Example

The method used in this work for the feature extraction is primarily based on the principle of *concave decomposition of edges*. The basic heuristic used is that “root faces of a form-feature could be extracted by decomposing the object at concave edges”. Another heuristic similarly implies that “a face whose all neighbouring faces are at convex angle does not form part of a feature”. The boundary faces of the feature are also recognised as well as root faces. Once the boundary and root faces associated with each feature on the object are identified, it is not too difficult to determine titles and characteristics of the features. The input to the system is a B-Rep data base. An example part and its B-Rep information (just for faces 1,2 and 3) are given in Figure 4.12 and Table 4.4, respectively. Whole B-Rep information of the example part can be found in *Appendix A*. The B-Rep data base of the example part includes the topology for faces, edges, vertices and loops. The parameters ($v1x$, $v1y$, $v1z$) and ($v2x$, $v2y$, $v2z$) define vertices of the edges. However, in the case of holes

(i.e. Face No:1), ($v1x, v1y, v1z$) represents coordinates of centerline of the hole at the top surface; whereas $v2x$ defines diameter, $v2y$ stands for the length, and finally the $v2z$ specifies face attributes of the hole. The next step is the determination of physical meanings of the object. For this purpose a special interface program *whose working principle based on view directions and clock wise edge looping* (Table 4.5) was prepared. The left bottom vertices of the faces are always used as a starting point for clockwise loops. The program determines the type and orientation of each face using the B-Rep data base including all topological and geometrical information for the object and the B-rep data base is modified accordingly.

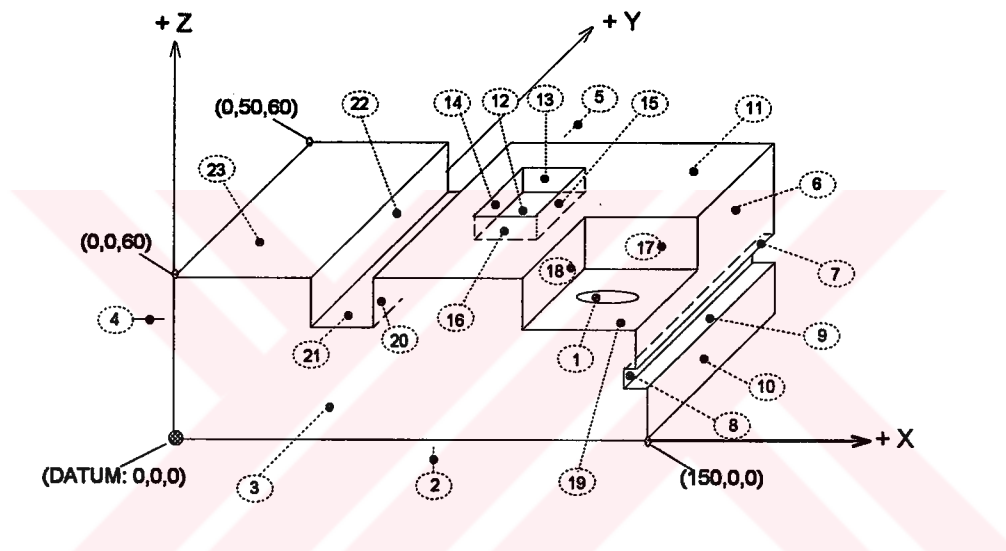


Figure 4.12. Example Part and Its Faces

Geometrical attributes of faces and edges on the object, and relational topology which is basically concerned with the adjacency relationships between the faces and edges are then determined. The face and edge attributes used in this work are tabulated in Table 4.6 and Table 4.7, respectively. These attributes and relations between all faces and edges of the object are converted into a *relation matrix*. The relation matrix derived for the example part is given in Table 4.8.

The feature extraction session is performed by scanning this matrix starting from the first row. The first negative entry starts the feature extraction cycle. There are two alternatives; a negative entry in an off-diagonal cell and a negative entry in a diagonal

cell. In the first case (the negative entry in a off-diagonal cell (i,j)), the faces F_i and F_j are defined as the root faces of the current (candidate) feature. Any of column for face having an off-diagonal (-1) entry in the i th row is also flagged as the root faces, while columns for face having an off-diagonal (1) entry in the i th row is also flagged as the boundary faces. The cycle is finished when no new root face is found. As an example, for the slot as shown in Figure 4.12, the root faces are found as; 20, 21 and 22, whereas the boundary faces found are 3, 5, 11, and 23.

Table 4.4. A Section of B-Rep Data Base of the Example Part

Face No	Edge No	v1x	v1y	v1z	v2x	v2y	v2z
1	0	105	10	50	15	50	-1
2	1	0	0	0	0	50	0
2	2	0	50	0	150	50	0
2	3	150	50	0	150	0	0
2	4	150	0	0	0	0	0
3	1	0	0	0	0	0	60
3	2	0	0	60	30	0	60
3	3	30	0	60	30	0	45
3	4	30	0	45	50	0	45
3	5	50	0	45	50	0	60
3	6	50	0	60	80	0	60
3	7	80	0	60	80	0	50
3	8	80	0	50	150	0	50
3	9	150	0	50	150	0	30
3	10	150	0	30	140	0	30
3	11	140	0	30	140	0	20
3	12	140	0	20	150	0	20
3	13	150	0	20	150	0	0
3	14	150	0	0	0	0	0

Table 4.5. The Key Words for Determining the Type and Orientation of Faces

Plane	View Directions	Starting Point for CW-Loop
XY	From - Z	Left-bottom vertice
ZY	From - X	Left-bottom vertice
ZX	From + Y	Left-bottom vertice

Table 4.6. Face Attributes

Type of Face	Attributes
Plane	0
Convex	1
Concave	-1

Table 4.7. Edge Attributes

Type of Edge	Attributes
Convex	1
Concave	-1
Non-adjacent	2

In the second case, the negative entry in a diagonal cell indicates the presence of a concave face like a hole. However, it is applicable if and only if when there are no zero and negative entries in the row corresponding to cell. Face 1 shown in Figure 4.12 constitutes a feature (i.e., hole) itself. The root and boundary faces of features are successfully determined at the end of the feature extraction cycle.

Having extracted the features, they are further identified by matching them with entries in the previously described feature data base, by using a similar approach used in Reference [149]. The view and list of recognised features together with their codes are given in Figure 4.13 and Table 4.9, respectively. Note that, the length of the codes is variable. The code length is $3(n+1)$, where n is the number of the root faces of a feature. The code digits are grouped as 3 by 3. The first digit in the first group is the *genus* (g) of the open object formed by the root faces and it is calculated by the well known *Euler's formula*. The second and third digits in the first group stands for the boundary faces and total number of convex adjacencies, respectively. The first, second and third digits in all subsequent groups denotes face attributes of the root faces, number of concave boundary edges, and number of convex boundary edges.

In this study, part features are denoted by simple abbreviations as shown in Table 4.10. Geometrical representation of these features and other types of features that can be added to the system with small modifications can be found in *Appendix B*. The final form of output of the feature recognition system is given in Table 4.11. The $d-x$, $d-y$, $d-z$, d , w , and l are all dimensional parameters adequate to generate machining instructions.

Table 4.8. Relation Matrix Derived for the Example Part

RELATION MATRIX																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	-1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	2	2	2
2	1	0	1	1	1	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2
3	2	1	0	1	2	1	1	1	1	1	1	2	2	2	2	2	2	1	1	1	1	1	1
4	2	1	1	0	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1
5	2	1	2	1	0	1	1	1	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1
6	2	2	1	2	1	0	1	2	2	2	1	2	2	2	2	2	1	2	1	2	2	2	2
7	2	2	1	2	1	1	0	-1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
8	2	2	1	2	1	2	-1	0	-1	2	2	2	2	2	2	2	2	2	2	2	2	2	2
9	2	2	1	2	1	2	2	-1	0	1	2	2	2	2	2	2	2	2	2	2	2	2	2
10	2	1	1	2	1	2	2	2	1	0	2	2	2	2	2	2	2	2	2	2	2	2	2
11	2	2	1	2	1	1	2	2	2	2	0	2	1	1	1	1	1	1	2	1	2	2	2
12	2	2	2	2	2	2	2	2	2	2	2	0	-1	-1	-1	-1	2	2	2	2	2	2	2
13	2	2	2	2	2	2	2	2	2	2	1	-1	0	-1	-1	2	2	2	2	2	2	2	2
14	2	2	2	2	2	2	2	2	2	2	1	-1	-1	0	2	-1	2	2	2	2	2	2	2
15	2	2	2	2	2	2	2	2	2	2	1	-1	-1	2	0	-1	2	2	2	2	2	2	2
16	2	2	2	2	2	2	2	2	2	2	1	-1	2	-1	-1	0	2	2	2	2	2	2	2
17	2	2	2	2	2	1	2	2	2	2	1	2	2	2	2	2	0	-1	-1	2	2	2	2
18	2	2	1	2	2	2	2	2	2	2	1	2	2	2	2	2	-1	0	-1	2	2	2	2
19	1	2	1	2	2	1	2	2	2	2	2	2	2	2	2	2	-1	-1	0	2	2	2	2
20	2	2	1	2	1	2	2	2	2	2	1	2	2	2	2	2	2	2	2	0	-1	2	2
21	2	2	1	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	-1	0	-1	2
22	2	2	1	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	-1	0	1
23	2	2	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	0

Table 4.9. The List of Recognised Features and Their Codes

RECOGNISED FEATURES								
1	Face	Recognised by geometric reasoning						
2	Rectangular Pocket	0	1	0	0	3	1	0 3 1 0 3 0
3	Thru Slot	0	4	0	0	2	2	0 1 3 0 1 3
4	Thru Slot	0	4	0	0	2	2	0 1 3 0 1 3
5	Blind Step	0	3	0	0	2	2	0 2 2 0 2 2
6	Thru Hole	1	2	0	0	2	2	0 2 2 0 2 2

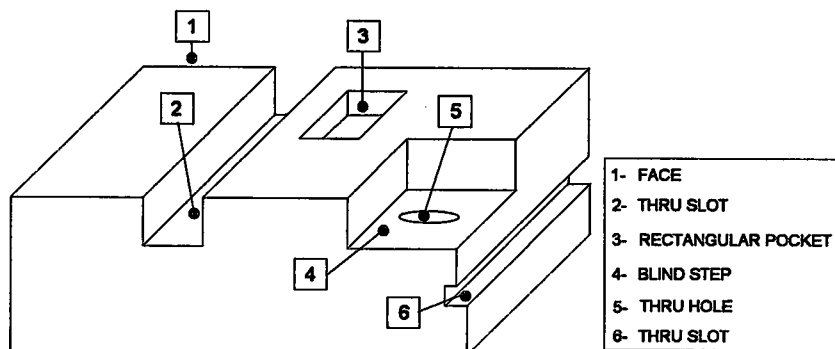


Figure 4.13. The View of Recognised Features

Table 4.10. Features and Their Acronyms Used in This Thesis

No	Feature	Acronym
1	Face	FACE
2	Thru Hole	T-HOL
3	Flat-Bottomed Blind Hole	FBB-HOL
4	Point-Bottomed Blind Hole	PBB-HOL
5	Thru Counter Bore	TC-BOR
6	Flat-Bottomed Counter Bore	FBC-BOR
7	Point-Bottomed Counter Bore	PBC-BOR
8	T-Slot	TSLOT
9	Thru-Step	T-STP
10	Thru Slot	T-SLT
11	Cylindrical Pocket	C-PKT
12	Rectangular Pocket	R-PKT
13	Rectangular Hole	R-HOL
14	Face Slot	F-SLT
15	Edge Pocket	E-PKT
16	Blind Slot	B-SLT
17	Blind Step	B-STP
18	Prismatic Boss on Prismatic Base	PB-PB
19	Prismatic Boss on Cylindrical Base	PB-CB
20	Cylindrical Boss on Prismatic Base	CB-PB
21	Cylindrical Boss on Cylindrical Base	CB-CB
22	Edge Cut	E-CUT
23	Corner Edge Cut	CE-CUT

Table 4.11. Output From the Feature Recogniser

No	Feature ID	Distance to main datum			Feature Attributes			Approach Directions						SR	FR
		d-x	d-y	d-z	d	w	l	1	2	3	4	5	6	Surface Rough.	Fillet Radius
1	R-PKT	50	30	53	7	10	50	1	0	0	0	0	0	-	-
2	T-SLT	140	0	20	10	10	50	0	0	1	0	0	0	-	-
3	FACE	0	0	60	-	50	150	1	0	0	0	0	0	-	-
4	T-HOL	105	10	50	15	50	0	1	0	0	1	0	0	1.7	-
5	T-SLT	30	0	45	15	20	50	1	0	0	0	0	0	-	-
6	B-STP	80	0	50	10	20	70	1	0	1	0	1	0	-	-

4.9. VALIDATION OF B-REP MODELS BY USING EULER'S LAW

Validity of a polyhedral object based on the B-Rep modeling scheme can be ensured by Euler's law. Euler's law states that a *polyhedron* is topologically valid if the following equation is satisfied;

$$F - E + V = 2 \dots\dots\dots (4.1)$$

where F, E and V are the number of faces, edges and vertices, respectively. This equation can be used for simple polyhedral objects like tetrahedron, cube, etc., without any passageways and holes. For example, a cube has 6 faces, 12 edges and 8 vertices. This satisfies the Equation. 4.1. However, Equation. 4.1 cannot be used for those polyhedral objects with passageways and holes. For these objects the following Generalised version of Euler's formula could be used;

$$F - E + V - L = 2 (B - G) \dots\dots\dots (4.2)$$

where; B is the number of bodies, G is the number of geniuses like holes, and L is the number of inner loops to be found in the faces. It is well worth to point out here that it is not so easy to apply Euler's law for objects with curved surfaces, cylinders, cones and spheres. For such cases, the structure of the curved objects have to be defined. Consider the example part shown in Figure 4.12. Assuming that the through-hole on the blind step is approximated by a cubic hole, the component has 26 faces (F), 69 edges (E), 46 vertices (V), 3 inner loops (L) (two loops for the through hole and one loop for face pocket), 1 body (B) and 1 genus (G) (hole). Euler's law is satisfied by $(0 = 0)$, and therefore the part model is valid.

Since the Euler's formula is weak for the curved surfaces (i.e. for holes or cylinders), a different type of definition can be assumed for the structure of these objects in order to satisfy Euler's law. For example, instead of using a cubic hole representation of holes, using an ambiguous representation for the hole with 2 vertices, 3 edges and 1 face, results in $F = 23$, $E = 60$, $V = 40$, $L = 3$, $G = 1$, $B = 1$. When these values are substituted, the Euler's law is also satisfied.

CHAPTER V

PROCESS PLANNING MODULES

5.1. INTRODUCTION

In this chapter, process planning modules prepared in this thesis are discussed with practical and illustrative examples. Section 5.2 is devoted to a brief review of the process planning including its structure, definitions and implementation tools. In the next sections, the methodology used in the development of the process planning modules, namely, “*workpiece size selection module*”, “*operation selection and sequencing module*”, “*tooling selection module*”, and “*set-up selection module*”, are given without considering any special optimisation tool or strategy.

5.2. COMPUTER AIDED PROCESS PLANNING

Computer Aided Process Planning (CAPP) consists of determining the processes required to produce the part and putting those processes in a suitable sequence. Many different process sequences may be obtained for a part, taking some criteria into account like machinability and safety. Although the ultimate goal in process planning is to decrease total production time, the primary objective is to provide a suitable way to produce the most functional parts with high quality in an optimum time.

5.2.1. Approaches to Process Planning

There are two basic approaches to the automated process planning [5,6,150]; one being the *variant process planning approach* and the other is the *generative process planning approach*. Schematic view of CAPP approaches and implementation techniques used for process planning are illustrated in Figure 5.1.

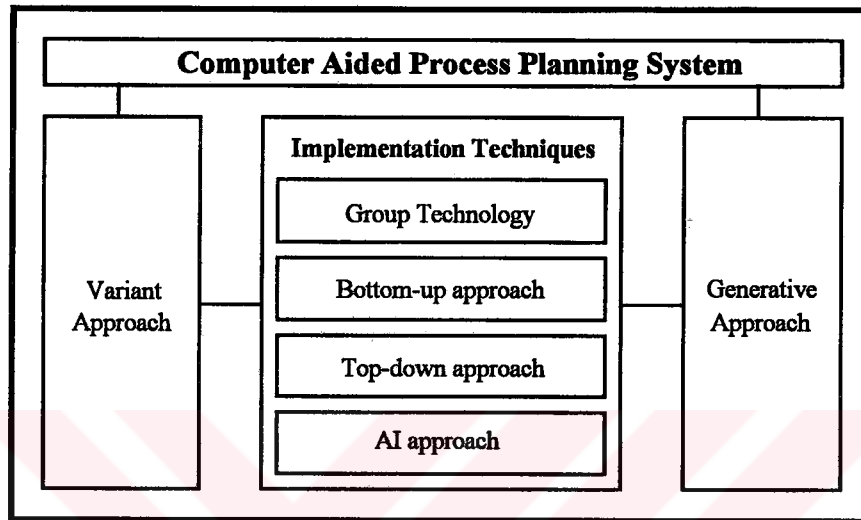


Figure 5.1. CAPP Approaches and Implementation Techniques

The variant approach is the earliest work in applying computers to aid the process planning. In this type of process planning approach; parts are grouped into a number of part families, characterised by similarities in manufacturing methods and thus related to group technology [8]. The group technology application means that the existing components are coded, classified and subsequently grouped into families by answering a number of pre-defined questions. For each group, a unique code is generated and a standard process plan is developed beforehand for each family. Process plan for a new part is created by identifying and retrieving a related standard plans from the computer. Since this plan contains knowledge for all parts in this family, some editing may be required for a specific part. Often a little editing is required, because a new plan is a variant of the standard process plan. As a result considerable time is saved and the development and hardware costs are lowered.

A generative CAPP system will automatically generate a process plan from engineering specifications of the finished part [151]. In generative process planning system, process plans are created from information available in manufacturing data base without any (or little) human intervention. Required data is obtained from the drawing of the part. After receiving the manufacturing data base, the system can identify the required operations and their sequence. Machine tool selection, cutting tool selection and determination of cutting conditions are usual planning activities of the generative approaches.

The variant approach is much simpler than the generative approach, but the system using this type of approach has an interactive nature. The generative planning approach has the following advantages; process plan is directly generated and new components can be planned without any intervention. However, it requires a large data base and a complex algorithm. A computer having a reasonably high speed and large memory capacity is needed for the generative approach.

5.2.2. Implementation Techniques of CAPP

There are four most popular types of implementation techniques which are;

- Group Technology (GT),
- Bottom-up approach,
- Top-down approach,
- Artificial Intelligence (AI) approach.

Group Technology is a manufacturing concept that seeks to take advantage of the design and processing similarities among parts. It involves categorising parts and recording them in files or catalogues; designs are retrieved, as needed. Similar parts are arranged into part families. The parts are coded and classified by the use of available classification and coding systems. *Bottom-up approach* develops process plans by means of tracing the task of process planning from the finished part to raw material. It is suitable for the methodology of the variant approach. *Top-down approach* develops process plans by means of tracing from the raw material to the finished product. This is

an automated process planning method which is suitable for methodology of the generative approach.

Expert systems or in other words *Knowledge Based Systems* are the first generation techniques using *Artificial Intelligence* and they would seem to be a promising approach in the automation of process planning. Because, the number of rules is considerable in this approach, expert systems produce process plans by using knowledge base. There are various ways in which expert systems can solve the problem of determining sequence of operations by using only the part geometry and possibilities in the machine shop. The direction of working in expert systems can be forward, backward or bi-directional. This refers to the possibility of starting with the problem that is posed and ending with the solution or answer, or reverse. Process planning, generally starts with *forward chaining*, uses search rules for the problem until the goal is reached. The rules are generally of the type “if <antecedent> then <consequent>” or condition- action, called *production rules* [5]. In this approach, the planning problem is divided into sub-problems. However, sub-problems are often interdependent which means that the interaction has to be handled by adding constraints to the sub-part solutions found by introducing new production rules [151]. The system can provide explanations if required by backtracking the rules to reach the results.

Knowledge base build-up refers to extracting, gathering, classifying and storing all information required for process planning. Some of the information required by the knowledge base are the type of machines available, their capacity that requires maximum operational dimensions and power, tool, etc. The knowledge about the part are generally described in the part definition data file, *for process selection, machine tool selection, process sequence and cutting tool selection*, is acquired from data base and several experts in machining domain [151]. There are basically two approaches, namely; *rule based systems* and *frame based systems* in terms of the knowledge representation. In case of *rule-based process planning systems*, process planning activities are determined by previously defined rules that are developed from the geometrical and engineering requirements. The term “*rule based*” refers to the rules utilised by process planners when generating process plans. The rules will help to

determine the functions of process planning. A rule based system makes use of an “if <antecedent> then <consequent>” representation of knowledge. Frame-based systems are based upon the construction of pyramid-like networks of know-how, which is referred to as a knowledge hierarchy. The knowledge base of a frame based system consists of a knowledge sketch and is characterised by three basic primitives; entities, attributes, relationships.

Process planning system developed in this work uses a generative approach. It is called an *Optimised Process Planning System for Prismatic Parts* and abbreviated with OPPS-PRI throughout this manuscript. The process planning segment of the OPPS-PRI is composed of four modules which are prepared for *workpiece size selection, operation selection and sequencing, tooling selection and set-up selection*. Methodology of each process planning module is discussed in the following sections.

5.3. WORKPIECE SIZE SELECTION MODULE

The terms; “*raw material*”, “*raw stock*”, “*blank*”, “*part*”, “*component*”, “*model*” and “*workpiece*” are generally used in a confusing manner. In this work, the term “*raw material*” is used for the materials unless they are pre-machined to pre-determined sizes constituting a raw stock data base. The “*workpiece*” or “*raw stock*” are defined as an intermediate stage between the raw material and finished component. The workpieces or raw stocks are pre-machined from raw materials to certain sizes with which a workpiece (raw stock) data base can be prepared. The “*blank*” is an imaginary rectangular block part which is slightly larger than the cuboid whose dimensions correspond to the maximum width, length and height of the designed component. A machining allowance can be added on top of dimensions of the *cuboid* which is composed of enveloping faces to find the blank dimensions. In some circumstances, the workpiece may be the same as the blank. However, the workpiece sizes may be considered to be slightly larger than the blank sizes. The terms; “*part*” and “*component*”, are general and the implicit names of a model. The “*model*” is description of a part in the design stage. Figure 5.2 shows a typical material flow in process planning.

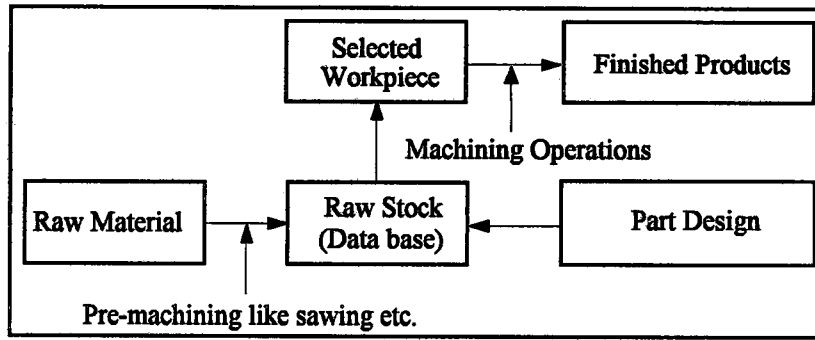


Figure 5.2. Typical Material Flow in Process Planning

Workpiece size selection is the first stage in the process planning. The volume of the material to be removed depends on the size of the selected workpiece. The preparation of the workpiece for a component is vital, since economic production is often dependent on minimising the range of stock held whilst reducing the machining time to a minimum. Typical parameters of raw stock are the cuboidal dimensions and material specification which affect the cutting conditions. These parameters are usually contained in a *raw stock data base* as for the manufacturing environment, and workpiece selection is performed by searching the blank size from this data base. A section of a workpiece data structure used in this thesis is shown in Table 5.1. In this work, it is assumed that the machining allowance can be selected between 0 and 2 mm. Faces of a workpiece can be cold rolled or saw cut. This affects the cutting parameters related to those faces. Therefore, surface information may be taken into consideration for the optimum selection of a workpiece. Pre-machining cost is another factor affecting the total manufacturing cost. However, in this work, these two factors were not included in the selection strategy of the workpieces, since the machining allowance is quite small and the pre-machining cost is neglected. Instead, the emphasis was given to the minimization of waste material due to the selected workpiece. The *workpiece size selection module* is therefore responsible for selecting a suitable workpiece size for the part model from the raw stock data base, while minimizing the waste material.

Table 5.1. A Portion of Raw Stock (Workpiece) Data Base

Code	Material Type	Width (mm)	Length (mm)	Height (mm)
W101	ST-52	130	155	130
W102	ST-52	125	155	130
W103	ST-52	110	155	125

The finished model of the part determines the workpiece size. The workpiece should be as close as possible to the size of the finished part to avoid extra stock removal, as large as possible to enclose the finished part. Initially, the maximum dimensions of the component in three mutually perpendicular axes; X, Y and Z are found by using the coordinate values of the part entities contained in the B-Rep file which is obtained from *feature recognition module*. A machining allowance is then added to the desired surfaces in order to find the *blank dimensions*. Workpiece to be used is then selected by searching the blank size from the raw stock data base. The blank size sometimes may not be found in the data base and it is therefore necessary to select the next larger stock size in mm³. Raw stock data base includes only cuboidal billets, since a large amount of the component is produced from rectangular blocks [39]. Finishing of a cast part, hence a *non-cuboidal raw stock* was not considered in this study.

Figure 5.3 shows an example part. The maximum width, length and height of the designed component are 120 x 150 x 125 (in mm), respectively. If 2 mm machining allowance is used for all directions, the dimension of the blank is equal to 122 x 152 x 127 (in mm). From the available stocks, it is found that the smallest raw stock size is 125 x 155 x 130 (in mm), and W102 (as shown in Table 5.1) is assigned as the workpiece to be used.

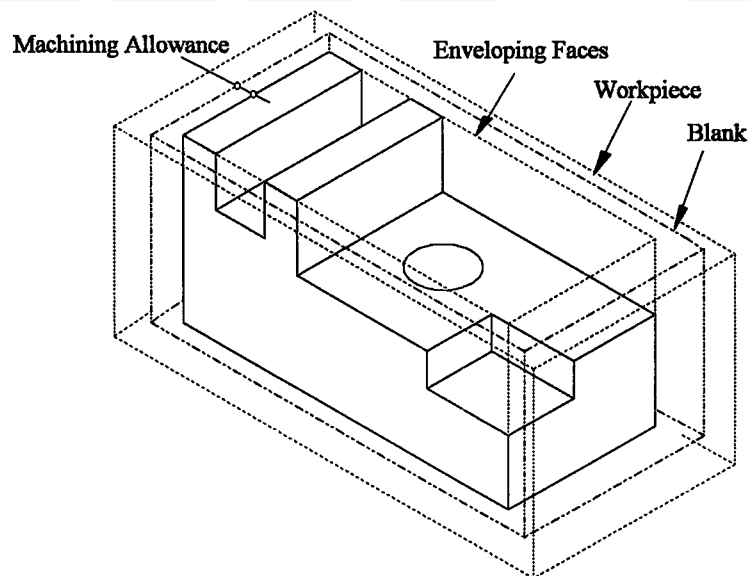


Figure 5.3. Enveloping Faces, Blank and Workpiece Size for An Example Part

5.4. OPERATION SELECTION AND SEQUENCING MODULE

The performance of a CAPP system is mainly based on the *determination of sequence of operations*. The function of *operation sequencing* in the CAD/CAM integration is to provide a technological knowledge about the machining which is further used by the down-stream process planning phases like *tool selection*, etc. Therefore, determination of the sequence of operations is the heart of CAPP, and the efficiency of CAPP systems depends upon the sequence of operations. It should also be noted that optimisation of operation sequences is strictly necessary to satisfy the cost requirements which will be discussed in Chapter 7. The place of “*determination of sequence of operations*” in CAD/CAM integration is illustrated in Figure 5.4.

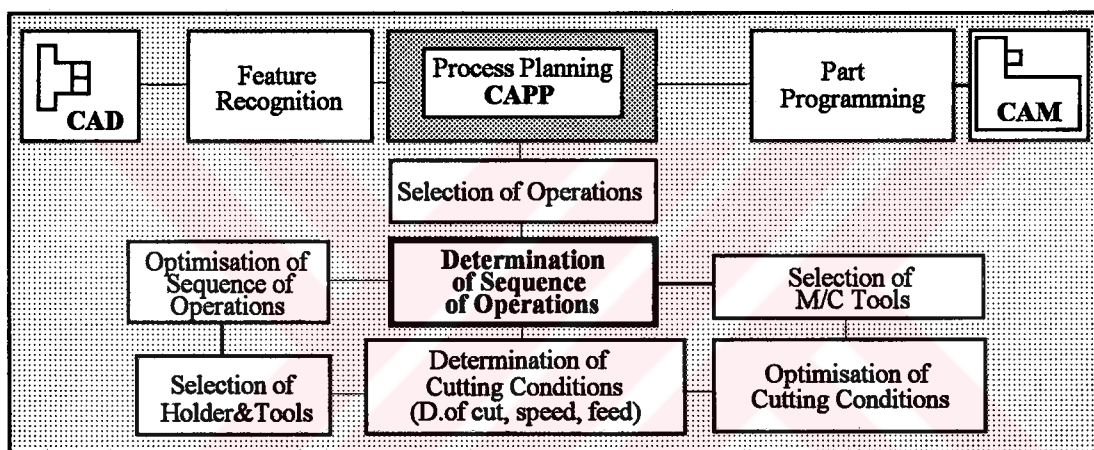


Figure 5.4. Corner Stones in a Typical CAD/CAM Integration

5.4.1. Computer Aided Determination of Sequence of Operations

This section covers the methodology proposed for the *operation sequencing module*. Upon receiving the list of recognised machining features and their characteristic parameters including datum, dimensions and approach directions, the system automatically determines a reasonable process sequence and converts it into a usable process planning sheet. While doing this, the program uses prescribed sequencing rules constructed on the basis of characteristics of features like area, depth and direction, as well as the rules due to machinability. Each factor has its own machining priority value which is further used to calculate total machining priority value for each feature. The sequence of features is determined by putting the features

in a descending order starting from the one which has the highest total machining priority value. The developed operation sequencing system is heuristic based, so yields only reasonable machining orders.

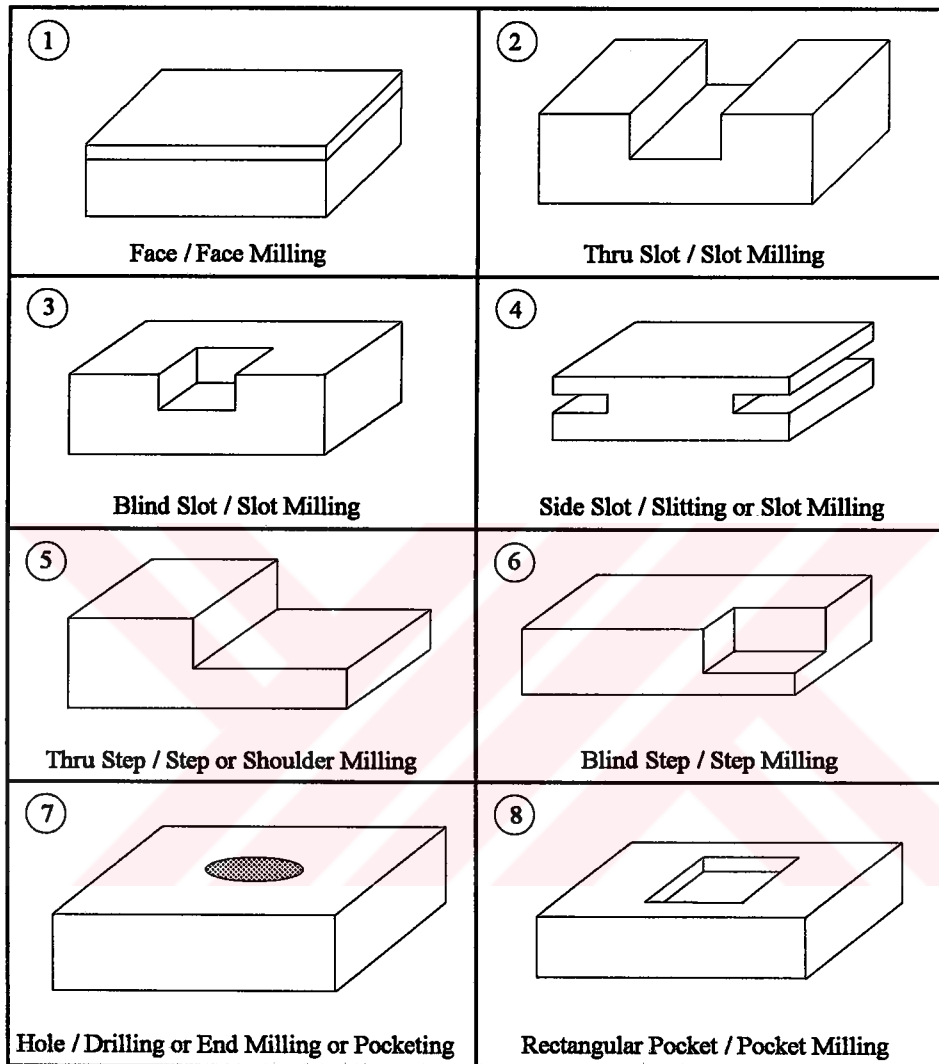


Figure 5.5. Features and Corresponding Milling Operations

5.4.2. Selection of Machining Operations

All of the machining features that are conceivable by the *feature recognition module* can be processed by the operation selection program. Eight of the features which are the most commonly found on prismatic components are illustrated in Figure 5.5. The whole feature spectrum that can be handled both in the feature recognition module and the operation selection/sequencing module can be found in *Appendix B*.

Each feature corresponds to at least one or more particular milling operation(s). For example, a slot feature corresponds to slot milling. A point-bottomed blind hole feature corresponds to twist drilling. A thru-hole corresponds to either twist drilling, end milling, or twist drilling plus reaming according to the value of surface roughness specified in the input data base. The input data base for the sample part shown in Figure 5.6 is given in Table 5.2. The convention for approach directions is illustrated in Figure 5.7. Note that the order of the features given in Table 5.2 is not feasible. It is the order of the features as they were recognised in the feature recognition module.

It is composed of the following information; name of the features, distances from the datum of the features to datum of the part $(0, 0, 0)$ for locating the features exactly on the part (i.e., $d-x$, $d-y$ and $d-z$), feature attributes including dimensions or dimension descriptors of features (i.e., d , w and l), and approach directions.

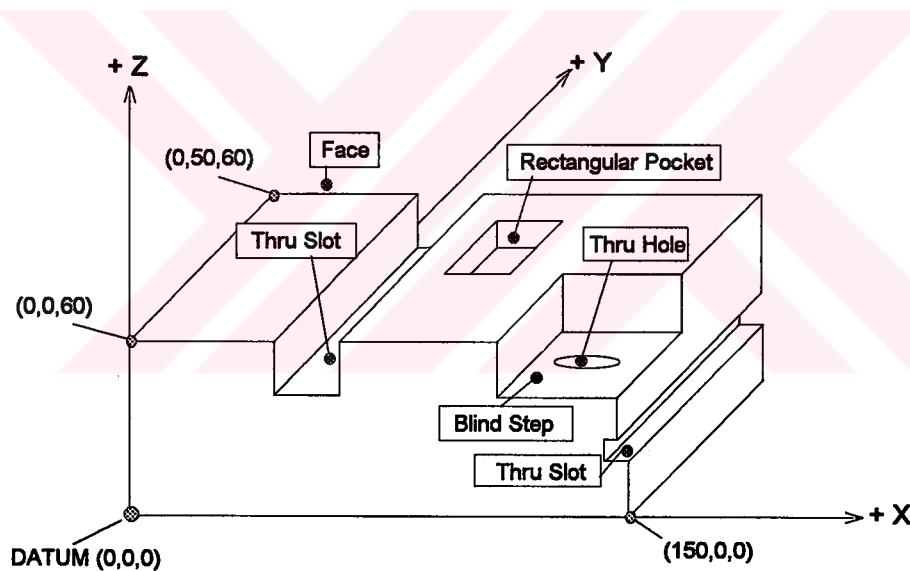


Figure 5.6. Sample Part

Table 5.2. The Input Data for the Sample Part

No	Feature	Distance to main datum			Feature Attributes			Approach Directions						SR	FR
		d-x	d-y	d-z	d	w	l	1	2	3	4	5	6	Surface Rough.	Fillet Radius
1	R-PKT	50	30	53	7	10	50	1	0	0	0	0	0	-	-
2	T-SLT	140	0	20	10	10	50	0	0	1	0	0	0	-	-
3	FACE	0	0	60	5	50	150	1	0	0	0	0	0	-	-
4	T-HOL	105	10	50	15	50	0	1	0	0	1	0	0	1.7	-
5	T-SLT	30	0	45	15	20	50	1	0	0	0	0	0	-	-
6	B-STP	80	0	50	10	20	70	1	0	1	0	1	0	-	-

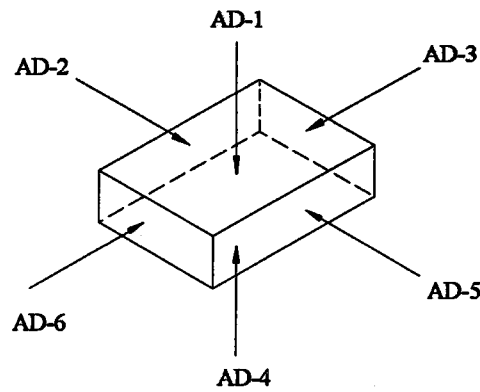


Figure 5.7. Convention for Approach Directions

The sequence of the features with their accompanying machining operations are first determined by using heuristic based *priority values* generated within the system. Therefore, the operation sequencing system is actually a heuristic-based program. The priority value for each *characteristic* inherent in the features is determined by means of *weighting factors*. Some of characteristics that are important in calculating the priority values of features are; *area of the root faces of the features*, *depth of the features from the top* and *access directions*. Of these, the area of root faces of the features is the most important factor in sequencing of features, since the machining of features having a larger surface area than others may facilitate to reach the final part geometry. For instance, the priority value for a feature on the area basis is calculated by multiplying its total area with the related weighting factor defined in the program. The weighting factors are normalised to obtain a scale for ordering and they can be changed; increased or decreased according to characteristics to be stressed.

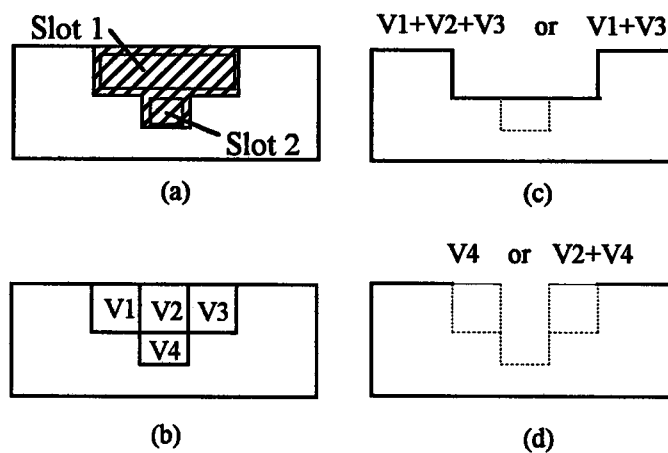


Figure 5.8. Elementary Machined Volumes

The *depth factor* is also an important factor for ordering the features. The idea behind the depth factor is originated from the heuristic is that a deep feature may be reached just after machining some upper features on the component. For example, consider Slot 1 and Slot 2 shown in Figure 5.8(a). As compared with the depth factor, Slot 1 has a greater priority value such that it should be machined before the Slot 2. It is actually not wise to machine the Slot 2 before the Slot 1.

The *direction factor* is used to rate the features based on their locations on the component by means of approach directions. The priorities and total priority values calculated for each of the features on the sample part are shown in Table 5.3. The ordered features with descending priority values are illustrated in Table 5.4.

Table 5.3. Priority Values for Each Characteristic Factors and Their Totals

No	Feature	Factors			Total Priority Value
		Area	Depth	Direction	
1	R-PKT	50	240	160	450
2	B-STP	200	240	160	600
3	FACE	600	400	200	1200
4	T-HOL	25	0	160	185
5	T-SLT	100	320	160	580
6	T-SLT	50	160	120	330

Table 5.4. The Order of Features Based on Priority Values

Seq. No	Feature	Total Priority Value
1	FACE	1200
2	B-STP	600
3	T-SLT	580
4	R-PKT	450
5	T-SLT	330
6	T-HOL	185

As can be seen from Table 5.3 and Table 5.4, face is always the first feature to be machined. Similarly, holes are generally left to the end of the feature list. The order given in Table 5.4 is a heuristic sequence of features, not the sequence of operations. The next step is the determination of the sequence of operations by using the fact that each feature corresponds to at least one or more particular milling operation(s). However, for some features, it is not so easy to name the operations corresponding to

the features. For instance, consider again Figure 5.8(a). Slot 1 and Slot 2 are the two reduced features that are the two parts of the compound feature. To obtain the compound feature in Figure 5.8(a), volumes V1, V2, V3, and V4 must be removed as shown in Figure 5.8(b). Volumes V1, V2 and V3 can be considered as removed part of a first slot. The other volume V4 can be considered as a second slot. The first slot may be cut by removing volumes V1, V2 and V3. Volume V4 might be removed to obtain final part geometry as shown in Figure 5.8(c). Or, the volumes V2 and V4 may be considered as the first slot to be produced. Then, volumes V1 and V3 being step features may be removed to obtain whole geometry as shown in Figure 5.8(d).

Table 5.5. Sequence of Operations Derived From Sequence of Features

Sequence No	Feature	Operation
1	FACE	Face Milling
2	B-STP	Step Milling
3	T-SLT	Slot Milling
4	F-PKT	Pocket Milling
5	T-SLT	Slot Milling
6	T-HOL	Twist Drilling

Using the feature(s)-operation(s) correspondences defined in the program and operation identifiers, the sequence of operation can be derived. For example, the surface roughness value specified for the thru hole in Table 5.2 can be used to determine the operation type for that feature. The rules constructed for selecting the type of operation(s) for a thru-hole are as follows;

if (surface roughness > 1.6) then (assigned operation is twist drilling)
 if (0.4 ≤ surface roughness ≤ 1.6) then (assigned operation is counter sinking)
 if (0.2 ≤ surface roughness < 0.4) then (assigned operation is drilling plus reaming)

The sequence of operation found for the example part shown in Figure 5.6 is tabulated in Table 5.5. This sequence is verified whether the order of operations obey machinability rules through the use of *sequencing rules* constructed by “*if statements*”. A typical machinability rule in pseudo form is as follows;

tapping cannot be performed before pre-machining the hole to be tapped

The grouping of fully similar operations is the final modification of the program. For example, if there exist two thru-holes of the same diameter and if these two holes are required to be machined for the surface roughness of 1.7, the initial operation list suggested as:

1. Drilling (*Hole 1*) - 2. Reaming (*Hole 1*) - 3. Drilling (*Hole 2*) - 4. Reaming (*Hole 2*)
is changed to;

1. Drilling (*Hole 1*) - 2. Drilling (*Hole 2*) - 3. Reaming (*Hole 1*) - 4. Reaming (*Hole 2*)

5.5. TOOL SELECTION MODULE

Tooling has become the heart of manufacturing processes and one of the key steps in the process planning to increase productivity and to decrease the cost of machining while offering a superior quality in the highly competitive manufacturing area. So, effective selection of proper and productive tools for machining operations is vital to realise the objective of the IMS.

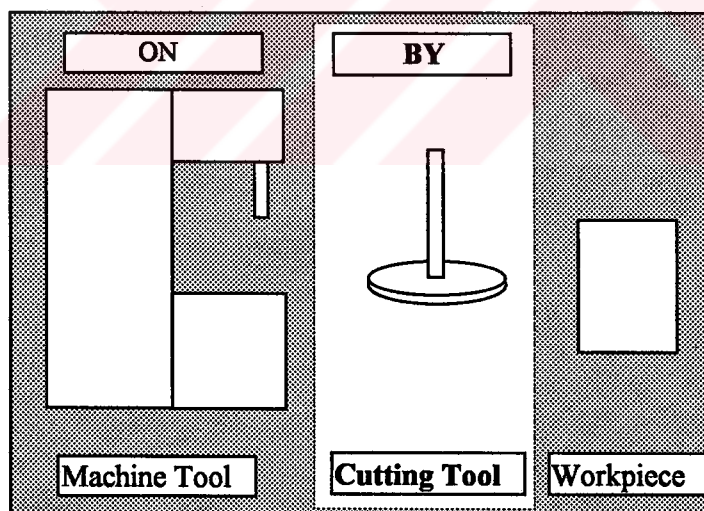


Figure 5.9. The Three Basic Elements of Milling Processes

Many variables exist in a typical milling operation. They are influenced considerably by capabilities of machine tool and cutting tools employed in cutting as well as the workpiece. Any milling operation requires awareness of various limitations such as; *size, shape and strength of workpiece, rigidity, metal removal rate, chucking*

and fixturing, spindle speed and feed rate limitations, available power, holder overhang and cutting tool. Note that all of these parameters are interrelated to each other. For example, the size of a cutting tool is directly effective on the power required for cutting. It is worth emphasising that the *cutting tool* is one of the most effective variables in milling process since it is the most manageable one (i.e., for a certain milling operation there are many alternative cutting tools to be employed). Figure 5.9 shows the three basic elements of a milling operation. Although it can be recognised that this manageability makes *cutting tool selection* easy, unfortunately it is not always true. It is actually very difficult to select the most suitable cutting tool for a particular milling operation that provides the required surface quality, works well with the productive cutting speed, feed rate and depth of cut, and at the same time offers a long tool life and low total cost.

Choosing a cutting tool for a certain milling operation is quite a complicated task. There are actually three predominant parameters to be considered in any cutting tool selection procedure. They are; *cost of cutter, productivity of cutter and quality of machined component.* Each of these factors has its own impact on the total cost of a milling operation. There may often be a need to make a compromise between these three factors. Apart from these, there is another criterion: *sequence of operations* to cut the workpiece. Sometimes, a complicated milling work can even be accomplished with a single tool but machining time would be increased dramatically. However, selecting the most appropriate cutter for each of features on the component will come out with a large tool set resulting in higher number of tool changes in machining. There should also be a compromise between these two aspects by using different optimisation strategies like minimisation of number of tool changes. In order to make such optimisations, alternative tools should be assigned for each of machining operations. Therefore, the tool selection modules of any process planning package should be able to suggest alternative tools for machining operations.

This part of the work in the first place is an attempt to develop a *Cutting Tool Selection System* (CTSS) that can provide alternative cutting tools to be used as candidates which are to be reconsidered in the down-stream optimisation mechanisms

of the process planning system. Cutting tool selection strategy is given with a simplified block diagram shown in Figure 5.10.

Main input to the CTSS is the list of machining features extracted from the feature recognition module. As shown in Table 5.2, this data base includes all the geometrical information of the features. It is also accompanied by the machining operations assigned to the machining features by the operation selection module. In addition to the list of operations, data base of cutting tools, tool holders and materials etc., are also provided. Candidate tools are assigned from the tooling data base for each of the part features (operations) using the tooling selection strategy.

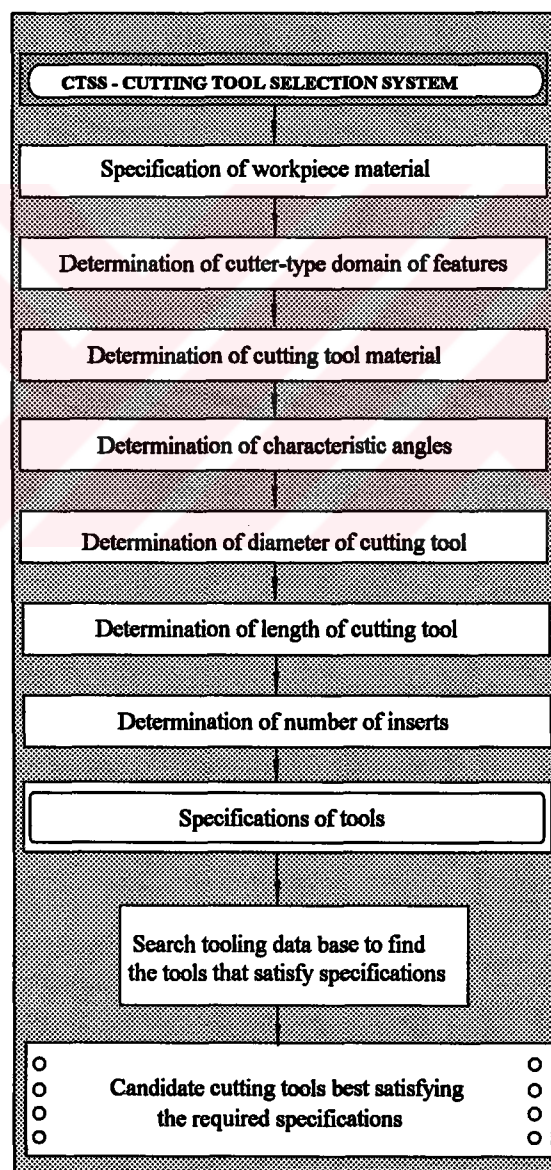


Figure 5.10. Block diagram of CTSS

CTSS offers three types of cutting strategy which are;

- only roughing with a roughing tool,
- both roughing and finishing with a single tool,
- both roughing and finishing with a roughing tool and a finishing tool, respectively.

The user should decide upon one of the cutting strategies before executing the CTSS. Basic specifications of a typical cutting tool are; *material type, rake angles, approach angle, diameter, number of (teeth) inserts, length and insert information*. CTSS checks the geometry of the component and determines basic tool specifications which are the basic characteristics of candidate tools to be selected from the tooling libraries of machine tools or from a general cutting tool data base that has been prepared by using milling catalogue of Hertel Company [152] based on *ISO coding scheme (See Appendix C)*. After determining those basic characteristics, the tooling data base in use is searched to find candidate tools which fulfil the determined characteristics. Stages in the determination of the tool specifications are explained below.

5.5.1. Specification of Workpiece Material

The first stage of the CTSS is the specification of workpiece material. Tool manufacturers have improved the tool life with their coated inserts and cracked many barriers in front of the machining. Cutting tools to be used in cutting of extremely hard materials with high speeds are easily available. They provide a huge amount of alternative cutting tools and use their own coding in workpiece material specification in their handbooks. Therefore, it is preferable to use their categorisation scheme for the workpiece material selection. In this thesis, materials are specified in terms of the Hertel's material groups [152].

5.5.2. Determination of Cutter-Type Domain for Features

The cutters are usually designed for performing certain cutting operations. For example, a thread milling operation can not be accomplished with an end mill.

However, some of the operations can be performed by using different types of cutters. For example, a shoulder can be machined either using a shoulder mill or an end mill depending upon the size and geometry of the shoulder. A slitting tool may also be used if the geometry is suitable. There are a lot of factors to be considered in the determination of a proper tool type. The geometry of the features on the part has a considerable effect on the type of cutter. Figure 5.11 shows three different orientations of a step feature on a component. If the height and width of the step are of a relatively medium size and they are more or less equal as shown in Figure 5.11.a, a shoulder mill and an end mill can be placed in the cutter-type domain of step feature. If the width of the step is small as compared with its height as shown in Figure 5.11.b, an end mill is sufficient. But, if the height of the step increases, it is better to add a long-edge end mill to the cutter-type domain of step feature. If the width of step is large as compared with its height as illustrated in Figure 5.11.c, in this case a slitting mill may also be added to the type domain of *step* feature.

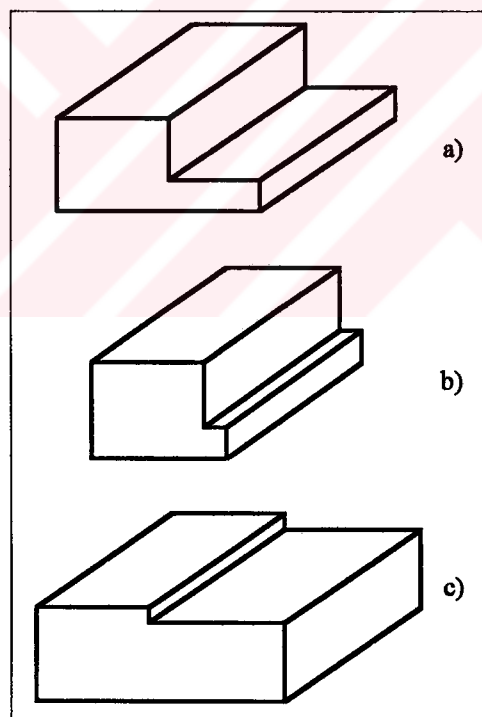


Figure 5.11. Three Different Orientations of a *Step* Feature

Therefore, it is not reasonable to select the cutter type using some simple rules based on the type of features like; "*if feature is a shoulder then cutter is a shoulder mill*". CTSS allows the assignment of candidate tools from the tooling domain of the

machining feature which is mainly related to the geometry of volume to be removed. For example, for a rectangular slot, this domain may have a slotting mill, a long-edge mill, shoulder mill and roughing mill, etc.

5.5.3. Determination of Cutter Material

High Speed Steel (HSS) and carbides are stated as the two most common cutting tool materials for milling operations in [153]. HSS is used generally for soft and ductile materials whereas carbides are used for harder materials. However, for extremely-hard workpiece materials, either CBN (Cubic Boron Nitride) or ceramic inserts should be the first choice. Coated inserts are also available to allow increased speeds and extended tool life in specific materials. Determination of the cutter material is performed in accordance with the workpiece material selection. In this work, material selections for both workpiece and cutting tools are based on the Hertel's material groups [152].

5.5.4. Determination of Characteristic Angles

Design angles of cutters have considerable effects on the cutting performance for some of the milling operations like face milling. Axial and radial rake angles together with the approach angle are the most effective parameters of a face mill.

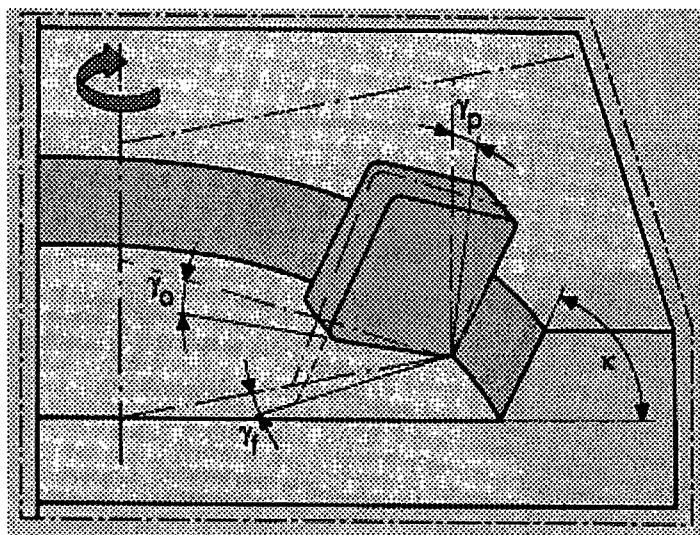


Figure 5.12. Rake Angles and Approach Angle [154]

The orientations of radial rake (γ_f), axial rake (γ_p) and approach (entering) angle (κ) on a cutting tool are depicted in Figure 5.12. Rake is the angle between the base and the face measured in a plane perpendicular to the base. Rake angles have considerable effects on the cutting forces and power required to accomplish the operation. Larger rake angle leads to lower cutting forces and power consumption and better surface finish. Smaller rake angles are generally used for harder materials whereas larger rakes are used for soft-ductile materials [153]. There are mainly three main designs as regards geometry of rake angles; *double negative*, *double positive* and *positive-negative*, as illustrated in Figure 5.13.

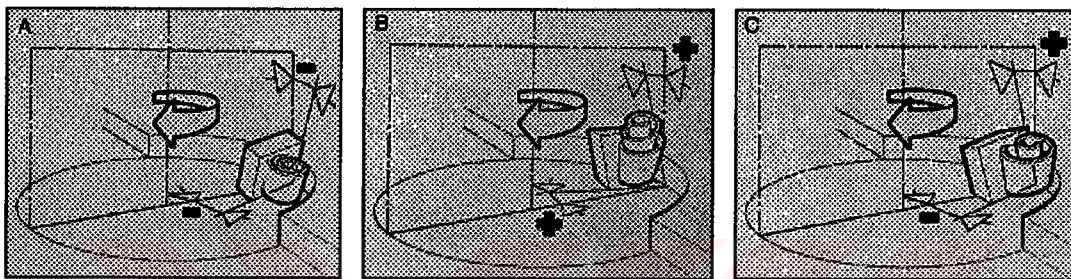


Figure 5.13. Rake Angle Designs [154]

Positive axial-rake and negative radial-rake combinations can cut all materials well. Double positive geometry (positive axial-rake and positive radial-rake) is used for soft materials like aluminium. Double negative geometry (negative axial-rake and negative radial-rake) is most suitable for hard materials like cast iron. For example, if the workpiece is fragile, positive angles are used for both of the rake angles. If the cutter is an end mill, and length of cut is too long, then it is better to use positive and negative values for axial and radial rake angles, respectively. The selection of rake angles is performed by using the assignments of Table 5.6.

The approach angle affects the cutting forces produced on the machine, cutter and component. If smaller approach angle is used, smaller radial forces are generated. A 75 degree approach angle is usually applied since it is suitable for roughing and finishing. A 45 degree approach angle presents a stronger cutting edge for heavy duty machining. In this case, the axial cutting force will be more or less equal to the radial cutting force. Therefore, if a long spindle overhang is expected, 45 degree should be

preferred. 45 degree approach angle is also suitable when milling cast irons for which edge chattering of the workpiece is a problem. The determination of approach angle also depends on the geometry of the part features. For example, it is obvious that approach angle is directly taken as 90 degree when selecting a cutter for a square shoulder on the part.

Table 5.6. Rake Angle Combinations with Hertel Material Groups [152,155]

Hertel Material Group No	Axial Rake	Radial Rake
1	Positive	Positive
2	Positive	Positive
3	Positive	Negative
4	Positive	Positive
5	Positive	Positive
6	Positive	Negative
7	Positive	Negative
8	Negative	Negative
9	Positive	Positive
10	Positive	Negative
11	Positive	Negative
12	Positive	Negative
13	Positive	Negative
14.1	Positive	Positive
14.2	Positive	Negative
14.3	Negative	Negative
15.1	Positive	Negative
15.2	Negative	Negative
16	Positive	Negative
17	Positive	Negative
18	Negative	Negative
19	Positive	Negative
20	Positive	Positive
21	Positive	Negative
22	Positive	Negative

5.5.5. Determination of Cutter Diameter

This stage is the most critical step in the determination of cutting tool specification. The diameter of the cutting tool is basically determined by the size of the feature to be machined. For hole features, the diameter is the diameter of the hole to be machined. The determination of cutter diameter for face milling is one the most

complicated task among all the milling operations. There is a wide range of diameters to be used and recommendations on this range vary from source to source, or from manufacturer to manufacturer. For example, in order to get best result for face milling, Gülesin and Jones [4] have recommended that the cutter diameter should be 1.6 times the width of the region machined. Instead of 1.6, Hertel [155] gives 1.25, whereas a range (1.20-1.50) is given by Sandvik [9]. This problem is solved in such a manner that tool diameter is calculated by using the multiplication factor specified in the corresponding manufacturer's catalogue. If the diameter found is bigger than the biggest tool available in the tooling data base, the biggest tool diameter available is selected.

When selecting an end mill for a milling operation, the workpiece material to be cut is an extremely important consideration. However, part (feature) configuration plays the most important role when determining the tool size. Since deflection is another important parameter to be kept in control when selecting an end mill, the most desirable tool size is assumed as the one which has the largest diameter with the shortest length.

5.5.6. Determination of Cutter Length and Number of Inserts

The effective length is generally determined by the depth of feature to be machined. For example, the effective length of a twist drill to be used to cut a thru hole, should be 1.1 times longer than the depth of the hole. It is usually recommended to use a cutter with the smallest number of teeth (inserts) that will keep at least two inserts in contact with the workpiece. The number of inserts is approximated by using the following equation [4];

$$N_i = \frac{1.4\pi D}{W_c}$$

where N_i is the number of inserts, D is the diameter of the cutter and W_c is the width of the cut. Milling cutters are classified into coarse, close or extra close pitch cutters. There are some special cases for some milling applications or milling tools. For

example, when performing a heavy cut like slotting with an end mill (or slot mill), the removal of chip is essential for the quality. For these applications, it is always better to select an end mill which provides sufficient room for making the chip removal easy (i.e., a 2-teeth end mill).

When all of the cutting tool specifications are determined, the most suitable cutter(s) that meet(s) these basic requirements is/are retrieved from the cutting tool data base. Tool selection process for arbitrarily-shaped pockets is different from the procedure discussed in this section. It will be considered later in Chapter 8.

5.6. TOOL HOLDER SELECTION MODULE

Cutters are mounted onto spindle of milling machines by means of tool holders. Having selected the cutters in the tool selection module, the corresponding tool holders are then selected from the tool holder data base, by taking the machine tool and cutter's requirements into account. The tool holder data base is prepared by using catalogues of several tool-holder manufacturers [152,154,156,157].

There may be sometimes a necessity of using adaptors between cutters and basic tool holders mounted on the spindles of milling machines [158]. Type of adaptation depends mainly upon design and size of cutter to be employed. Common types of adaptors are shown in Figure 5.14 to Figure 5.16. The combination arbor with key driver shown in Figure 5.14 is generally used for adapting milling cutters with a cylindrical bore. The arbor shown in Figure 5.15 provides centering of cutter. The centering mandrel with steep taper shank illustrated in Figure 5.16 is used for adapting large and heavy cutters directly onto the spindle of the machine tool.

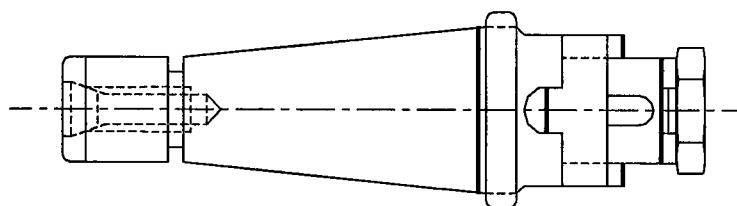


Figure 5.14. Combination Arbor

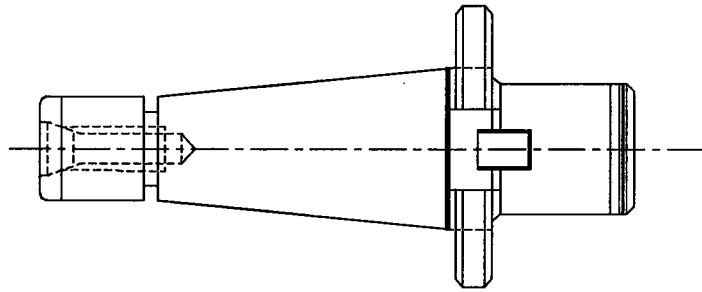


Figure 5.15. Arbor with Internal Centering

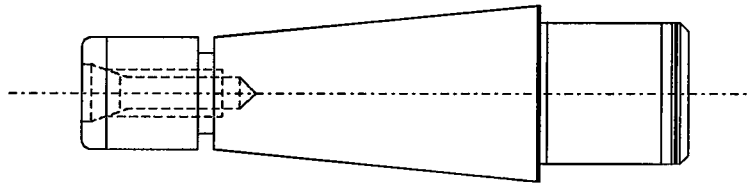


Figure 5.16. Centering Mandrel with Steep Taper Shank

Unfortunately, there is not a unique coding scheme for tool holding of milling operations developed by any organisation. One of the reason for this is that the tooling in milling is more complex and requires much more detail than the tooling in turning. Cutting tool manufacturers develop their own tool holding systems while keeping the specific standards given for each type of adaptation into view. But they all specify DIN standards for the adaptors they produce [158].

In this study, tool adaptors are selected for TAKSAN TMC 700 VMC which requires BT40 types of adaptors. An example for selection of cutting tools and tool holders for machining operations will be presented in Chapter 9.

5.7. SET-UP PLANNING AND SEQUENCING MODULE

The main objective behind set-up planning is to maximise the number of machining operations which can be accomplished in a single set-up, in other words; is to minimise the total number of set-ups required to machine a component. However, the accuracy of machined component is also a vital requirement of set-up planning. Set-up planning mainly depends on [159];

- tool approach direction(s),
- tolerances,
- feature interactions,
- good machining practice.

In this work, a hybrid set-up planning and sequencing system has been developed. The inclusion of the word “hybrid” means that the developed system combines the advantages of the methods suggested in References [61] and [62]. It is assumed that a feature can be finished in one set-up. Cases in which rough cutting and finish cutting of the same feature may need different set-ups are not considered. It is also assumed that the workpiece is safely and firmly fixed on the machine tool table by any means of clamping and fixturing apparatus.

The set-up plans with the minimum number of set-ups for machining prismatic components on a 3-axis VMC can be found by using the heuristic rules based on the tool approach direction(s), tolerances, feature interactions and good machining practice. Since the precedence relationships are also considered in the set-up planning, the resulting sequence is an inherently-natural set-up sequence without considering the geometrical tolerances. This sequence can be modified to include the geometrical tolerance requirements by using the converted tolerance scheme proposed in Reference [62], if the precedence relationships between the tolerances allow.

5.7.1. Motivation and Methodology

In Zhang’s implementation [61], tool approach directions together with bilateral precedence relationships obtained for different type of set-up constraints have been taken into consideration to achieve a good set-up plan. These relationships are due to geometric constraints, feature interactions, good manufacturing practice and datum to reference requirements (not the tolerances). An example to the feature interactions is depicted in Figure 5.17. As shown in this figure, *Slot 1* has a precedence over *Slot 2*, that is *Slot 2* can only be machined after *Slot 1*. The detailed information on the other examples can be found in Reference [61]. Linear tolerance (datum to reference) requirements determine some precedence relationships between

certain features which must be satisfied by the set-up sequence. Therefore, choosing a set-up for a feature must fulfil the datum to reference requirements of the feature. However, in the example given in Reference [61], the effect of these requirements were not included and demonstrated.

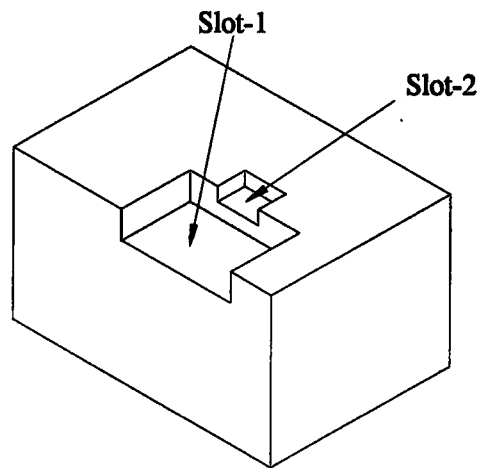


Figure 5.17. Interaction Between the Slots

FIXES (A System for Automatic Selection of Set-ups and Design of Fixtures) developed by Boerma and Kals is mainly based on geometrical tolerances. They claimed that closely related features have to be machined in one set-up, while less accurately related features can be machined in different set-ups, so the set-up selection has to be preceded by an evaluation of the tolerances concerning the geometrical relations between different features [62]. The reason for this claim is that the positional accuracy of a feature which has already been machined in a previous set-up cannot be sufficient to realise the required accuracy in the relations between that feature and the ones which have to be machined in the present set-up. Therefore, the main objective was to reduce the number of critical tolerances in the geometrical relations between features belonging to different set-ups and to keep the number of set-ups as low as possible. Comparison of significance of the different tolerances in the relations between the features plays an important role in their set-up planning procedure. However, different types of tolerances cannot directly be compared with each other. For this purpose, Boerma and Kals introduced a tolerance factor (T.F.) for each type of relation. A tolerance is converted into a tolerance factor by dividing the tolerance value by the representative length. This length depends on the

type of tolerance and the dimensions of the part (see Figure 5.18). So, the T.F. represents the tangent of the maximum admissible angle of rotation of the feature concerned (see Figure 5.19).

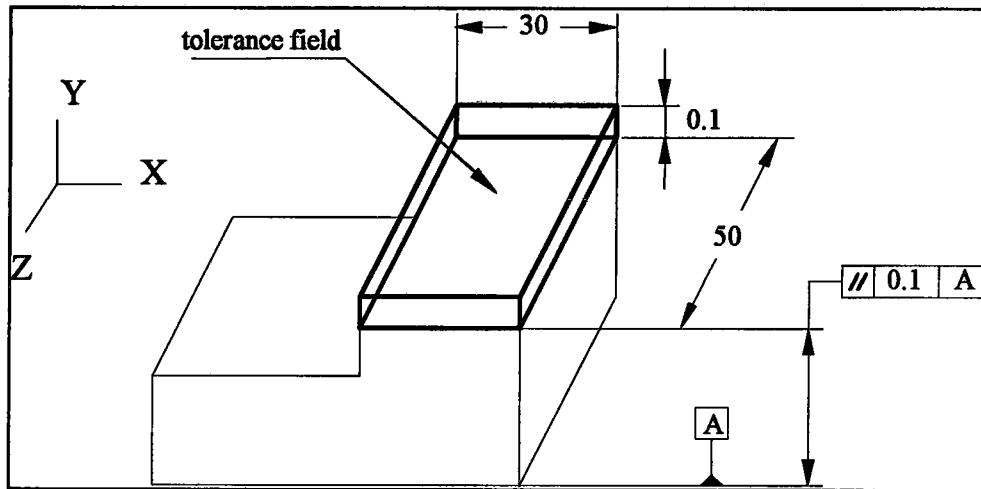


Figure 5.18. Relevant Lengths of a Tolerance [62]

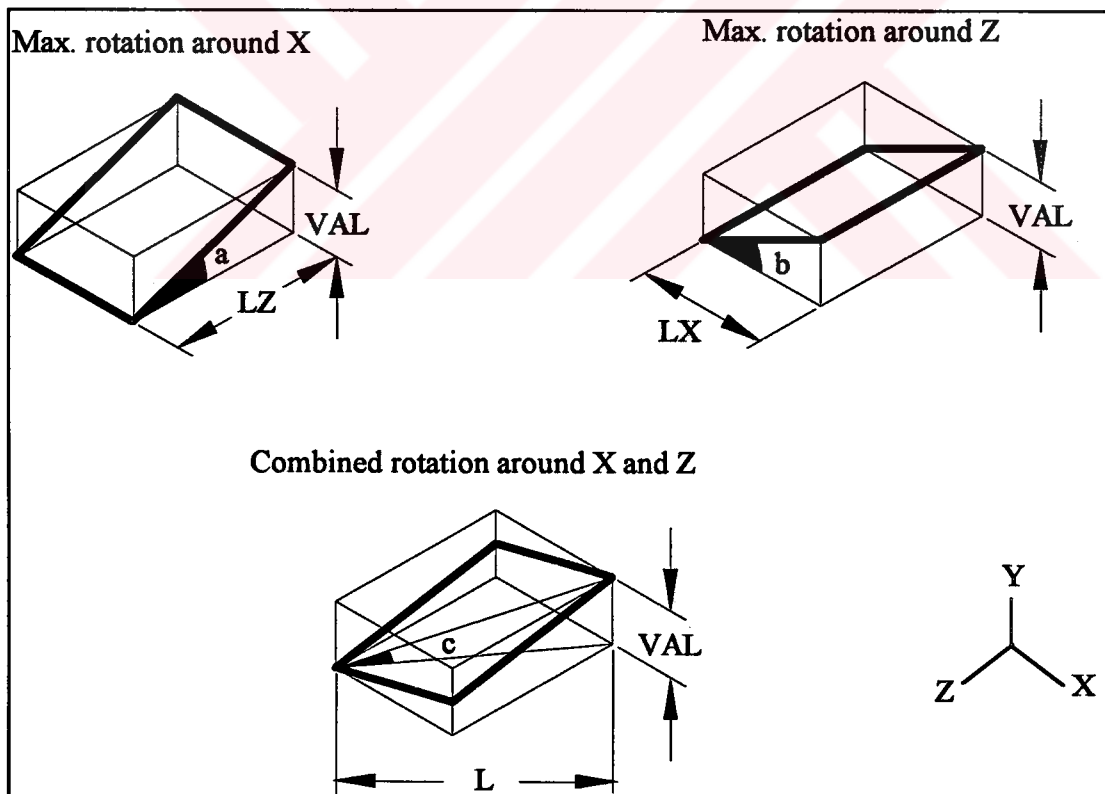


Figure 5.19. The Maximum Admissible Rotation Within the Tolerance Field [62]

In Figure 5.19, *Tolerance Factor* (T.F.) is introduced as follows;

Maximum rotation around X is $R_X = \tan(a) = VAL / LZ$

Maximum rotation around Z is : $R_Z = \tan(b) = VAL / LX$

Combined rotation around X & Z is : $T.F = \tan(c) = VAL / L$

where, LX and LZ are the relevant lengths, VAL is the tolerance value determining the dimension of the tolerance field and L is the representative length given by;

$$L = \sqrt{(LX^2 + LZ^2)}$$

T.F. is a factor for the admissible rotation deviation of the dark-lined face with a combined rotation around X and Z. The sequence of set-ups is determined by using a heuristic such that the last set-up contains on the average the most accurate relations and the first set-up is left with the less accurate relations. For details on their methodology, see Reference [62].

An important distinction between those two approaches discussed above is that FIXES can handle the geometrical relationships with non-feature regions (surfaces or faces) of a component. Indeed, most of the technical drawings of machining industry are of these type of representations, not only the relationships between features on the component. So, these non-feature regions must also be considered in the set-up planning and sequencing function. In this work, the effect of geometrical tolerances on the set-up sequences are also included using the heuristic proposed in Reference [62]. The developed set-up planning and sequencing system has two submodules; set-up planning by using feature precedence relationships and refinement of the obtained set-ups (set-up sequencing).

Input to the first submodule are the attributes of precedence relationships derived from set-up constraints and possible approach directions of features. It is assumed that a feature can be finished in one set-up. Therefore, the maximum number of set-ups required for machining a prismatic part is 6. All possible solutions which satisfy various set-up constraints are computed; features that can be machined from

the same approach direction are grouped into set-ups. In a reasonable set-up plan; every feature must appear in one set-up, must have the approach direction associated with that set-up, and the order of individual set-ups in a set-up plan must not violate the precedence relationships between the features. The inherently-optimal solution is then searched among all possible set-up plans; from 1 to 6. The solution with the minimum number of set-ups (approach directions) is selected as the inherently-optimal set-up plan. All feasible set-ups as well as the inherently-optimal set-up plan are stored for future use. For each set-up plan, number of set-ups, set-up orientation, and features to be machined in that set-up are specified.

The function of the second submodule is actually to modify the inherently-optimal set-up sequence obtained by the first submodule, if possible. This submodule takes a converted tolerance chart prepared for a component as the input. It includes the list of geometrical tolerances, their types and dimensions, tolerance and datum surfaces.

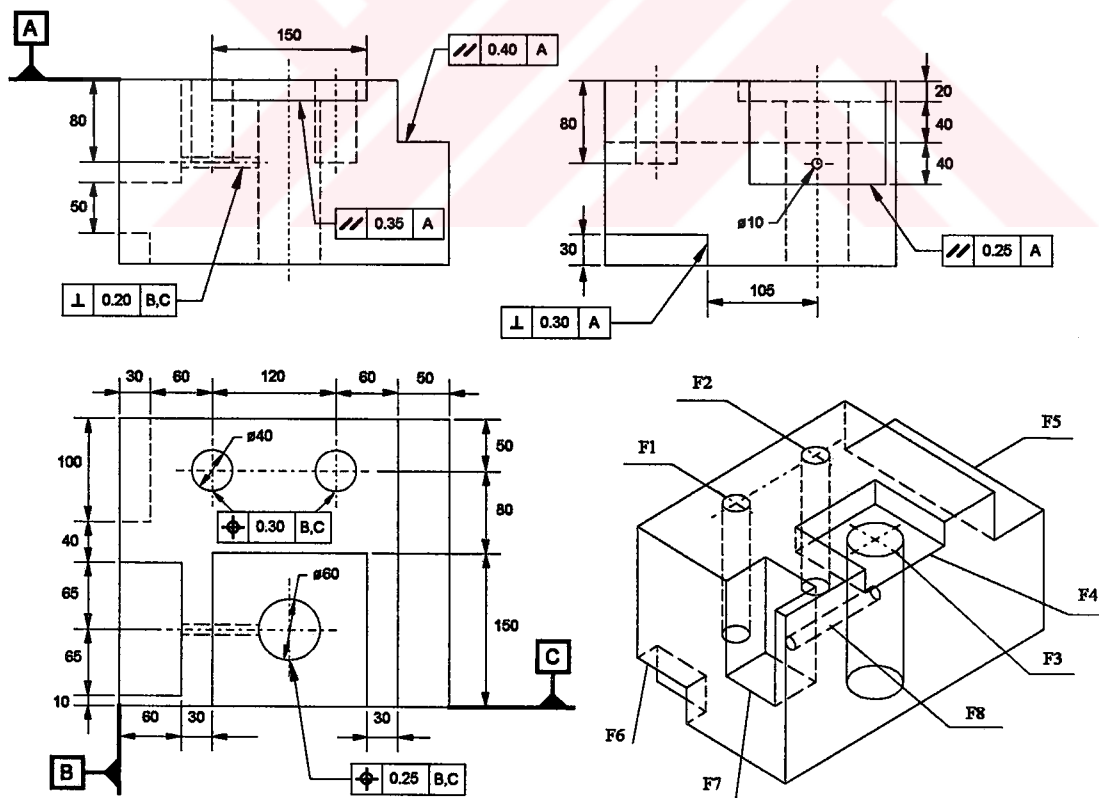


Figure 5.20. An Example Part for Set-up Planning

For each feature, a tolerance factor to evaluate and compare the specified geometrical relations on the part has been calculated by using the dimensions and tolerance values given for the features. The set-ups that have the most accurate average of tolerance factors are left to the end of set-up sequence, if precedence relationships are not destroyed. If there is no geometrical tolerance imposed on the part, the set-up sequencing is performed in such a way that the set-ups containing the smallest number of features are left to the end of the set-up sequences.

5.7.2. An Example to Set-up Planning

The example part is reproduced from Reference [61]. The isometric view of the example component with its orthographic projections is given in Figure 5.20. Precedence relationships between the part features and approach directions of the features are presented in Table 5.7 and Table 5.8, respectively.

Table 5.7. Precedence Attributes of the Example Part in Fig. 5.20

Feature	F1	F2	F3	F4	F5	F6	F7	F8
F1	0	0	0	0	0	0	0	0
F2	0	0	0	0	0	0	0	0
F3	0	0	0	-1	0	0	0	1
F4	0	0	1	0	0	0	0	1
F5	0	0	0	0	0	0	0	0
F6	0	0	0	0	0	0	0	0
F7	0	0	0	0	0	0	0	1
F8	0	0	-1	-1	0	0	-1	0

Table 5.8. Tool Approach Directions of Example Part in Fig. 5.20

Feature	AD-1	AD-2	AD-3	AD-4	AD-5	AD-6
F1	1	0	0	0	0	0
F2	1	0	0	0	0	0
F3	1	0	0	1	0	0
F4	1	0	0	0	1	0
F5	1	0	1	0	0	0
F6	0	1	0	1	0	1
F7	1	0	0	0	0	1
F8	0	0	0	0	0	1

All possible set-ups are found by the first stage of the developed program is given in Table 5.9, except 6 set-ups. As can be seen from this table, there are two alternative set-ups for optimal solution which includes 2 set-ups; Plan No:1 and Plan No:19. These two optimal solutions are summarised in Table 5.10.

Table 5.9. All Feasible Set-ups for the Example Part in Fig. 5.20

Plan No	# Set-ups	Approach Directions								Features							
		A	B	C	D	E	F	G	H	A	B	C	D	E	F	G	H
1	2	1	1	1	1	1	1	6	6	1	2	3	4	5	7	6	8
2	3	1	1	1	1	1	1	2	6	1	2	3	4	5	7	6	8
3	3	1	1	1	1	1	1	4	6	1	2	3	4	5	7	6	8
4	3	3	1	1	1	1	1	6	6	5	1	2	3	4	7	6	8
5	3	5	1	1	1	1	1	6	6	4	1	2	3	5	7	6	8
6	4	2	3	1	1	1	1	1	6	6	5	1	2	3	4	7	8
7	4	2	5	1	1	1	1	1	6	6	4	1	2	3	5	7	8
8	4	3	1	1	1	1	1	4	6	5	1	2	3	4	7	6	8
9	4	3	5	1	1	1	1	6	6	5	4	1	2	3	7	6	8
10	4	5	1	1	1	1	1	4	6	4	1	2	3	5	7	6	8
11	4	5	4	4	1	1	1	1	6	4	3	6	1	2	5	7	8
12	4	5	4	4	6	6	1	1	1	4	3	6	7	8	1	2	5
13	5	2	3	5	1	1	1	1	6	6	5	4	1	2	3	7	8
14	5	2	5	4	1	1	1	1	6	6	4	3	1	2	5	7	8
15	5	2	5	4	6	6	1	1	1	6	4	3	7	8	1	2	5
16	5	3	5	1	1	1	1	4	6	5	4	1	2	3	7	6	8
17	5	3	5	4	4	1	1	1	6	5	4	3	6	1	2	7	8
18	5	3	5	4	4	6	6	1	1	5	4	3	6	7	8	1	2
19	2	1	1	1	1	1	1	6	6	1	2	3	4	5	6	7	8
20	3	1	1	1	1	3	6	6	6	1	2	3	4	5	6	7	8
21	3	1	1	1	1	4	6	6	6	1	2	4	5	3	6	7	8
22	3	1	1	1	1	1	6	6	2	1	2	3	4	5	7	8	6
23	4	1	1	1	3	4	6	6	6	1	2	4	5	3	6	7	8
24	4	1	1	1	1	3	6	6	2	1	2	3	4	5	7	8	6
25	4	1	1	1	1	4	6	6	2	1	2	4	5	3	7	8	6
26	4	1	1	1	5	4	6	6	6	1	2	5	4	3	6	7	8
27	5	1	1	1	3	4	6	6	2	1	2	4	5	3	7	8	6
28	5	1	1	3	5	4	6	6	6	1	2	5	4	3	6	7	8

Table 5.10. Optimal Set-up Plans Obtained in the First Stage for the Example Part

Optimum Set-up Plan No	Set-up Plan No	# Set-ups	Orientation	Features in the Set-up
1	1	1	AD - 1	F1, F2, F3, F4, F5, F7
		2	AD - 6	F6, F8
2	19	1	AD - 1	F1, F2, F3, F4, F5
		2	AD - 6	F6, F7, F8

After obtaining the naturally optimal set-up plans, the second stage of the program is executed to sequence the set-ups or to check whether a modification might be done in the set-up sequences. The input to the second stage is the converted tolerance chart prepared for the component which is given in Table 5.11. For each set-up, an average tolerance factor is calculated. The average values of tolerance factors and corresponding set-up sequences for the example component are given in Table 5.12.

Table 5.11. The Converted Tolerance Table for the Example Part in Fig 5.20

Tolerance		Reference Feature		TOLERANCE								
NO	F.O	NO	F.O.	TYPE	VAL	LENGTH			ROTATIONAL ERROR			T.F.
						LX	LY	LZ	RX	RY	RZ	
F1	-Y	B,C	X,Z		0.30	0	80	0	0	0.30/80	0	0.00375
F2	-Y	B,C	X,Z		0.30	0	80	0	0	0.30/80	0	0.00375
F3	-Y	B,C	X,Z		0.25	0	160	0	0	0.25/160	0	0.00156
F4	-Y	A	Y	//	0.35	150	0	150	0.35/150	0	0.35/150	0.00165
F5	-Y	A	Y	//	0.40	50	0	280	0.40/50	0	0.40/280	0.00140
F6	-Z	A	Y		0.30	30	30	0	0.30/30	0.30/30	0	0.00707
F7	-Y	A	Y	//	0.25	60	0	130	0.25/60	0	0.25/130	0.00174
F8	+X	A,C	Y,Z		0.20	75	0	0	0.20/75	0	0	0.00266

Table 5.12. Set-up Sequences Based on Tolerance Factors

Optimum Set-up Plan No	Set-up Plan No	# Set-ups	Orientation	Average T.F.	Set-up Sequence
1	1	1	AD - 1	2.31×10^{-3}	6 - 1
		2	AD - 6	4.87×10^{-3}	
2	19	1	AD - 1	2.42×10^{-3}	6 - 1
		2	AD - 6	3.83×10^{-3}	

If the values and types of the tolerances of the part are changed as given in Table 5.13, the tolerance factors are also changed. In this case, the set-up sequences are different from the previous case. The results are given in Table 5.14.

Table 5.13. A Converted Tolerance Table II for the Example Part in Fig. 5.20

Tolerance		Reference Feature		TOLERANCE								
NO	F.O	NO	F.O.	TYPE	VAL	LENGTH			ROTATIONAL ERROR			T.F.
						LX	LY	LZ	RX	RY	RZ	
F1	-Y	B,C	X,Z		0.20	0	80	0	0	0.20/80	0	0.00250
F2	-Y	B,C	X,Z		0.20	0	80	0	0	0.20/80	0	0.00250
F3	-Y	B,C	X,Z		0.15	0	160	0	0	0.15/160	0	0.00093
F4	-Y	A	Y	//	0.35	120	0	120	0.35/120	0	0.35/120	0.00206
F5	-Y	B	Y		0.40	50	0	280	0.40/50	0	0.40/280	0.00141
F6	+Y	A	Y	//	0.10	30	0	100	0.10/30	0	0.10/100	0.00096
F7	+X	B	Y	//	0.25	0	100	130	0	0.25/100	0.25/130	0.00152
F8	+X	A,C	Y,Z		0.10	75	0	0	0.10/75	0	0	0.00133

Table 5.14. Set-up Sequences Based on Tolerance Factors (Second Case)

Optimum Set-up Plan No	Set-up Plan No	# Set-ups	Orientation	Average T.F.	Set-up Sequence
1	1	1	AD - 1	1.82×10^{-3}	1 - 6
		2	AD - 6	1.15×10^{-3}	
2	19	1	AD - 1	1.88×10^{-3}	1 - 6
		2	AD - 6	1.27×10^{-3}	

CHAPTER VI

DESIGN FOR MANUFACTURING

6.1. INTRODUCTION

In this chapter, *Design for Manufacturing (DFM)* which is sometimes considered as one of the implementation tools used for *Concurrent Engineering (CE)* is introduced. Then, DFM module developed in this work is explained.

6.2. DESIGN FOR MANUFACTURING / CONCURRENT ENGINEERING

Design for Manufacturing (DFM) or in other words *Concurrent Engineering (CE)* (sometimes called as *Simultaneous Engineering*) can be defined as a way of work where the various engineering activities in the product and production development processes are integrated and performed simultaneously as much as possible rather than in sequence [160]. This means that all of the design constraints are included as part of the functional optimisation of the design. In this way, DFM enables designers to consider all aspects of the product design and manufacturing at early stages of the design cycle, so that design iteration and accompanying engineering changes can be made easily and effectively. This has great advantages because it leads to minimum manufacturing problems. Consideration of DFM principles during the design stage will enhance the producibility of the part while

reducing the cost. Note that the term CE is more inclusive than DFM, even DFM is accepted as a tool for the implementation of CE.

DFM allows the designer to work with a set of manufacturing constraints and requirements in the design of the product. It is clear from the above discussion that *DFM is only achievable by establishing some design rules in the form of knowledge engineering used to guide the design to obey manufacturing requirements*. CE refers to considering manufacturing problems early in the design process in order to reduce product development time and product manufacturing time. Two basic scenarios can generally be considered to realise the objectives of CE [161]; a) design of a new manufacturing system for existing products, b) design of a product for an existing manufacturing system. Whichever approach is used, products should be designed by considering the manufacturing system, or manufacturing system should be designed for the products.

6.3. ELIMINATION OF CRITICAL REGIONS ON THE COMPONENT

Little work has been reported on the automatic evaluation of critical regions of components in the literature [162]. This work is an attempt for the automatic determination of critical regions between interacting features on prismatic components. This section presents the methodology of the DFM module that automatically evaluates critical regions between various types of machining features (hole, pocket, etc.) on prismatic components and modifies original design (if possible) and reflects the modifications to all down-stream and up-stream applications in the CAD/CAM integration, like process planning.

Part representation is considered as one of the most difficult problems in CAD/CAM integration. *Features* have been extensively used in the feature recognition or in feature-based design systems developed for the part representation during the last decade. In a machining environment, individual features cannot be considered in isolation since cutting strategy consideration has to take into consideration the relationship between interacting features. This relationship may be

due to proximity, overlap, geometric tolerances or other considerations. Feature interactions may cause serious problems in process planning and manufacturing. Therefore, the control of feature interactions are vital for complete and true CAD/CAM integration.

There may be critical regions on a component between different features[163]. Moreover, location or size of two interacting features may require a special production method. The component to be machined must also withstand machining and fixturing forces, and it should not be deformed or deformations must be within tolerable limits. The critical regions like thin sections as shown in Figure 6.1., govern the stiffness of the component. However, definition of a thin section is a complex task. For the same minimum wall thickness t , shown in Figures 6.1, the effect on stiffness would differ [65]. The program prepared for the fulfilment of this “*Design for Manufacturing*” facility uses a CAD data base, a material data base and a tooling data base as the input. On the basis of selected type of remedy, the program can find the critical regions between features that are evaluated by the system for each type of feature interactions. If necessary, some of the features are shifted or removed from the part design in order to eliminate critical regions. Cutting conditions, tools to machine features and operation sequence can also be changed according to the user specifications.

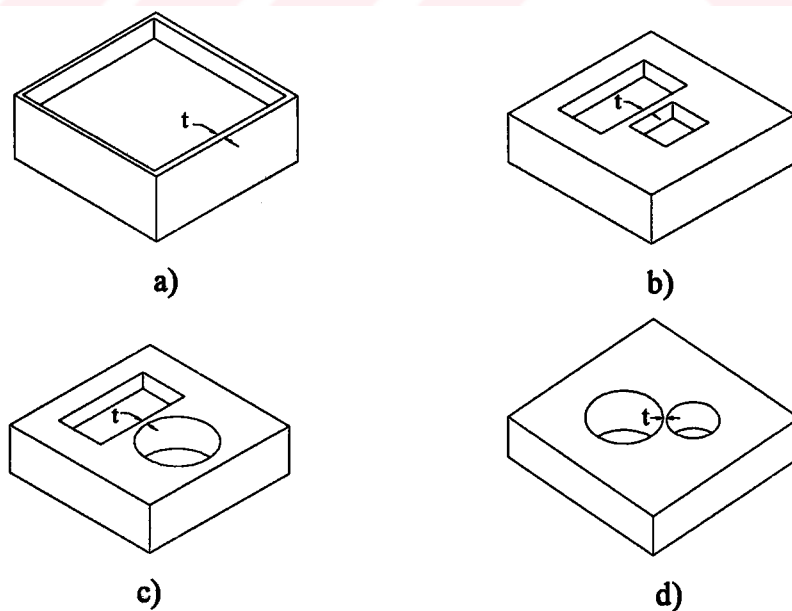


Figure 6.1. Thin Sections Between Part Features

6.3.1. Methodology of the Developed System for Critical Regions

The required CAD data is obtained by post-processing the B-Rep based CAD output provided by the feature recognition module. Upon executing the program, user is required to select the driving criterion. The next stage is the determination of each minimal distance, t , between interacting features. By scanning the explicit CAD data, each distance relationship between the features are identified. Design variables to be checked with driving criterion are then calculated. Torque, T , is calculated by;

$$T = F_t * (D/2) \dots\dots\dots (6.1)$$

where; F_t is tangential force and D is diameter of cutting tool. F_t is determined from the following formula;

$$F_t = 60000 * P_c / V \dots\dots\dots (6.2)$$

where; V is cutting speed and P_c is the power required for cutting. The cutting speed is obtained by the formula;

$$V = \pi D N \dots\dots\dots (6.3)$$

where; N is spindle speed. Maximum stress at critical section is calculated by the following equation;

$$\tau = T / (2tA) \dots\dots\dots (6.4)$$

where; t is minimal distance between interacting features and A is representative critical area. The derivation of Equation 6.4 can be found in *Appendix D*. One of the most critical step in this session is the calculation of representative critical area. Some examples on the derivation of these representative areas can be found in [163]. The representative (effective) critical area between two neighbouring rectangular pockets is shown in Figure 6.2, and equal to;

$$A = (a + t) * (b + t) \dots\dots\dots (6.5)$$

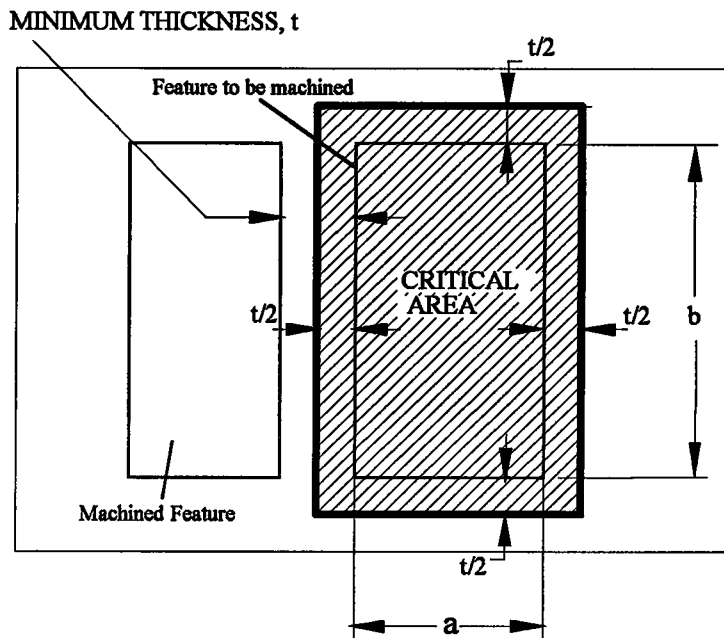


Figure 6.2. Representative Critical Area Between Two Rectangular Pockets [162]

The representative critical area between two neighbouring circular pockets or holes is shown in Figure 6.3, and it is equal to:

$$(\pi / 4) [(a + t / 2) (a + t / 2)] \dots\dots\dots (6.6)$$

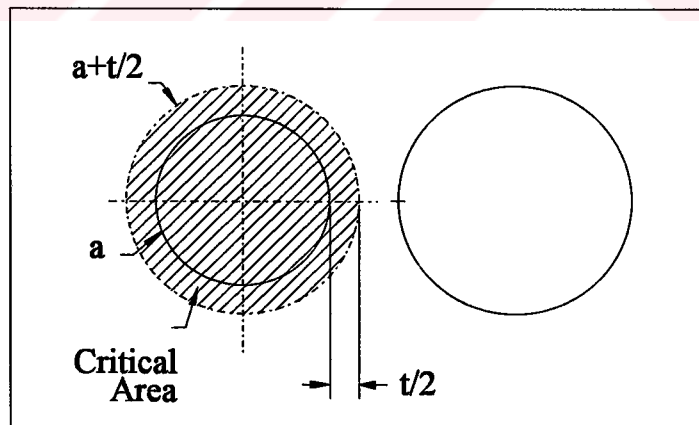


Figure 6.3. Representative Critical Area Between Two Holes [163]

Calculated maximum shear stress τ is compared with the Shear Yield Strength (S_{sy}) of the material. If the maximum stress is less than or equal to S_{sy} , then interacting features can be machined in the specified operation sequence and under the specified cutting conditions. Otherwise, the machining of second feature can lead to a failure. At this stage, the program suggests five remedies; *changing the material, shifting/removing some of the features, changing the cutting conditions, changing the cutter diameter(s) and changing the operation sequence*. Among these, the last three solutions are not advised, especially in an integrated process planning environment where all optimisations are previously accomplished. Actually, the type of remedies to be selected depend on the stage at which the program is used for CAD/CAM activities. Feature interactions can only be eliminated in the design stage by using the estimated values of required parameters like cutting parameters and by assuming cutters and the operation sequence, etc. The procedure for this *Design for Manufacturing* module proceeds in the following stepwise fashion;

Step 1: Start 'Design for Manufacturing' session

Step 2: Get geometrical and technological information of the component

Step 3: Get the information from material and tooling data base in use

Step 4: Determine critical area of the region between interacting features

Step 5: Calculate design variables

Step 6: Compare calculated variables with the permissible values

Step 7: If OK, go to Step 8.

If checking fails, suggest remedies;

i) Change Material

ii) Shift Features

iii) Change Cutting Conditions

iv) Change Cutter Diameter

v) Change Operation Sequence

and go to Step 5 for re-check.

Step 8: Print: "No critical region is found on the part".

Step 9: Stop

6.3.2. An Example

Consider the example part shown in Figure 6.4. There may be critical regions on the component regarding the distances between the features. Candidate critical regions that may be found on this component include; a thin wall (Feature 1), area between neighbouring rectangular pockets (Feature 2 and Feature 3), area between a rectangular pocket and a circular pocket (or hole) (Feature 3 and Feature 5) and area between neighbouring circular pockets or (holes) (Feature 5 and Feature 6). These minimum distances are automatically calculated by using the B-Rep data base. Top view of the example part with its dimensions (in mm) is shown in Figure 6.5.

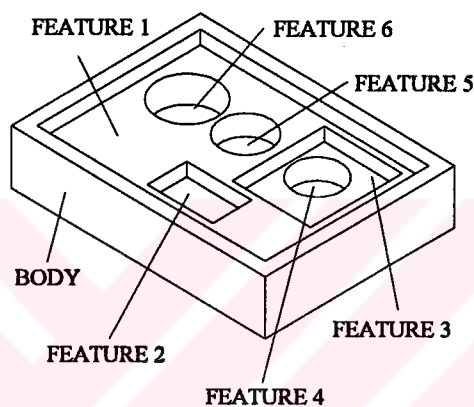


Figure 6.4. An Example Part

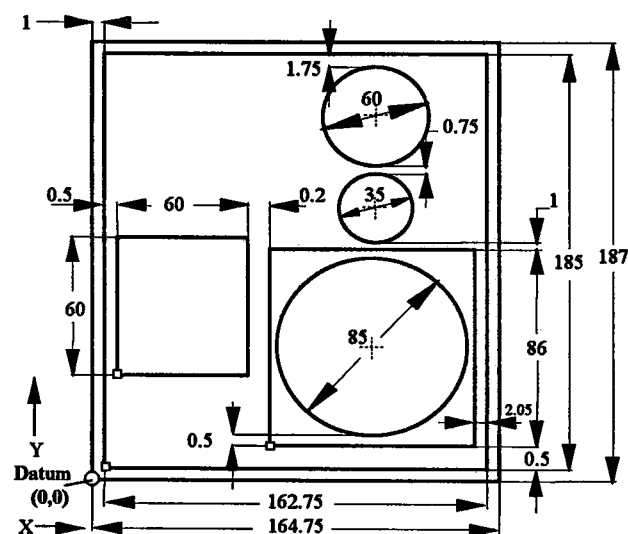


Figure 6.5. Top View of the Example Part

Table 6.1. Geometric Database of the Example Part

FEATURE NO	FEATURE TYPE	dx (mm)	dy (mm)	WIDTH/RADIUS (mm)	LENGTH (mm)	DEPTH (mm)
1	R-PKT	1	1	162.75	185	10
2	R-PKT	1.5	44	60	60	10
3	R-PKT	61.7	1.5	100	86	10
4	T-HOL	111.7	44.5	42.5	-	20
5	T-HOL	111.7	106	17.5	-	20
6	T-HOL	111.7	154.25	30	-	20

After post-processing the CAD output regarding the part representation, the geometrical data base as tabulated in Table 6.1 for the part features are obtained. This data base includes the type and label of features, dimensions (width, radius, length and depth) of features and the distances dx and dy (in x and y axis) between the main datum on the body which is always at lower-left corner and the features' datum. The datum for hole features are located on their center coordinates, while they are located on the lower left corner of pockets. In Table 6.1, R-PKT denotes Rectangular Pocket whereas T-HOL stands for Thru Hole. By using this geometrical data base, the minimum distances between interacting features are calculated and are given in Table 6.2. The diameter of tools and speed values that are to be used for machining features on the example part are given in Table 6.3. Blank material is assumed to be Steel ASTM A-30 which has an S_{sy} of 145 MPa.

Table 6.2. Minimum Distances (mm) Between Interacting Features for Example Part

	BODY	FEATURE 1 (F1)	FEATURE 2 (F2)	FEATURE 3 (F3)	FEATURE 4 (F4)	FEATURE 5 (F5)	FEATURE 6 (F6)
BODY	0	1	0	0	0	0	0
FEATURE 1	0	0	0.5	0.5	0	0	1.75
FEATURE 2	0	0	0	0.2	0	0	0
FEATURE 3	0	0	0	0	0.5	1	0
FEATURE 4	0	0	0	0	0	0	0
FEATURE 5	0	0	0	0	0	0	0.75
FEATURE 6	0	0	0	0	0	0	0

Table 6.3. Tool Diameter and Speed for Machining Features on Example Part

FEATURE NO	Diameter (mm)	Speed (RPM)
1	30	300
2	30	800
3	40	600
4	85	250
5	35	300
6	60	500

Assuming that the user selects “*Shift of Features*” type of remedy, the program begins to scan all distance relationships between the interacting features and the data bases of the part. In this way, a safe sequence of operations can also be found. As an example, for the machining sequence of; F1-F2-F6-F5-F3-F4, it is found by the program that there is a critical wall thickness between *Feature 5* and *Feature 3*, and *Feature 6* and *Feature 5*. This means that *Feature 3* and *Feature 5* cannot be machined safely under the specified machining conditions, since *Feature 5* and *Feature 6* are forced to be machined before *Feature 3* and *Feature 5*, respectively in the sequence. This information may also be used in a feature sequencing work as a constraint. The distances between these features are then automatically increased to 1.6, and 0.9 mm by the program as shown in Table 6.4. The geometrical data base of the part is immediately refreshed. The program is then executed for a recheck cycle to see whether the modifications lead to new problems or not.

Table 6.4. Elimination of Critical Distances for the Sequence: F1-F2-F6-F5-F3-F4

Sequence: F1→F2→F6→F5→F3→F4		
Features	Minimum Distance (mm)	Modified Distance (mm)
5 → 3	1	1.6
6 → 5	0.75	0.9

CHAPTER VII

OPTIMISATION OF PROCESS PLANNING EVENTS

7.1. INTRODUCTION

This chapter presents optimisation methodologies of some of the process planning modules discussed in Chapter 5. Combinatorial optimisation techniques including genetic algorithms together with *Travelling Sales Person (TSP)* method and heuristics are used as the optimisation tools. In Section 7.2, optimisation of the sequence of machining operations are discussed. Section 7.3 introduces the *Right Tool Selection System (RTSS)* whose primary function is actually to optimise the sequence of machining operations based on the objective of *minimum tool replacement*. Section 7.4 is devoted to the optimisation of index positions of cutting tools on an automatic tool changer of a CNC machine tool. *Cutting Parameters Optimisation System (CPOS)* is presented in Section 7.5. Each of the optimisation methodologies are illustrated with practical examples.

7.2. OPTIMISATION OF SEQUENCE OF MACHINING OPERATIONS BY USING A GENETIC ALGORITHM

This section describes the strategy behind an optimisation system software developed for determining optimal sequence of machining operations based on either Minimum Tool Change (MTC) or Minimum Tool Travelling Distance (MTTD) or

safety (based on either geometric constraints or strength). Combinations of these criteria might also be used. Among the three alternative objective functions genetic algorithm gives best response to the safety criterion. The first criterion is also handled in a distinct system called RTSS which will be later discussed in this chapter. MTTD is individually not an adequate criterion.

Inputs of the optimisation system include an explicit CAD data base for each component. The structure of this data base is similar to the input data base used in the operation sequencing module as shown in Table 5.2 in Chapter 5. This data base is obtained from the feature recognition module as in the case of the operation sequencing module. Features and their accompanying machining operations (determined in the operation selection module) are sequenced for each set-up (determined in the set-up planning module). Therefore the features' datum are described in 2D.

A reward/penalty matrix for each set-up is automatically prepared by a reward/penalty generator according to the selected criterion by using the corresponding rules structured in the system. Feature sequencing is performed using a *genetic algorithm* to find an optimal sequence which is the one that has the least total penalty or largest total reward. The algorithm takes the list of features and generates an initial population of sequences. Strings from *TSP* algorithm are also fed into the initial population to enhance the production of good sequences. Genetic operators of mutation, reproduction and crossover are used in generating the new sequences.

7.2.1. The Proposed Approach

The approach we adopted is based on a GA in which the initial population is fed with closed-end alternative solutions obtained via *TSP* strategy. Simplified flowchart of developed system is given in Figure 7.1. The selected tools for operations can also be accessed from the tooling module. The features are listed as collapsed according to approach directions (set-ups) into 2D, in the CAD data base. The representative (projected) surface areas of the features perpendicular to the approach directions are calculated and made available. The physical interactions as well as

relationships and associations between features are then determined. Intersections or inter-feature relations are considered when two or more feature volumes that physically interact or overlap with each other, whereas an association can be considered when there are two features of the same type or similar surface area on the part.

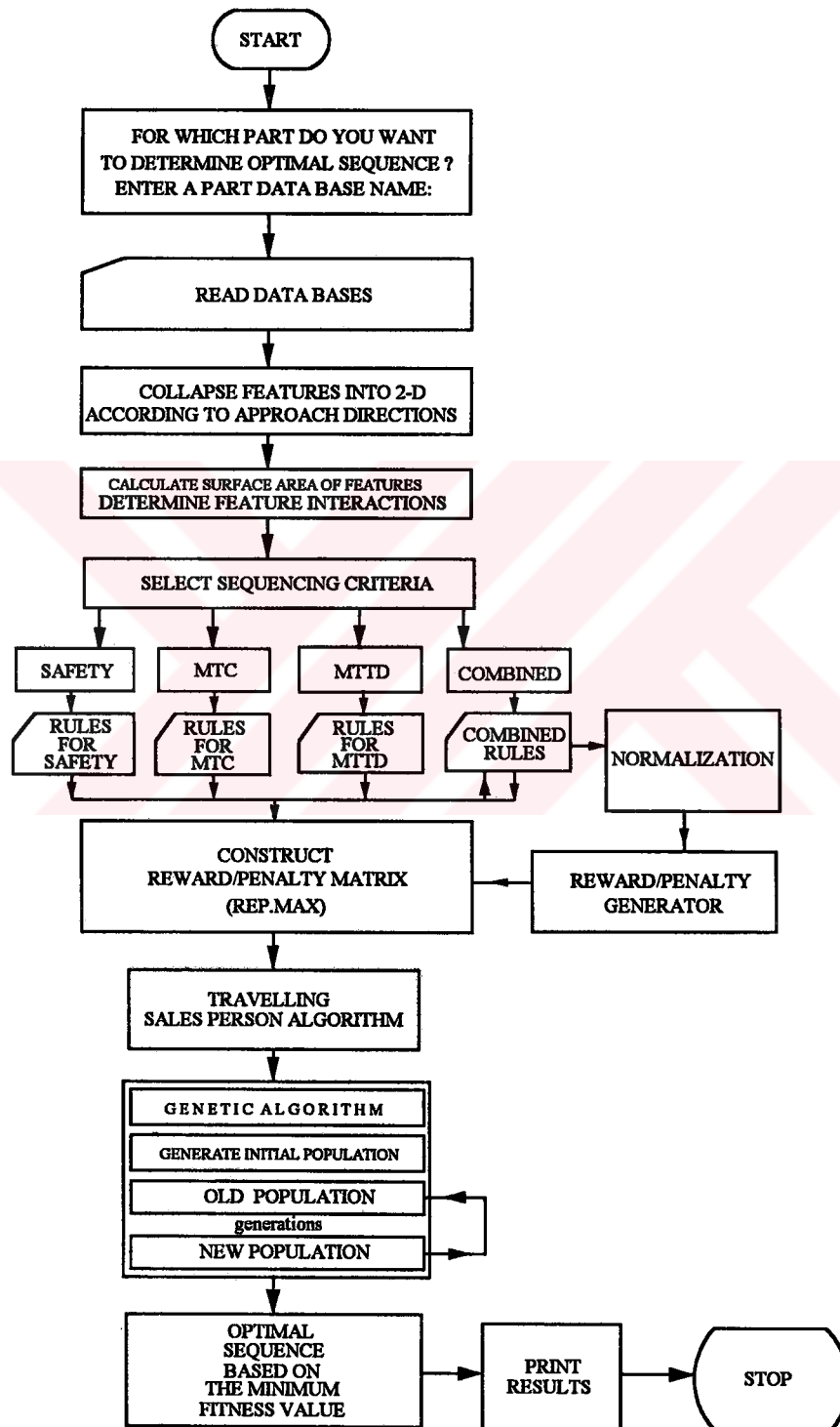


Figure 7.1. Flowchart of the GA Based Optimisation System [165]

The next stage is the selection of criterion(s) to be used in the optimisation. For each type of criterion, a *rule base* is prepared. The rules for MTC and MTTD are quite clear. Some of the rules for the safety criterion implemented in the system are as follows;

-
- *RULE 1: if <two features partially intersect>, then <machine the feature with smaller surface area first>,*
 - *RULE 2: if <two features are nested inside one another>, then <machine the top-most feature first>,*
 - *RULE 3: if <there is an edge cut on the part>, then <it should always be machined last>, etc.*
-

Figure 7.2 shows that RECTANGULAR POCKET partially intersects with THRU HOLE-1 which has a smaller surface area than the pocket. It is therefore safe to assume that THRU HOLE-1 is to be machined before the RECTANGULAR POCKET. If this constraint is not applied (i.e. when machining the hole after the pocket), a tool slippage may occur because of the missing area. Figure 7.2 also illustrates the usefulness of the second safety constraint which requires that the RECTANGULAR POCKET be machined before THRU HOLE-2 in order to avoid machining deep features. Machining of deep features is too difficult, since a long cutting tool may be subjected to bending and breaking. Therefore, the third constraint is active here.

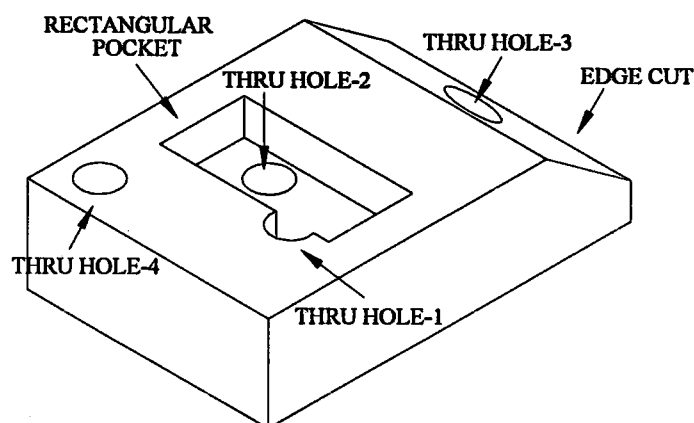


Figure 7.2 Interaction Between a Thru Hole and a Face Pocket

The last interesting feature intersection in Figure 7.2 is between THRU HOLE-3 and EDGE CUT. If the second constraint is applied to this situation, the EDGE CUT is to be machined first. However, machining the EDGE CUT before the THRU HOLE-3 would remove the perpendicular surface for THRU HOLE-3. The removal of this surface makes it harder to machine THRU HOLE-3 as the tool is likely to slip-off the surface of the edge cut, especially if the starting block is made of a hard material.

Construction of the reward/penalty matrix (REPMAX) based on the rules of selected criterion is an important stage of the methodology. Each rule has a penalty or reward (negative penalty). The REPMAX for each set-up is automatically prepared by a reward/penalty generator. The penalty scale used in the preparation of REPMAX is as follows; if precedence of operations such as *i* and *j* satisfies the sequencing rules, according to degree of satisfaction of rules, a reward (negative penalty) is given. If precedence of operations does not satisfy the sequencing rules, according to degree of non-satisfaction of the rules, a positive penalty is given. For example, an edge cut must normally be machined last due to safety criteria. In a precedence relationship of two operations, if an edge cut is to be machined first, its penalty would be a huge positive penalty (a positive value) in the REPMAX.

At this stage, TSP algorithm takes the REPMAX as the input and gives closed-end solutions (strings of features sequence in which the relationship between the last and first operation is also considered) to be used in the initial population of genetic algorithm together with other randomly generated strings. Number of strings from TSP is equal to total number of features on the component. Other members of initial population of candidate sequences are generated by a random number generator which is initialised with the randseed parameter. Total number of strings in the initial population is 200. The random number generator is also used to select the mutation points that are used to generate the initial population. New sequences can be generated by using the genetic operators; *partially matched crossover*, *reproduction* and *mutation* in different combination of generation cycles. The internal mechanism of these operators were previously explained in Chapter 3. A *fitness function* is then used to evaluate the goodness of each sequence in terms of penalties or rewards

specified in the REPMAX between the feature pairs. Based on the fitness values, the next generation is formed from the newly generated sequences and old population such that it includes 80% of good sequences and 20% of bad sequences. After a while, the better sequences with higher fitness values dominate in the population and the system eventually converges to an optimal solution. The GA cycle is terminated when total number of iterations specified by the user is reached. The number of iterations depends on the size of the problem (i.e., the # of operations to be sequenced).

An example part and its top view are illustrated in Figure 7.3 and Figure 7.4, respectively. CAD data base of the part is given in Table 7.1. The CAD data base includes the type and *label* of features, dimensions (width, radius, length and depth) of features and the distances dx and dy (in x and y axis, respectively) between the main datum on the body which is always at lower-left corner and the features' datum. The datum for hole features are located on their center coordinates, while they are located on the lower left corner of the pockets and slots. For step features datum they are considered to be on the inner corner of the feature. Figure 7.5 illustrates how to measure the distances (dx and dy) between a hole and a rectangular pocket and the main datum on the component in x and y axes.

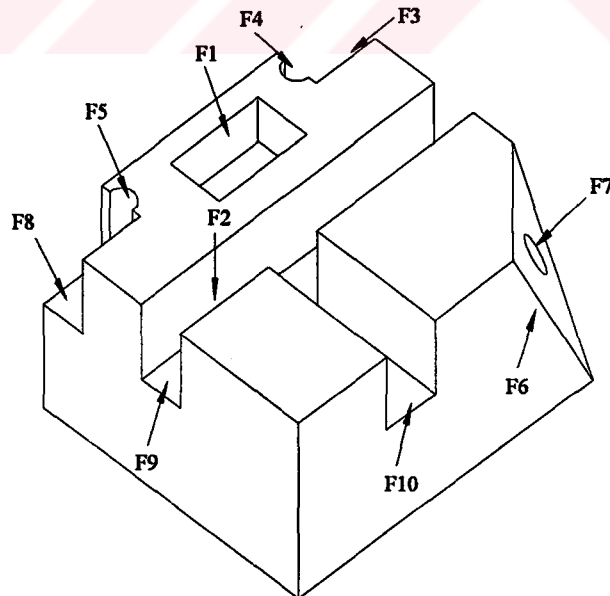


Figure 7.3. Isometric View of the Sample Part

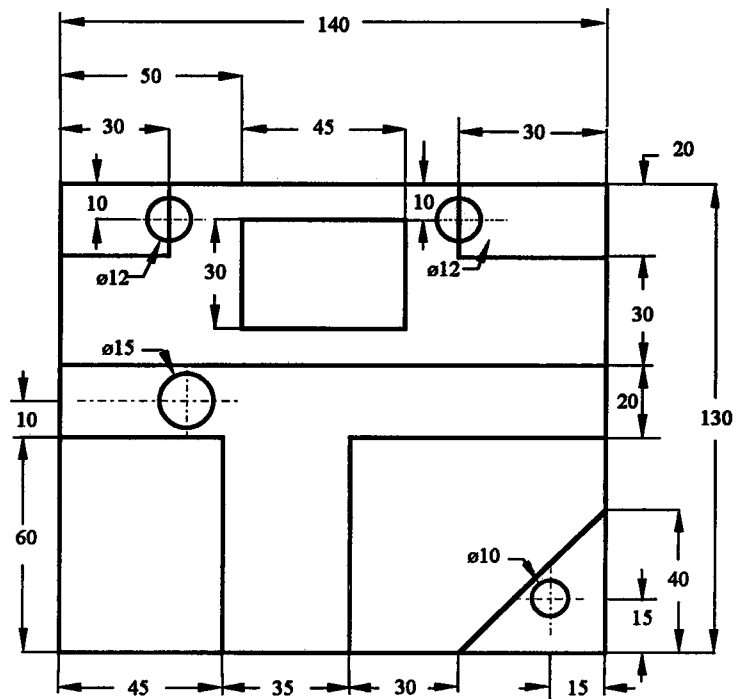


Figure 7.4. Top View of the Sample Part shown in Fig. 7.3

Table 7.1. CAD Data Base of the Sample Part Shown in Fig. 7.3

Feature No	Feature Type	dx mm	dy mm	Width or Radius, mm	Length mm	Depth mm	Projected Area, mm ²	Volume mm ³
F1	R-PKT	50	90	45	30	20	1350	27000
F2	T-HOL	35	70	7.5	-	25	176.71	4418
F3	B-STP	110	110	20	30	25	600	15000
F4	T-HOL	110	120	6	-	35	113.1	3960
F5	T-HOL	30	120	6	-	35	113.1	3960
F6	E-CUT	110	0	40	40	30	1600	48000
F7	B-HOL	125	15	5	-	25	78.54	1960
F8	B-STP	30	110	20	30	25	600	15000
F9	T-SLT	0	60	20	140	25	2800	70000
F10	B-SLT	45	0	35	60	25	2100	52500

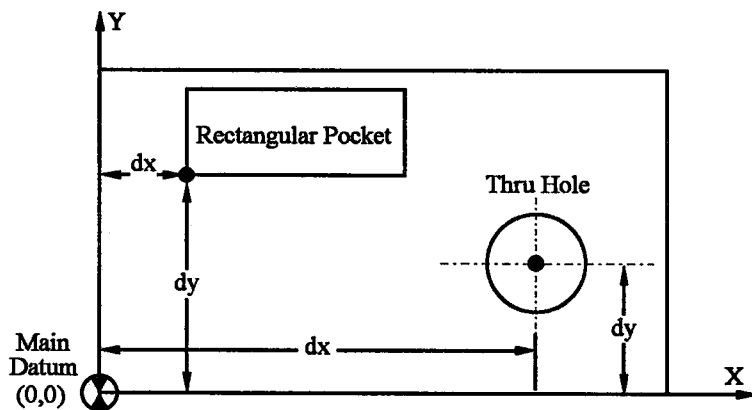


Figure 7.5. Distances Between Features' Datum and the Main Datum

In this example, *safety* is selected as the sequencing criterion. Consider the precedence relationship between F6 (E-CUT) and F7 (B-HOL). The sequence from 7 to 6 violates the rule; “*edge cut would always be machined last*”. So such a sequence is not preferable and it must be punished by a highly positive penalty in the reward/penalty matrix (REPMAX). As can be seen in Table 7.2, the relation between 6 to 7 (not 7 to 6) that is REPMAX (6,7), is described by a positive penalty (105). Other relations, and their penalties are also shown in Table 7.2. Notice that REPMAX (7,6) is equal to -95 which promotes the sequences in which F7 is before F6. Penalties of no relations are taken as 5 due to nature of the algorithm. In an anticipated optimal sequence, F5 must be machined before F8, F4 must be machined before F3, F2 must be machined before F9, F7 must be machined before F6, for satisfying the safety rules.

Table 7.2. Reward/Penalty Matrix (REPMAX) of the Sample Part shown in Fig. 7.3

FN	1	2	3	4	5	6	7	8	9	10
1	∞	5	35	25	25	5	5	35	45	5
2	5	∞	5	5	5	5	5	5	-45	5
3	-25	5	∞	85	5	5	5	5	5	5
4	-15	5	-75	∞	5	5	5	5	5	5
5	-15	5	5	5	∞	5	5	-75	5	5
6	5	5	5	5	5	∞	105	5	5	5
7	5	5	5	5	5	-95	∞	5	5	5
8	-25	5	5	5	85	5	5	∞	5	5
9	-20	55	5	5	5	5	5	5	∞	-15
10	5	5	5	5	5	5	5	5	25	∞

Table 7.3. The Sequences from TSP Algorithm for Sample Part Shown in Fig. 7.3

Sequence of Features	Closed-End Fitness Value	Open-End Fitness Value
7-6-4-3-5-8-1-2-9-10	-310	-315
5-8-1-2-9-10-7-6-4-3	-310	-315
4-3-5-8-1-2-9-10-7-6	-310	-315
2-9-10-7-6-4-3-5-8-1	-310	-315
8-1-2-9-10-7-6-4-3-5	-310	-235
9-10-7-6-4-3-5-8-1-2	-310	-265
10-7-6-4-3-5-8-1-2-9	-310	-295
6-4-3-5-8-1-2-9-10-7	-310	-215
3-5-8-1-2-9-10-7-6-4	-310	-235
1-2-9-10-7-6-4-3-5-8	-310	-285

The strings of sequences found by TSP algorithm upon inputting the REPMAX is given in Table 7.3. The TSP problem is well known and detailed information on its solution techniques can be found in various sources [166-168]. Therefore, implementation of the TSP algorithm is not included here.

The next stage is the execution of the genetic based algorithm. The first step is to randomly generate an initial population of sequences. The results obtained in the TSP module as shown in Table 7.3 are also included in the initial population. A section of the initial population is shown in Table 7.4.

Table 7.4. Randomly Generated Initial Population of Sequences (Parents)

Chromosome No	Sequence of Features
1	1-2-5-3-10-7-6-4-9-*8*
2	10-3-9-1-2-8-4-5-6-*7*
3	8-9-*5-6*-1-3-10-7-2-4
4	5-3-*8-6*-2-7-10-1-4-9
...	...
Strings were taken from the TSP module	7-6-4-3-5-8-1-2-9-10
	5-8-1-2-9-10-7-6-4-3
	4-3-5-8-1-2-9-10-7-6
	2-9-10-7-6-4-3-5-8-1
	8-1-2-9-10-7-6-4-3-5
	9-10-7-6-4-3-5-8-1-2
	10-7-6-4-3-5-8-1-2-9
	6-4-3-5-8-1-2-9-10-7
	3-5-8-1-2-9-10-7-6-4
	1-2-9-10-7-6-4-3-5-8
...	...
197	5-7-6-2-*3*-4-1-9-10-8
198	6-7-8-5-*4*-1-2-3-9-10
199	4-7-*5-8-9*-2-10-6-3-1
200	8-6-*9-10-4*-1-7-3-5-2

Table 7.5. Sequences Derived by the Genetic Operators (Children)

Chromosome No	Sequence of Features
201	1-2-5-3-10-8-6-4-9-7
202	10-3-9-1-2-7-4-5-6-8
203	5-9-8-6-1-3-10-7-2-4
204	8-3-5-6-2-7-10-1-4-9
...	...
...	...
397	5-7-6-2-4-3-1-9-10-8
398	6-7-8-5-3-1-2-4-9-10
399	5-7-9-10-4-2-8-6-3-1
400	10-6-5-8-9-1-7-3-4-2

New chromosomes (children) are then generated from the initial population (parents) by using the genetic operators. The children obtained from the parents are shown in Table 7.5. The first operator used in the genetic algorithm is *partially matched crossover operator* which requires two crossover points (shown by ‘*’ in Table 7.4) are randomly chosen.

Generation of child chromosomes 201 and 202 from the parent chromosomes 1 and 2 is as follows. Two random crossover points (i.e. 9 and 10) are picked along the sequence (by ‘*’). The *features* at the crossover points are switched between the parents and then children are generated. We have [Parent1:1-2-5-3-10-7-6-4-9-8] and [Parent2:10-3-9-1-2-8-4-5-6-7]. The crossover operator produces the following children; [Child1:1-2-5-3-10-7-6-4-9-7] and [Child2:10-3-9-1-2-8-4-5-6-8]. These intermediate sequences are not valid, since some of the features appear more than once (7 in Child 1 and 8 in Child 2). The children are validated (modified) to produce valid sequences from the invalid children, making sure that the features at the crossover points are not changed and feature appears only once in a sequence. The final result is then: [Child1:1-2-5-3-10-8-6-4-9-7] and [Child2:10-3-9-1-2-7-4-5-6-8].

As the execution of the genetic algorithm reaches to certain number of iterations, the other genetic operators are also used in generating child chromosomes. At the end of the each iteration, the goodness of each sequence (parents and child chromosomes) in terms of penalties or rewards specified in the REPMAX is evaluated by using a *fitness function*. The fitness function (objective function) can be expressed as a mathematical function by the following form;

$$f_{\text{chromosome}} = \sum_{i=1, j=2}^{i=n-1, j=n} \text{REPMAX}[\text{gene}[i]][\text{gene}[j]] \dots\dots\dots (7.1)$$

where; n is the total number of genes in a chromosome or in other words is the total number of features on the component, and *gene* is the vectorial representation of genes in a single chromosome. For example, the fitness value of *chromosome 1* that is [1-2-5-3-10-7-6-4-9-8] given in Table 7.4 is calculated as -45;

REPMAX	[gene[1]]	[gene[2]]	=	REPMAX	[1][2]	=	5	+	
REPMAX	[gene[2]]	[gene[3]]	=	REPMAX	[2][5]	=	5	+	
REPMAX	[gene[3]]	[gene[4]]	=	REPMAX	[5][3]	=	5	+	
REPMAX	[gene[4]]	[gene[5]]	=	REPMAX	[3][10]	=	5	+	
REPMAX	[gene[5]]	[gene[6]]	=	REPMAX	[10][7]	=	5	+	
REPMAX	[gene[6]]	[gene[7]]	=	REPMAX	[7][6]	=	-95	+	
REPMAX	[gene[7]]	[gene[8]]	=	REPMAX	[6][4]	=	5	+	
REPMAX	[gene[8]]	[gene[9]]	=	REPMAX	[4][9]	=	5	+	
REPMAX	[gene[9]]	[gene[10]]	=	REPMAX	[9][8]	=	5	=	-45

The next step is to put all the 400 chromosomes (200 parent chromosomes from the initial population and 200 chromosomes generated by the genetic operators) in a descending order starting from the one which has the most negative cumulative fitness. Based on the fitness values, the next generation (current population) is formed from the newly generated sequences and old population such that it includes 80% of good sequences and 20% of bad sequences. The iterations are continued by this way. GA seeks to find sequence with the most negative cumulative fitness. After a while, the better sequences with highly negative fitness values dominate in the population and the system eventually converges to an optimal solution.

For this relatively simple example, the GA finds optimal solutions which have a fitness value of -315 (more than fifty solutions) in 3 to 10 iterations. Some of these solutions are already given by the TSP algorithm. As shown in Table 7.3, the open-end fitness value of the best four strings are also equal to -315. However, for highly difficult components comprising many interacting features, the proposed GA finds optimal sequences which are better than TSP algorithm's strings.

Table 7.6. Reward/Penalty Matrix Prepared for a Highly Difficult Part

<i>FN</i>	1	2	3	4	5	6	7	8	9	10
1	∞	-30	-20	-10	-10	-10	-10	-10	-10	-10
2	50	∞	-10	-5	-5	-5	-5	-5	-10	-20
3	40	10	∞	-30	-50	-10	5	10	15	20
4	45	-45	-5	∞	4	40	-25	60	-30	-10
5	100	40	60	55	∞	5	20	-10	-20	5
6	100	70	-50	-15	40	∞	40	20	-5	10
7	60	5	30	-25	5	5	∞	-15	30	20
8	90	30	-5	10	-5	-35	80	∞	15	-25
9	-75	10	-5	5	-40	40	20	-90	∞	-5
10	80	30	-20	45	75	25	-30	30	35	∞

Table 7.7. The Sequences from TSP Algorithm Based on Table 7.6

Sequence of Features	Closed-End Fitness Value	Open-End Fitness Value
9-1-10-7-4-2-8-6-3-5	-345	-325
6-3-5-9-1-10-7-4-2-8	-345	-310
10-7-4-2-8-6-3-5-9-1	-345	-335
3-5-9-1-10-7-4-2-8-6	-345	-295
4-2-8-6-3-5-9-1-10-7	-345	-320
7-4-2-8-6-3-5-9-1-10	-345	-315
8-6-3-5-9-1-10-7-4-2	-345	-340
1-10-7-4-2-8-6-3-5-9	-345	-270
5-9-1-10-7-4-2-8-6-3	-345	-295
2-8-6-3-5-9-1-10-7-4	-345	-300

For instance, for a part with highly interacting features, the reward/penalty matrix is given in Table 7.6. The best string of the TSP algorithm is [8-6-3-5-9-1-10-7-4-2] which has an open-end fitness value of -340 (See Table 7.7). However, the optimal sequence found by the proposed GA is; [1-10-7-4-2-9-8-6-3-5] or [1-6-3-5-9-8-10-7-4-2] that has an open end fitness value of -345. This result can be achieved by performing on the average 40 iterations completed in 4 minutes. It is worth noting that the total number of iterations required to reach an optimal value is fluctuating from 1 to 100, as the execution time varies from 1 to 10 minutes, since the iteration process is based on a random number generator.

7.3. AN ALTERNATIVE APPROACH TO THE OPTIMISATION OF SEQUENCE OF OPERATIONS: RTSS -Right Tool Selection System

GA based system developed for optimisation of sequence of operations which is discussed in the previous section may find optimal sequences. However, a long time is required for the optimal answer, especially for large operation sets. It may also not converge to an optimal solution when minimum tool change constraint is used alone. Therefore, an alternative heuristic-based approach for the optimisation of sequence of operations is developed called RTSS.

The effect of tool replacement is not included in the operation sequencing and initial tool assignment. However, the effect of tool replacement should also be considered to provide a global optimisation in a machining cycle of a component and in order to minimise the non-cutting time. The goal of RTSS is to find an optimal operation sequence which is characterised by a *minimum number of tool changes within a set-up*. When the specifications of candidate cutters are determined, they are placed in a matrix (called Tool Matrix) to express the relations between the features and the tools, along the horizontal axis the operations and along the vertical axis the cutters. In this way, it is possible to express that different operations can use the same tool and that for one feature alternative tools exist.

Consider the part in Figure 7.6. On this part, there are two features to be machined; a *step* and a *side slot*. Assume that the target machine is a vertical milling centre. The cutter-type domain of the step feature includes a shoulder mill, an end mill and a disc mill, while cutter-type domain of side-slot includes only a disc mill.

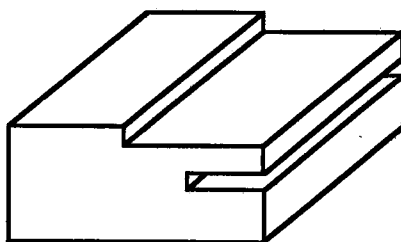


Figure 7.6. A Part with Two Features: a *step* and a *side-slot* [169]

To obtain an optimal operation sequence with the minimum number of tool changes, both features should be machined with the same tool. This is possible only if the disc mill used for the side-slot feature would also be used for machining the step feature. RTSS uses following rules to select the tool for a feature between alternative tools [49];

-
- *select the tool which is used by the most successive operations*
 - *if there are alternatives, choose the tool which is used for one of the previous operations*
 - *if there are still alternatives, chose the tool that can be used for the largest number of operations*
-

7.3.1. Methodology

Operations determined in “*operation selection and sequencing module*” are placed in the same order along the horizontal and vertical axes of a matrix called *Priority Matrix* by using the technique adopted in Reference [43,49]. If an operation *i* has to be machined before operation *j*, then “1” is placed at row *i* and column *j*, in the priority matrix. An example of priority matrix is given in Table 7.8.

Table 7.8. Priority Matrix [49]

Operations→ ↓	1	6	9	10	2	3	8	4	5	7	UM	RM
1		1			1						1	1
6				1				1			2	3
9				1	1						1	3
10							1			1	4	6
2						1				1	4	5
3							1		1		6	6
8								1	1		7	7
4											8	10
5											8	10
7											6	10

The last columns; UM and RM (Up Movement and Right Movement) are the ranges that an operation could be placed in. The UM is determined from the bottom, until the first “1” in that column is reached. So, UM is equal to row number found plus one. The right movement is approached from the left to right until first “1” in that row is reached. Hence, RM is equal to column number found minus one.

The matrix including cutting tool(s) versus operations is given in Table 7.9. This matrix is actually a systematic demonstration of the cutter-type domains for the operations defined in the tool assignment stage of the process planning session.

Table 7.9. Tool(s) versus Operations

Operations→ Tools ↓	1	6	9	10	2	3	8	4	5	7
1	1					1				
2		1	1			1		1	1	
3		1								
4				1			1			1
5			1	1	1					1
6		1	1					1	1	
7							1			

The next stage is the assignment of a single tool to each operation by using the heuristic-based tool selection rules discussed in the above paragraphs. The selected tools for the operations among the alternatives given in Table 7.9 are shown in Table 7.10.

Table 7.10. Tool versus Operations

Operations→ Tools ↓	1	6	9	10	2	3	8	4	5	7
1	1					1				
2										
3										
4							1			
5			1	1	1					1
6		1						1	1	
7										

By taking UM and RM values and selected tools into account, *that is, if the movement of an operation in the operation list is possible and its tool is suitable to the operation(s) to be moved*, operation(s) which use(s) the same tool are grouped in minimum number so that the number of tool change is minimised while precedence rules in the sequence of operations are not violated. In other words, this means that if an operation is shifted, this could have the consequences for the number of tool replacement. In this example, the UM and RM value of the *Operation 7* is suitable for the movement of this operation between the *Operation 2* and *Operation 3*. When this movement is performed, an optimal block structure is obtained in the *cutting tool versus operations matrix* as shown in Table 7.11. With this optimisation, it should be noticed that the number of tool changes is reduced from 6 (See Table 7.10) to 5 (See Table 7.11).

Table 7.11. Optimal Block Structure: Optimal Sequence of Operations
with Their Assigned Tools

Operations→ Tools ↓	1	6	9	10	2	7	3	8	4	5
1	1						1			
4								1		
5			1	1	1	1				
6		1							1	1

7.4. OPTIMISATION OF TURRET INDEXING POSITIONS BY USING A GENETIC ALGORITHM

Determination of optimal positions of cutting tools on the turret or magazine of CNC machine tools is an important task for achievement of optimal process plans. Especially for those machines that cannot provide a fast turret-indexing time, it is extremely important to decrease the total tool-indexing time which directly affects total non-cutting time. One of the main goals of this module is therefore to reduce the *non-machining* time. Although today's machine tool manufacturers furnish their machines with superior and faster turret indexes, automatic tool changers or magazines in order to increase the time in cut by maximising their movement speeds, turret indexing time should be further reduced by applying an effective tool magazine arrangement policy [170].

This section presents the module prepared for determining optimal turret-index positions of cutting tools that are assigned to machining operations. This module is also an integrated element of OPPS-PRI. Position selection is performed using a genetic algorithm which leads to least total tool-indexing time. The algorithm takes a list of cutting tools classified with certain numbers assigned to machining operations together with the total number of positions available on the turret and the catalogue value of indexing time (from tool to tool) specified in the manuals of CNC machine tools as the input and generates an initial population of positions (chromosomes). The type of the turret (tool changer) such that whether it is uni-directional or bi-directional is also considered. HITEC Turning Center produced by Hitachi Seiki Co., Ltd. [171] has a turret indexing time of 0.1 sec (the time moving between two neighbouring tools, from tool to tool, in other words; from face to face) for a 12-station turret. Compact Universal CNC lathe Hi-Eco10 produced by Hwacheon [172] has a higher value; 0.2 sec for a 10-station turret. TAKSAN TMC-700 VMC which has a 12-station ATC (Automatic Tool Changer) provides a turret indexing time of 0.69 sec [173]. They all offer a bi-directional turret index or ATC. Turret indexing time from tool to tool (t_i) is described in Figure 7.7.

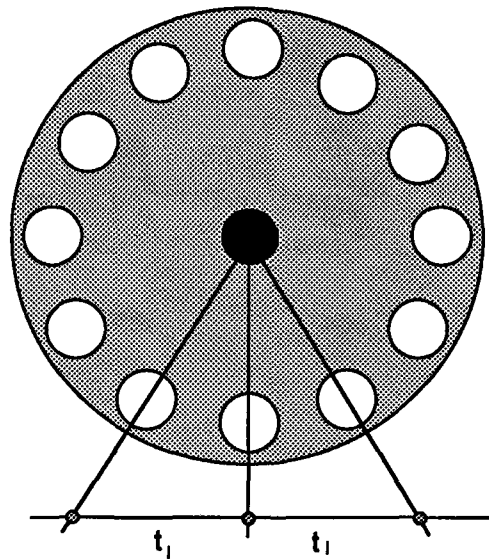


Figure 7.7. Turret Indexing Time (From Tool to Tool), t_i

The methodology used in the optimisation of turret indexing time is similar to that used in the optimisation of sequence of operations via a genetic algorithm. The only difference between these two applications is the use of different objective functions (fitness functions) to be minimised. Instead of the objective function given by Equation 7.1 which is used to minimise the total rewards/penalties for a given chromosome (set of operations), a simple objective function is used in this case in order to minimise the total turret indexing time for a given chromosome (set of cutting tools). The value of the objective function can easily be calculated by multiplying the total number of unit rotations between the indexes with the turret indexing time given in the machine tool manuals.

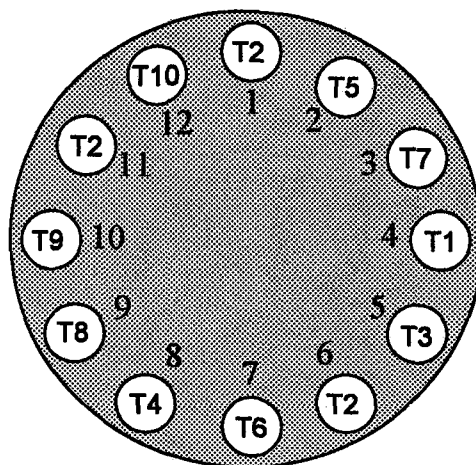


Figure 7.8. A Typical Tools Arrangement on a 12-Station ATC

Figure 7.8 shows a typical arrangement of cutting tools indexed on a 12-station ATC. The representation of this arrangement in the GA, i.e., a typical chromosome of cutting tools for a 12-station turret is as follows;

Turret Position	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
Chromosome of Cutting Tools	T2	T5	T7	T1	T3	T2	T6	T4	T8	T9	T2	T10

New chromosomes of cutting tools are generated from the initial population (parents) by using the genetic operators. Fitness values of the chromosomes are then calculated and based on the fitness values, the next generation is formed from the newly generated chromosomes of tools and old population. As the iterations of the genetic algorithm continues, the better tooling arrangements (chromosomes) with lower total indexing time dominate and the system eventually converges to an optimal positional set of cutting tools.

When the program is designed, the *tool arrangement problem* is handled in three phases in terms of the relation between the total number of cutting tools and the total number of available index positions on the ATC of machine tool to be used [170];

-
1. The number of cutting tools is equal to number of index positions on ATC
 2. The number of cutting tools is smaller than the number of index positions on ATC
 - a) without duplicated tools
 - b) with duplicated tools
 3. The number of cutting tools is higher than the number of index positions on ATC
-

The overall aim is to minimise the total manufacturing cost by reducing the tool operating or tooling cost with the use of different tool indexing policies like *tool sharing* and loading *duplicate tools*. If the problem falls into *Phase 1*, there is no need to duplicate the cutting tools in the tooling set to avoid the second ATC set-up which increases the total non-machining time considerably. If the total number of the cutting tools that are assigned for fully machining a component, are smaller than the total

number of available index positions on the ATC of a machine tool (*Phase 2*), then the effect of the duplicated tools on a possible decrease in the tool indexing time should be tested. For example, certain cutting tools can be duplicated on the ATC, so in the chromosome of cutting tools in GA as well. In the case of *Phase 2*, the problem is somewhat different, so it changes to selecting the tools to be used (shifted) in the second set-up. At the moment, *Phase 1* and *Phase 2.a.* are made available to the user. *The other phases (Phase 2.b and Phase 3) are under development.* The example arrangements of cutting tools on a 12-station ATC for each phases are given in Figure 7.9 to Figure 7.12.

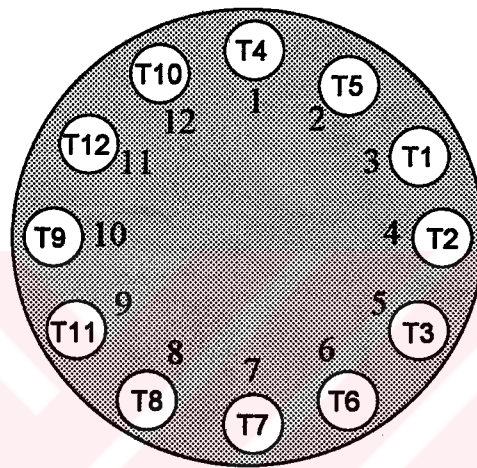


Figure 7.9. ATC Arrangement; # Tools = # Indexes (Phase 1)

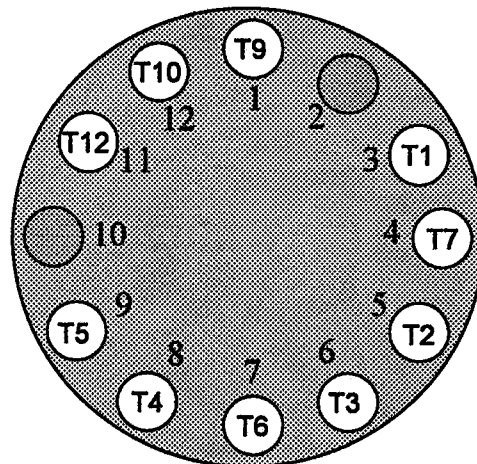
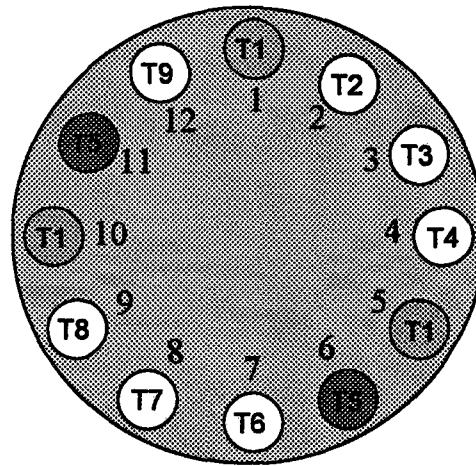


Figure 7.10. ATC Arrangement; # Tools < # Indexes (Phase 2.a)



Tool 1 is duplicated twice, Tool 5 is duplicated once

Figure 7.11. ATC Arrangement; # Tools < # Indexes (Phase 2.b)

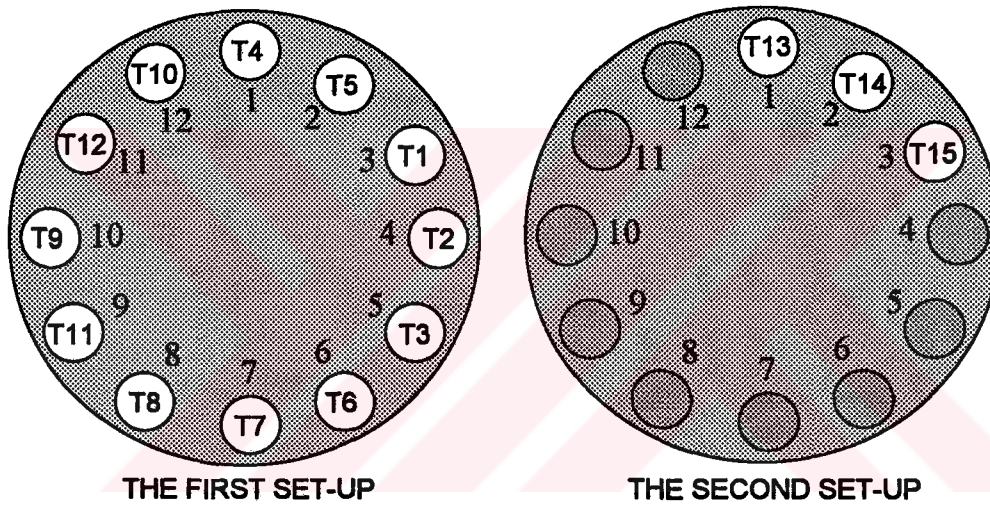


Figure 7.12. ATC Arrangements; # Tools > # Indexes (Phase 3)

However, it should also be noticed that there may also be other *sub-phases* between the tool set-up phases specified above. For instance, re-consider the *Phase 2.b* described with the Figure 7.11. In Figure 7.11, the duplicated tools are used in such a way that no unloaded index is left on ATC. However, there is another case where the optimal arrangement of cutting tools may require an ATC organisation in which an index (or more than one index) is left unloaded.

GA is an effective tool for the tool arrangement problem described above. It is very difficult to solve the problem using *Brute-force Search* algorithm, since there are many alternatives to be controlled and they require a long time. For instance, there are

479 millions alternative tooling set-ups that can be used for a 12-station ATC and a tool set of 12 cutting tools. As the total number of cutting tools employed in machining and the ATC capacities increase, the benefit to be obtained from the right tool arrangement would have a great importance. Even 1 second save in the tool indexing time would reduce the total production time reasonably, *especially in mass production*. For example, if a 1.000.000 part is produced per year, 1 second save in the tool indexing time per part will cause a gain of about 12 working-days.

7.4.1. An Example

A machine tool data base is prepared by using the manuals of various CNC machine tool manufacturers. When the user selects the machine tool, parameters like tool capacity of the ATC, type of ATC movement and indexing time are captured and fed into the optimisation software. The second and the main input of the software is the list of cutting tools assigned to the machining operations which is derived and prepared by the up-stream modules of OPPS-PRI, namely, tool selection module and operation selection module. For ease of manipulations, both of the machining operations and corresponding cutting tools to be used are characterised by numbers. Their accompanying labels or specifications are also stored. An example of an input list is given in Table 7.12. The other inputs are;

Tool Capacity of ATC = 10,

Type of ATC movement = Bi-directional

Indexing Time Between Two Adjacent Slots = 0.20 second

Table 7.12. Cutting Tools Assigned to Machining Operations

Operations	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Cutting Tools	1	2	3	4	5	6	3	6	7	3	8	9	6	9	10

The problem described here falls into *Phase 1* where the total number of cutting tools employed is equal to the total number of index positions available on ATC; 10. Therefore it is not possible to duplicate any tool on the ATC. The total number of operations is found to be 15.

The set-ups of cutting tools proposed by GA based optimisation software for different number of iterations are given in Table 7.13.

Table 7.13. Proposed Solutions by GA

No	# Iterations / Elapsed Time	Positions on ATC										Fitness Value: # of Unit Rotation	Fitness Value: Total Index Time
		1	2	3	4	5	6	7	8	9	10		
1	1 / 6 sec.	9	8	5	6	7	3	4	2	1	10	27	5.4 sec
2	2 / 12 sec.	9	8	5	6	7	3	4	2	1	10	27	5.4 sec
3	3 / 17 sec.	9	8	5	6	7	3	4	2	1	10	27	5.4 sec
4	6 / 26 sec.	7	6	9	10	8	1	2	4	5	3	26	5.2 sec
5	10 / 35 sec.	10	8	9	6	7	3	5	4	2	1	25	5.0 sec
6	15 / 50 sec.	7	3	6	8	9	10	5	4	1	2	25	5.0 sec
7	30 / 80 sec.	10	9	8	7	6	3	4	5	2	1	25	5.0 sec
8	100 / 233 sec.	2	5	4	3	7	6	9	8	10	1	25	5.0 sec

Note that GA converges rapidly to an optimal solution; 25 unit rotations of ATC or in other words a total turret indexing time of 5 seconds. The above problem is also asked to more than 10 average technicians, workers and operators. Average value of total unit rotations of ATC and indexing time obtained from this quiz is equal to 30 and 6 seconds. Even this small sized problem the gain is 1 second per component to be produced. It should be noticed that indexing time of the machine tool is also important. If the size of the problem increases, the gain obtained from GA will also increase.

7.5. THE OPTIMISATION MODULE FOR CUTTING PARAMETERS:

CPOS - Cutting Parameters Optimisation System

Determination of optimal cutting parameters *including number of passes, depth of cut for each pass, speed, and feed applicable for selected cutting tools* is a crucial module in process planning, since the economy of machining operations plays an important role in increasing productivity and competitiveness.

Since CNC machine tools are extensively employed in manufacturing industry, *economic machining* has gained a great importance. As everyone accepts, CNC machine tools have eliminated the auxiliary tooling and reduced the set-up times considerably. However, it is not possible to run the CNC machine tools effectively and economically without using *optimised machining parameters*. The machining time is almost the same for both the CNC machining and conventional machining (because of the unchanged cutting principles and mechanisms), if the same machining parameters (conditions) are used. In conventional machining, the usual application was to select cutting parameters from machining databases or handbooks. The ranges given in the machining data handbook or catalogue recommendations specified for a large variety of *cutting tool material versus workpiece material* combinations are actually starting values and are not the optimised values. However, CNC machine tools are more expensive than the conventional machines, therefore there is a strong need to operate the CNC machine tools as efficiently as possible in order to obtain the required pay-back. The optimal values of cutting parameters ought to be determined before loading a part on the CNC machines and starting the production cycle.

Up to date, many works have been done to optimise the cutting parameters. A literature survey has been performed on the optimisation of cutting conditions. It was seen that the most of the works done on the subject are restricted to turning operations, although milling operations play significant roles in today's manufacturing world with the CNC milling machines/centers. They rely on methods which neither guarantee the optimal solutions nor provide clearly defined economic characteristics of the optimisation problem [174]. Most of them used a single constraint "*power*" in their optimisation strategies while ignoring other constraints like "*surface finish*".

Therefore, there is still a need to develop an optimisation system for determining the best values of cutting parameters for milling operations.

In this work, an optimisation system software called CPOS (Cutting Parameters Optimisation System) has been developed for determining the optimal values of cutting parameters. It is also integrated with the process planning system; OPPTS-PRI.

CPOS was initially constructed on a methodology using geometric programming plus dynamic programming techniques. However, this initial stage of the work was restricted only to face and plain milling operations due to lack of availability of the parameters, constants and empirical mathematical models for the other type of milling operations. A detailed discussion on the developed strategy can be found in [175]. The system was previously based on *maximum production rate criterion*. One of the disadvantages of this initial system is that the execution speed was low due to large empirical mathematical models involved in the algorithm and the large amount of derivatives.

When the work continued, it has been recognised that the *AI* tools can also be used in the development of cutting parameters optimisation system, and then CPOS has been modified. The modification includes the replacement of the implementation tool; *using a GA instead of geometric programming* that eliminates extensive mathematical operations by increasing the processing speed. *Minimum production cost* is also considered as an objective function (criterion) as well as the *maximum production rate* criterion. If desired, other objective functions like *maximum profit rate* can also be used, since CPOS has been designed as a system open to further modifications.

CPOS has a multi-pass optimisation strategy incorporating several technological constraints such as power, surface finish, speed, feed limitations, etc. It makes use of a method called *volume sectioning* and a *genetic algorithm*. GA proposed here is similar to the algorithm that is proposed for the optimisations of turret indexing time and sequence of operations. Here, the genetic chromosomes

represent the feed and speed strings and are defined in binary coding. They also require normalising, since the operating scales of them are different.

7.5.1. Optimisation Model

Three basic cutting parameters are usually considered (optimised) in the optimisation of milling operations. They are;

- optimal number of passes and depth of cut for each pass
- optimal cutting feed(s)
- optimal feed rate(s)

Among these, *depth of cut* is the dominant parameter which is determined in upstream process planning modules like operation selecting. Although it is preferable to machine the features or volumes in single pass in order to reduce the cost and time of machining (if possible), this is usually unachievable. The reason is that the machining operations are constrained by not only the cutter but also by other constraints imposed by the machine tool (by the characteristics of the feed drive and main drive systems) and the workpiece. This means that although the *total depth of cut* for machining the features are determined in the upstream values of process planning, they may require more than one pass, due to unavailability of the cutting tools that can provide the required cutting edge or due to low power capacity of available machine tool, etc. It should be noticed that the multi-pass scheme inherently includes the single-pass option, as well.

7.5.2. Derivation of the Objective Functions

The unit cost (C_u) can generally be represented by the sum of four cost terms;

- cost of raw material
- cost of set-up
- cost of machining
- cost of tool changing

Set-up cost and *machining cost* are the sum of the corresponding labour and overhead costs. On top of labour and overhead cost, the cost of the cutting tool is added in the case of the *tool changing cost*. However, *tool changing cost* is rationalised with the machining time divided by tool life, since a cutting tool would have been replaced before the machining operation takes place. Hence, the unit cost can be represented by following formula;

$$C_u = c_{mat} + (c_l + c_o)t_s + (c_l + c_o)t_m + (c_l t_{tc} + c_t + c_o t_{tc}) \left(\frac{t_m}{T} \right) \dots\dots\dots (7.2)$$

Note that all symbols used in the formulation of the objective function given in Equation 7.2 and forthcoming equations are given in Table 7.14.

Machining time for usual milling operations can be calculated by following equation;

$$t_m = \frac{K}{F} = \frac{K}{fzN} \dots\dots\dots (7.3)$$

Spindle speed can be formulated by;

$$N = \frac{1000V}{\pi d} \dots\dots\dots (7.4)$$

Substituting Equation 7.4 into Equation 7.3, and representing a constant K_1 by;

$$K_1 = \frac{\pi d K}{1000z} \dots\dots\dots (7.5)$$

the machining time equation can be represented as follows;

$$t_m = K_1 V^{-1} f^{-1} \dots\dots\dots (7.6)$$

Table 7.14. Nomenclature Used in the Formulation of Objective Function

Symbol	Defines	with Unit
A	chip cross-sectional area	mm ²
a, a _{rad}	axial depth of cut, radial depth of cut	mm
ac, al	clearance and lead angle of the cutting tool	degree
c _l , c _o	labour cost, overhead cost	\$/min
c _{mat} , c _t	cost of machining, raw material and a cutting tool	\$
C _u	unit cost	\$
d	diameter of the cutting tool	mm
e	machine tool efficiency factor	-
F	feed rate	mm/min
f	feed (per tooth)	mm/tooth
f _{opt}	optimum	mm/tooth
F _c , F _{c(per)}	cutting force, permitted cutting force	N
F _F , F _R , F _T	feed force, radial force, tangential force	N
G, g	slenderness ratio, exponent of slenderness ratio	-
K	distance to be travelled by tool for the operation	mm
K _p	power constant depending on workpiece material	-
N	spindle speed	rpm
n	tool life exponent	-
P, P _m	required power, motor power	kW
Q	contact proportion of cutting edge	-
R _a , R _{a(at)}	arithmetic and attainable value of surface finish	μm
T	tool life	min
t _m , t _s , t _{tc}	machining time, set-up time, tool changing time	min
T _u	unit time	min
V	cutting speed	m/min
V _{opt}	optimum cutting speed	m/min
w	exponent of chip cross-sectional area	-
W	tool wear factor	-
z	number of teeth (insert) on the cutting tool	

A corrected empirical formula for the practical tool life of a cutting tool to be used in milling operations has been proposed in Reference [176];

$$T = \frac{60}{Q} \left(\frac{C(G/5)^8}{A^w V} \right)^{\frac{1}{n}} \dots\dots\dots (7.7)$$

where the values of constant C and exponent n depend on the quality of the cutting tool and the hardness of the workpiece material. The value of n is 0.15 for HSS tools, 0.30 for the carbide tools. The parameter Q is the contact proportion of cutting edge with workpiece per revolution. Substituting the slenderness ratio ($G = a/f$) and chip cross-sectional are ($A = af$) into Equation 7.7, and representing a constant K_2 by;

$$K_2 = 60Q^{-1} C^{1/n} 5^{-g/n} a^{(g-w)/n} \dots\dots\dots (7.8)$$

the tool life equation becomes;

$$T = K_2 V^{-1/n} f^{-(w+g)/n} \dots\dots\dots (7.9)$$

By defining a constant $K_3 = K_1/K_2$, the ratio of machining time per tool life can be described by the following equation;

$$\frac{t_m}{T} = K_3 V^{(1/n)-1} f^{[(w+g)/n]-1} \dots\dots\dots (7.10)$$

Substituting Equation 7.7 and Equation 7.10 into Equation 7.3, gives the extended representation of objective function based on unit cost in terms of cutting speed (V) and feed rate (f);

$$C_u = c_{mat} + (c_1 + c_o)t_s + (c_1 + c_o)K_1 V^{-1} f^{-1} + (c_1 t_{tc} + c_t + c_o t_w)K_3 V^{(1/n)-1} f^{[(w+g)/n]-1} \dots (7.11)$$

However, the tool changing time can be used regardless of ratio of machining time to tool life, hence Equation 7.11 can be re-written in the form of;

$$C_u = c_{\text{mat}} + (c_1 + c_o)t_s + (c_1 + c_o)K_1 V^{-1} f^{-1} + c_t K_3 V^{(1/n)-1} f^{[(w+g)/n]-1} + (c_1 + c_o)t_{tc} \dots (7.12)$$

The first two and the last cost terms represent the material cost, set-up cost and tool changing cost, respectively. These parameters are not influenced by machining parameters and therefore can be excluded from Equation 7.12. The final optimisation model for the minimum production cost per operation can be represented by;

$$C_u = (c_1 + c_o)K_1 V^{-1} f^{-1} + c_t K_3 V^{(1/n)-1} f^{[(w+g)/n]-1} \dots (7.13)$$

The objective function for the *minimum production time criterion* can be derived in a similar way. The unit time (T_u) can be represented by the sum of three time terms; set-up time, machining time and tool changing time. Therefore, it can be represented simply by following equation;

$$T_u = t_s + K_1 V^{-1} f^{-1} + t_{tc} \dots (7.14)$$

Since set-up time and tool changing time are not influenced by machining parameters, the final optimisation model for the minimum production time per operation becomes;

$$T_u = K_1 V^{-1} f^{-1} \dots (7.15)$$

7.5.3. Constraints Used in the Optimisation

Following constraints are considered in the optimisation of cutting parameters;

- Machine power
- Surface finish
- Available feed rates on the machine tool
- Available spindle speeds on the machine tool
- Cutting force

Machine Power

The size of a cut (depth of cut) depends on the maximum available power on the machine tool to be used. Power required for a cutting operation should not exceed the effective power transmitted by the machine tool. However, optimal values of feed and speed values should be selected such that maximum machine power would be used. The power required for machining can be represented by [176];

$$P = \frac{0.78K_p Wza_{rad}a}{60\pi de} Vf^{0.8} \dots\dots\dots (7.16)$$

The power constraint can be written as the following inequality;

$$P \leq \frac{P_m}{e} \dots\dots\dots (7.17)$$

Surface Finish

The required surface finish value should be smaller than the maximum attainable surface finish under the existing machining conditions. The surface finish values for both *plain & end milling* and *face milling* can be represented by Equation 7.18 and Equation 7.19, respectively [176];

$$R_a = 318 \frac{f^2}{4d} \dots\dots\dots (7.18)$$

$$R_a = 318 \frac{f}{\tan(\alpha_l) + \tan(\alpha_c)} \dots\dots\dots (7.19)$$

The surface finish constraint can be described by the following inequality;

$$R_a \leq R_{a(attachable)} \dots\dots\dots (7.20)$$

Available Feed Rates

Optimal feed rate should be in the range determined by the minimum and maximum feed rates available on the machine tool. The feed rate constraint can be represented in the following form;

$$F_{\min} \leq F \leq F_{\max} \dots\dots\dots (7.21)$$

Note that the relation between the feed rate (F) and feed per tooth (f) can be given by;

$$F = fzN \dots\dots\dots (7.22)$$

Cutting tool manufacturers generally recommend optimum working ranges (in terms of feed/tooth) to be used with certain cutting inserts. These feed ranges usually change from 0.050 to 0.300 (mm/tooth). In this work, this range is also used as a constraint. However, it may not be used in the optimisation or the range can be changed by the user.

Available Spindle Speeds

Optimal spindle speed for an operation must be in the range determined by the minimum and maximum spindle speeds available on the machine tool. The spindle speed constraint can be characterised by;

$$N_{\min} \leq N \leq N_{\max} \dots\dots\dots (7.23)$$

Note that the relation between the cutting speed (V) and spindle speed (N) can be given by;

$$V = \frac{\pi d N}{1000} \dots\dots\dots (7.24)$$

Cutting Force

When milling a component, the total cutting force due to machining should not exceed the permitted cutting force that the cutting tool withstand. Permitted values of cutting forces have been introduced by tool manufacturers for different cutting tools. These values have been determined in actual machining conditions and represent maximum cutting forces under which cutting tools can be used safely. Therefore, when these values are respected, *some factors such as the tool's deflection, chatter and torsion are automatically taken into consideration*, since the results of the experiments have been affected by these factors [176]. The cutting force constraint can be characterised by following inequality;

$$F_c \leq F_{c(per)} \dots \dots \dots (7.25)$$

It should be noticed that there is a strong relation between the cutting force and power required to machine a volume with;

$$F_c = \frac{60000 * P}{V} \dots \dots \dots (7.26)$$

7.5.4. Optimisation Methodology

The optimisation problem discussed above involves four variables which are; the *number of passes* and the corresponding *depth of cut*, *cutting speed* and *feed rate* for each pass required to machine a component. The values of these four variables should be determined such that combination of them will minimise the objective functions given by Equation 7.13 and Equation 7.15.

The methodology proposed herein is twofold. In the first stage, tentative number of passes and depth of cut(s) to be removed are determined through a method called *volume sectioning*, while the cutting speed and feed rate for each pass are calculated by using a *GA* in the second stage.

The *volume sectioning* can be considered as a multi-stage decision process in which each of single-stage optimisation problem can be stated such that the volume to be cut is divided into possible sections (depth of cuts). The decision variable in *volume sectioning* is the *depth of cut* a_j to be taken in the i th pass, which is represented as $a(i,j)$. Total depth of cut a_t is divided into N equal sections. The minimum increment of depth of cut is therefore equal to: a_t / N . This increment should always be smaller than the maximum depth of cut and higher than the minimum depth of cut allowed for a machine tool workpiece system; $a_{\max}$ and $a_{\min}$, respectively. The *volume sectioning* procedure applied to multipass milling operations can be summarised in conjunction with a figure (Figure 7.13) as follows.

The total depth of cut to be machined (a_t), the minimum ($a_{\min}$) and maximum ($a_{\max}$) allowable depth of cuts and the number of sections (N_s) are the inputs to the volume sectioning procedure. Among them, the selection of a proper number of sections for the problem has the extreme importance, since higher precision, i.e., selecting a higher number, will increase the execution time dramatically, although more effective optimal values are calculated for the objective function. The value of N_s should be selected always by making a compromise between the execution time and the precision.

The thickness of each section (a_t / N_s) is called unit depth of cut (unit section). The problem is to find all the alternative possible passes that are composed of certain number of unit sections. Figure 7.13 shows a pass distribution in which total depth of cut is divided into 5 sections and $a_{\max}$ and $a_{\min}$ are equal to 4 sections and 1 sections, respectively. Here, thickness of each section can be considered as 1 mm.

The procedure is started from the 5th section. The stock can be machined from the outer end (right end) 5th section to the inner ends (right ends) of 4th, 3rd, 2nd sections, as shown in Figure 7.13. Notice that a cut (5,5) including 5 sections from the outer end 5th section to the inner ends of 1st section is not possible, since the maximum allowable depth of cut is 4 in terms of sections.

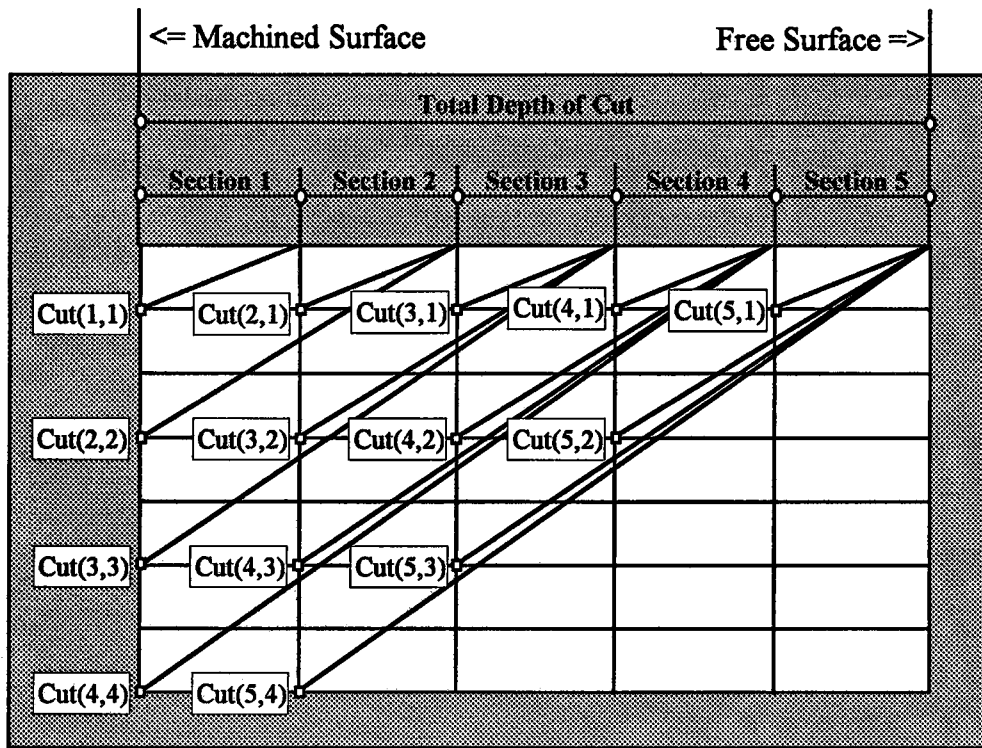


Figure 7.13. Schematic Representation of Sectioning Strategy ($N=5$, $a_{\max}=4$, $a_{\min}=1$)

The same procedure is continued successively until reaching to the first section and all possible cuts are stored for future use. According to Figure 7.13, to reach to first section from 5th section (to machine the total depth of cut), we have several alternative cutting strategies. These alternatives are;

- Cut(5,4) + Cut(1,1)
- Cut(5,3) + Cut(2,2)
- Cut(5,3) + Cut(2,1) + Cut(1,1)
- Cut(5,2) + Cut(3,3)
- Cut(5,2) + Cut(3,2) + Cut(1,1)
- Cut(5,2) + Cut(3,1) + Cut(2,2)
- Cut(5,2) + Cut(3,1) + Cut(2,1) + Cut(1,1)
- Cut(5,1) + Cut(4,4)
- Cut(5,1) + Cut(4,3) + Cut(1,1)
- Cut(5,1) + Cut(4,2) + Cut(2,2)
- Cut(5,1) + Cut(4,2) + Cut(2,1) + Cut(1,1)
- Cut(5,1) + Cut(4,1) + Cut(3,3)

- $\text{Cut}(5,1) + \text{Cut}(4,1) + \text{Cut}(3,2) + \text{Cut}(1,1)$
- $\text{Cut}(5,1) + \text{Cut}(4,1) + \text{Cut}(3,1) + \text{Cut}(2,2)$
- $\text{Cut}(5,1) + \text{Cut}(4,1) + \text{Cut}(3,1) + \text{Cut}(2,1) + \text{Cut}(1,1)$

In milling processes; for example, $\text{Cut}(1,1)$ and $\text{Cut}(2,1)$ are equal because workpiece is fixed while cutter turns as opposed to the turning. As a result, we have;

- $\text{Cut}(5,4) = \text{Cut}(4,4)$
- $\text{Cut}(5,3) = \text{Cut}(4,3) = \text{Cut}(3,3)$
- $\text{Cut}(5,2) = \text{Cut}(4,2) = \text{Cut}(3,2) = \text{Cut}(2,2)$
- $\text{Cut}(5,1) = \text{Cut}(4,1) = \text{Cut}(3,1) = \text{Cut}(2,1) = \text{Cut}(1,1)$

Therefore, the number of alternative cutting strategies are reduced to 6. These alternatives are given in Table 7.15.

Table 7.15. The Alternative Cutting Strategies

Cutting Strategy No	# Sections to be cut in each pass				
	Pass 1	Pass 2	Pass 3	Pass 4	Pass 5
1	1	4	-	-	-
2	1	1	3	-	-
3	1	2	2	-	-
4	1	1	1	2	-
5	1	1	1	1	1
6	2	3	-	-	-

If $a_{\max}$ was given equal to the total depth of cut (5 sections), a single pass strategy ($\text{Cut}(5,5)$) would also be considered as an alternative solution with the ones given in Table 7.15.

After the cutting strategy alternatives are determined by the *volume sectioning procedure* discussed above, the developed *GA* finds the optimal values of feed-rate and cutting speed which minimise the objective function for each pass. When they are found for each pass, the cutting strategy (i.e. strategies given in Table 7.15) that leads to minimal objective function value for multi-pass operation is selected as the optimal strategy. The number of passes, depth of cut for each pass, feed-rate and cutting speed

values associated with the optimal strategy are the optimal cutting parameters to be used in machining.

The methodology used in the optimisation of the feed-rate and cutting speed with a GA is mainly similar to those used in the optimisation of sequence of operations and turret indexing time through genetic algorithms. However, in this case Equation 7.13 and Equation 7.15 are used as the objective functions (fitness function). Furthermore, the chromosomes used here are *binary* as opposed to those used in the turret indexing and operation sequencing cases. Here, we have two (feed and cutting speed) parameters to be controlled simultaneously within their specified ranges. The ranges of two parameters are also different from each other, hence normalisation of chromosomes are necessary. A typical chromosome of feed and cutting speed is involves two sections and are given in Table 7.16.

Table 7.16. A Typical Chromosome for Cutting Parameters Optimisation

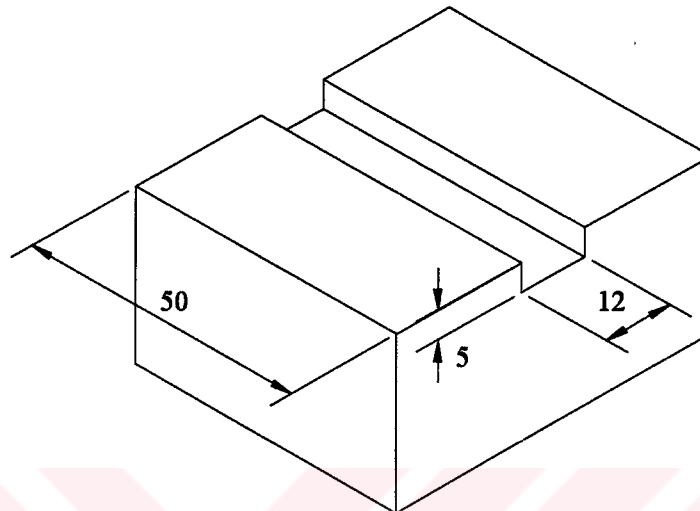
Positions	1	2	3	4	5	6	7	8	9	10	11	1	2	3	4	5	6	7	8	9	10	11
Binary Rep.	¹⁰ 2	⁹ 2	⁸ 2	⁷ 2	⁶ 2	⁵ 2	⁴ 2	³ 2	² 2	¹ 2	⁰ 2	¹⁰ 2	⁹ 2	⁸ 2	⁷ 2	⁶ 2	⁵ 2	⁴ 2	³ 2	² 2	¹ 2	⁰ 2
Chromosome	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Value	1024											1										
Divider	1000											10										
Normalised Value	1.024 mm/tooth											0.1 m/min										
	FEED SECTION											CUTTING SPEED SECTION										

New chromosomes of *feed and cutting speed* are generated from the initial population by using the genetic operators discussed in the previous sections. Fitness values of the chromosomes are then calculated and based on the fitness values, the next generation is formed from the newly generated chromosomes of feed and cutting speed and old population. As the iterations of the genetic algorithm continue, the better cutting parameters (chromosomes) that minimise the objective function based on minimum production criterion dominate and the GA converges to an optimal set of cutting parameters.

7.5.5. Case Study

A slot as shown in Figure 7.14 is to be machined on a CNC VMC. It is required to find optimal machining parameters feed and speed based on both the

minimum production cost and minimum production time criteria and using the constraints discussed above. Specifications of the machine tool, workpiece material and values of the constants are given in Table 7.17. The other information on the required machining operations and cutting tools are provided in Table 7.18 and Table 7.19, respectively.



All dimensions are in mm.

Figure 7.14. A Slot

Table 7.17. Specifications and Constants

Specification	
C_{mat}	0.25 \$
C_o	1.45 \$/min
C_l	0.45 \$/min
t_s	2 min
t_{tc}	0.5
C (for HSS tools)	33.98
C (for carbide tools)	100.05
K_p	2.24
W	1.1
n (for HSS tools)	0.15
n (for carbide tools)	0.3
g	0.14
w	0.28
Machine Tool	VMC
Machine Power	5.5 kW
e	0.95
Material Type	10L50 Leaded Steel
Material Hardness	225 BHN
N_s	5 mm
a_{max}	5 mm
a_{min}	1 mm

Table 7.18. Machining Operation and Its Requirements

Opr. No.	Opr. Type	Tool No	a (mm)	K (mm)	Ra (μm)	Face for surface roughness
1	Slot milling	1	5	62	-	-

Table 7.19. Tooling Data

Tool No	Tool Type	Quality	Ssy (MPa)	d (mm)	Cutting Length (mm)	z	Price (\$)	Shank Diameter (mm)	Helix Angle ($^{\circ}$)	al ($^{\circ}$)	ac ($^{\circ}$)
1	End-mill	HSS	1000	12	40	4	10	10	45	0	5

Table 7.20. Optimum Machining Parameters Found by the Optimisation Program

Depth of cut (mm)	Catalogue Values		Minimum Production Cost		Minimum Production Time	
	V(m/min)	f (mm/tooth)	V (m/min)	f (mm/tooth)	V (m/min)	f (mm/tooth)
1	25.0	0.100	51.2	0.224	96.0	0.272
2	25.0	0.100	51.5	0.248	38.4	0.241
3	20.0	0.100	38.0	0.254	15.2	0.252
4	20.0	0.100	25.6	0.254	11.2	0.256
5	20.0	0.100	25.6	0.255	25.3	0.142

when; feed range: 0.050-0.300 mm/tooth and # of iterations in the GA: 50

Table 7.21. Objective Function Values

Depth of cut (mm)	Catalogue		Cu (\$)	Tu (min)
	Cu (\$)	Tu (min)		
1	0.4441	0.2337	0.0990	0.0279
2	0.4413	0.2337	0.0922	0.0632
3	0.5551	0.2921	0.1165	0.1526
4	0.5551	0.2921	0.1709	0.2038
5	0.5551	0.2921	0.1703	0.1627

Table 7.22. Cutting Strategies for Machining with Their Ratings

Cutting Strategy (Pass Distribution) (mm)	(Catalogue) Tu (min)	(Catalogue) Cu (\$)	Tu (min)	Cu (\$)
1-4	0.5258	0.9992	0.2317	0.2631
1-1-3	0.7595	1.4433	0.2084	0.3009
1-2-2	0.7011	1.3267	0.1543	0.2834
1-1-1-2	0.9348	1.7736	0.1469	0.3892
1-1-1-1-1	1.1685	2.2205	0.1395	0.4950
2-3	0.5258	0.9964	0.2158	0.2087
5 (single pass)	0.2921	0.5551	0.1627	0.1703

The optimum cutting parameters found by the proposed volume sectioning and GA methodology as well as the catalogue values are tabulated in Table 7.20. The associated objective function values are given in Table 7.21. The possible cutting strategies with their total objective-function results are given in Table 7.22. As can be seen from Table 7.21 and Table 7.22 (in which the objective function values have been determined based on optimum machining parameters in comparison with those from handbook recommendations) considerable cost or time savings have been achieved with the optimum machining parameters in all cases. For the minimum production time, the best cutting strategy is found to be (1-1-1-1-1) whereas for the minimum production cost, a single pass strategy (5mm) results in the better improvement. It should also be noted that these improvements can further be increased by using higher number of iterations in GA cycle and loosening the feed range. For example, when allowable feed range is relaxed to (0.050-0.600 mm/tooth) and 250 iterations are used, the more effective values of feed and cutting speed can be found. However, in this case the processing time is increased considerably. The values of machining parameters and their ratings for just a single depth of cut (5mm) are shown in Table 7.23 and Table 7.24, respectively.

Table 7.23. Optimum Machining Parameters for 5mm Depth (# Iterations=250)

Depth of cut (mm)	Catalogue Values		Minimum Production Cost		Minimum Production Time	
	V(m/min)	f (mm/tooth)	V (m/min)	f (mm/tooth)	V (m/min)	f (mm/tooth)
5	20	0.10	24.8	0.512	25.8	0.486
when; feed range: 0.050-0.600 and # of iterations in the GA: 250						

Table 7.24. Ratings of Parameters Shown in Table 7.23.

Cutting Strategy (Pass Distribution) (mm)	(Catalogue) Tu (min)	(Catalogue) Cu (\$)	Tu (min)	Cu (\$)
5 (single pass)	0.2921	0.5551	0.0466	0.0882

CHAPTER VIII

CAPP/CAM INTERFACE: CNC CODE GENERATION

8.1. INTRODUCTION

Computer Aided Manufacturing (CAM) and its elements with implementation techniques are briefly presented in Section 8.2. Section 8.3 discusses the specifications of TAKSAN TMC 700 VMC (Fanuc) used for the applications of this study. The methodology used in the development of *CNC code generation module* is introduced in Section 8.4. An extension of the CNC code generation module, namely, *CNC code generation for arbitrarily shaped pockets with islands*, is given in Section 8.5.

8.2. COMPUTER AIDED MANUFACTURING (CAM)

Computer Aided Manufacturing (CAM) is the term used commonly for all the computer-controlled activities that involve a process of converting raw material into finished products. CAM-I (Computer Aided Manufacturing - International, a non-profit firm based in USA) defines the term CAM as follows;

“the effective utilisation of computer technology in the management, control, and operations of the manufacturing facility through either direct or indirect computer interface with the physical and human resources of the company”.

The most important elements of CAM are;

- CNC manufacturing techniques
- Robotics
- Flexible Manufacturing Systems (FMS)
- Computer Aided Inspection (CAI)
- Computer Aided Testing (CAT)

8.2.1. Generation and Verification of Part Programs

In *Computerised Numerical Control (CNC)*, part program is the only key providing the interrelation between the man and machine. When the instructions in terms of a part program are supplied to the machine tool, its controller unit reads and interprets the set of commands, and executes them. That is, some actions had been already taught to the machine, and they have been kept in the memory. These pre-taught functions (ISO codes; G and M functions, etc.) determine the machining capability of the machine tool. Although there are many standard functions, it is not a necessity for a machine to have all these functions. For example, a machining center may have all G words, while another may be capable of performing only a few function. However, it is clear that the higher the number of those pre-taught functions, the more intelligent the CNC machine is [6]. The detailed lists of G and M codes provided by TAKSAN TMC 700 (Fanuc) VMC are given in *Appendix E*.

The automatic generation of the tool path and CNC part programs is one of the major issues in the CAPP applications. It also plays a key role being an interface between the CAPP and CAM. On the basis of the geometric data developed for a designed component to be machined and the process plan(s) prepared internally, a typical automatic CNC code generation module is capable of generating the cutter path and CNC part program for a specified machine tool (machine controller unit).

The animation of the machining processes and graphical display of the cutter path are two modes of simulation modules in CAPP applications. The tool path simulation is a helpful tool permitting the user to visualise the motion of the cutting

tool. The cutting tools are not shown in the tool path presentation. Therefore, the programmer can not see possible tool collisions. During the tool path display, rapid traverse movements can be represented by dotted lines, while the cutting motions by solid lines. In the case of colour screen, the colour of the path is changed between tool changes.

The machining animation is useful for debugging, understanding, and finding errors in part programs in the shop-floor without interrupting the production cycle. There has been a dramatic increase in the use of “*animation*” for manufacturing analysis in the past few years. Most of the controller units now offer a version of their software with animation capabilities which increases the credibility of the model. The machining animation allows the user to simulate exactly what will happen and therefore to check almost everything including the tool collisions when the program is executed. The programmer will be able to check the tool collisions.

8.3. MACHINE TOOL SPECIFICATIONS

TAKSAN TMC 700 is a CNC VMC for flexible batch production of components. It has a 5.5 kW spindle motor, a 1 kW feed motor for X and Y axes, and 2.1 kW feed motor for Z axis. Spindle speeds are programmable and variable from 60 to 6000 rpm. The rapid traverse feed rate is 20 m/min for all the axes. Feed rates for cutting can be programmed from 1 to 10000 mm/min. The X axis and Z axis travels are limited to 700 mm and 450 mm, respectively. It has a 16-station ATC. Tool change time from tool-to-tool is specified as 5 seconds. A detailed information on TAKSAN TMC-700 can be found in Reference [173].

TAKSAN TMC-700 VMC is fitted with the standard Fanuc Series O-MC machine tool control. The most of standard G and M words and canned cycles like rough milling, pocket milling etc., are implemented on the system. Part programs can be loaded to the machine through RS-232 serial interface as well as from MDI interface. A detailed information on the specifications of the machine tool controller, i.e., *from programming format to canned cycles*, can be found in Reference [177].

8.4. CNC CODE GENERATION MODULE

CNC code generation module generates part programs to be used on the machine tool for machining components by processing CAD/CAM data base prepared and optimised in the upstream process planning modules of OPPS-PRI. An example component and its accompanying data base are given in Figure 8.1 and Table 8.1, respectively. Notice that unless specified, all parameters are in mm in the Figure 8.1 and Table 8.1.

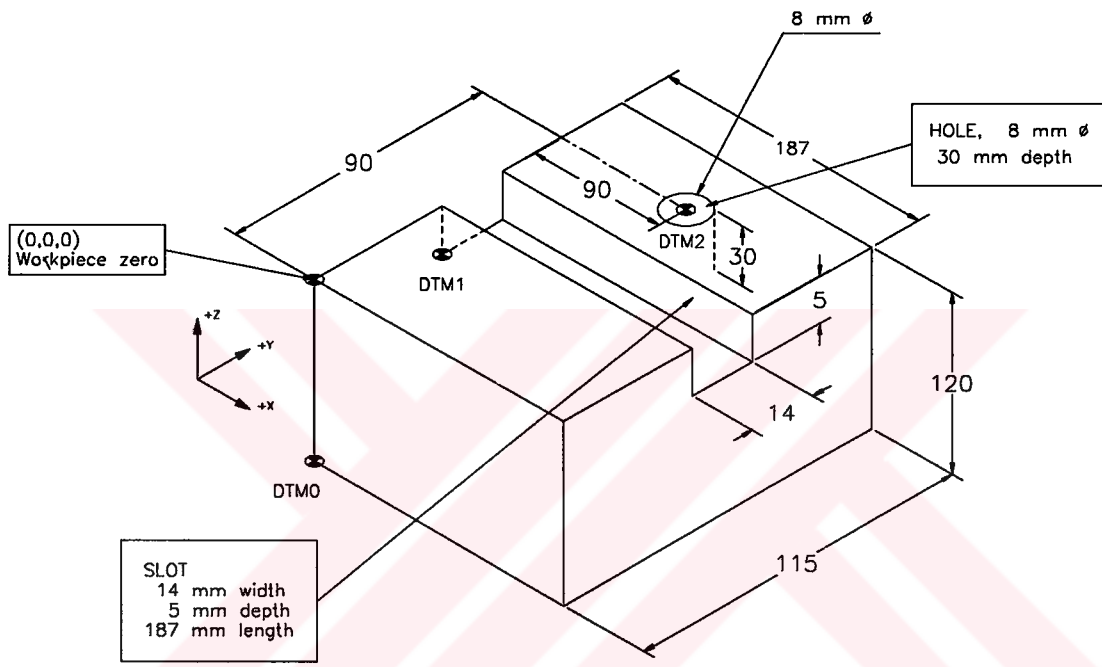


Figure 8.1. An Example Component

Table 8.1. CAD/CAM Data Base for the Example Component given in Fig. 8.1.

F. No	Feature ID	Distance to main datum			Feature Attributes			Approach Directions						SR/FR (μ m) / (mm)
		d-x	d-y	d-z	d	w	l	1	2	3	4	5	6	
1	T-SLT	0	47	115	5	14	187	1	0	0	0	0	0	-
2	B-HOL	90	90	120	-	8	30	1	0	0	0	0	0	-
Op. No	Operation Type	Workpiece Zero Shift			Associated Feature	Tooling								
		wsx	wsy	wsz		No	TN	ID	HOLDER	INDEX				
1	Slot Milling	0	0	120	1	1	EM14	W14	1					
2	Twist Drilling	Workpiece: ST-52			2	2	TD8	KMA1	2					
Pass	Operation	Depth of cut			Feed (mm/min)			Speed (rpm)						
1	1	5			400			1200						
-	2	-			60			600						
Tool	Diameter	Shank Diameter	Cutting Length	# of teeth	Material		Firm	Code						
EM14	14	12	26	4	HSS		MT	-						
TD8	8	8	10	-	HSS		MT	-						

The work done in the generation module is to prepare some machining instructions by considering the rules of MCU (Machine Controller Unit) and the geometry of the component. For each of the milling operations assigned to part features conceived by the feature recognition module, a *macro that is capable of proposing CNC codes to perform the specified milling process when called*, is prepared. As an example, the macro prepared for slot milling is documented by a pseudo language is as follows;

```

if <tool is at the Tool Reference Point>
position the tool for the slot to be cut (generate codes for cutter movements)
cut the slot with the specified feed value (generate codes for cutting)

if <tool is not at the Tool Reference Point>
retract the tool to a safe position (generate codes for cutter movements)
position the tool for the slot to be cut (generate codes for cutter movements)
cut the slot with the specified feed value (generate codes for cutting)

if <next operation uses the same tool>
retract the tool to a safe position (generate codes for cutter movements)
identify this position as the starting point for the next operation

if <next operation don't use the same tool>
retract the tool to the Tool Reference Point (generate codes for cutter movements)

```

Each macro has basically two levels for determination of the machining instructions. In the first level, the set of instructions are determined by the location of the cutting tool. If any cutter is at *Tool Reference Point (TRP)*, it is simply positioned for cutting after some functions and then machining is started with the associated G word. If the cutter is not at the *TRP*, the cutter is first retracted to a safe position for preventing any tool collision with the part. After the cutter is brought to its new location the next program blocks for positioning and cutting are appended to the program code file that is opened for the component.

In the second level, the *tooling information* plays a critical role in the determination of the machining commands. If the subsequent operation requires the same cutter, the current location is identified as the new reference point of the tool. No machining instructions are given before exiting the macro. If the next cutter is different; the cutter is first intended to be retracted to the tool changing position with rapid traverse command (G00) and the spindle is stopped by instating the necessary word (M05). The tool is then changed as requested in the CAD/CAM data base of the component.

The process of part program generation is performed by scanning the data base starting from the first operation and adding the CNC statements proposed in the macros prepared for the machining operations into a part program file. The process is terminated after all operations in the data base are scanned. CNC codes that can be used on TAKSAN TMC 700 VMC (Fanuc) generated for the example component are tabulated in Table 8.2. Note that H1 and H2 specified in the program are related to tool length offsets and they should be measured and input to the machine tool control, before actual machining takes place.

Table 8.2. The Part Program Generated for the Example Part

O1111									
N10	G91	G28	X0	Y0	Z0;				
N20	M3	S1200;							
N30	G0	G90	G43	G54	X-10	Y54	Z10	H1;	
N40	G0	Z-5;							
N50	G1	X197;	F400;						
N60	G0	Z50;							
N70	G91	G28	X0	Y0	Z0;				
N80	M5;								
N90	M6	T2;							
N100	M3	S600;							
N110	G0	G90	G43	G54	X90	Y90	Z5	H2;	
N120	G1	Z-30	F60;						
N130	G0	Z10;							
N140	G91	G28	X0	Y0	Z0;				
N150	M5;								
N160	M2;								

8.4.2. Simulation Tool Used for Verification of Part Programs

It is obvious that the *simulation* is a useful tool which increases productivity. It is a well known fact that the possibility of damage to machine tool and the amount of scrapped material can be reduced dramatically by the use of simulation method prior to actual machining. In this study, the generated part programs are verified by using the *CNCEZ - TORCAM* simulation software [178] that can simulate a virtual CNC machine tool with the Fanuc controller unit. The reason for this is the intention of eliminating the duplicate works, and hence, saving a considerable time and reflecting this gain to the other application areas in the CAD/CAM integration.

8.5. CNC CODE GENERATOR/SIMULATOR FOR CAVITY MACHINING

Cavities including *arbitrarily shaped pockets* are some of the most common features found on milling components. The level of automation in CNC code generators provided by commercial packages is somewhat low, especially in terms of specification of cutting tool to be used in machining. Therefore, fully automatic generation of error free CNC part programs for cavity machining has a great importance for CAD/CAM industry. In this essence, development of an *automatic CNC code generator for arbitrarily shaped cavities* implemented in an integrated environment has been a necessity.

This part of the work is an attempt for the development of a software for cavity machining. With this software, part programs to machine reasonably complex pockets could safely be generated and verified. Another motivation for the development of this submodule has been the fact that cutter path generation with the true selection of cutting tool could potentially form the bottleneck in the cavity machining. A number of cutter path generation systems are in use today, but most of them are application-specific and would fail if faced with new parts or new situations such as the cutter size being too big to machine some areas of the part.

The developed program for *CNC code generation for arbitrarily shaped pockets with islands*, calculates the minimum distance between islands, between islands and pocket contours, etc. The minimum distance found internally is then used for the selection of suitable tool size. The tool path generation is performed by using offsetting vertices of islands and borders of the pockets by the use of trigonometry such that the cutter can remove all material within the pocket without exposing the constraints by the pocket and island edges. Tool path is stored in the output data file and then be converted into CNC part program based on a specific machine tool controller (i.e. *Fanuc*). Main contribution achieved here is to combine “*contouring*” and “*staircase*” methods used for the generation of the tool path for pockets which have their own merits and demerits. Resulting part programs can be verified by animating chip removal processes on the graphics screen.

8.5.1. Background on Pocket Machining

The state-of-the-art about CNC code generation for pockets is *thoroughly* discussed in Reference [179]. The literature survey on the subject, types of tool path generation methodologies proposed for pocket machining with their advantages and disadvantages, and the current status of the *cavity machining* can be found in the same reference. Some of the definitions associated with pocket machining technology are also given below. There are basically two methods proposed in the literature for pocket machining; *contouring* and *stair-casing*, as shown in Figure 8.2.

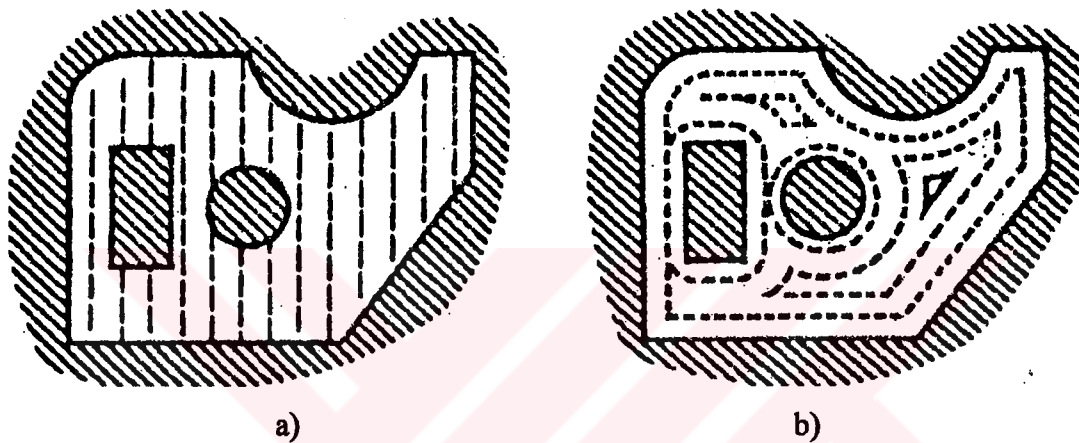


Figure 8.2. Pocket Machining Methods: a) Stair-casing b) Contouring [180]

Both methods have their own advantages and disadvantages. For example, although it is easy to apply stair-casing method for machining pockets with islands as depicted in Figure 8.2.a, the obtained surface quality is low. Moreover, the rapid traversing time spend in the machining is considerably high. The second one; contouring as depicted in Figure 8.2.b, produces good surfaces. However, the programming of this second method is quite difficult as compared to stair-casing, since it requires a large amount of tool compensation and offsetting. Moreover, it is a possibility that uncut islands can be left inside the pocket which are undesirable and requires re-machining (hence re-programming).

8.5.2. Methodology of Pocket Machining System

It should be noticed that the most of the CNC machine tool controls offer *canned cycles* for making pocket machining easier. However, the pockets that can be machined with the use of these canned cycles have a simple section like rectangle or circle, and have not got any islands. Therefore, the system developed for *CNC code generation for arbitrarily shaped pockets with islands* in this study fills this important gap. The developed system is actually composed of *four submodule* whose functions are as follows;

- tool selection
- tool path generation
- graphical simulation
- part program generation

Tool Selection for Pockets with Islands

In this study, machining of *arbitrarily shaped pockets with islands* by a single cutting tool is considered. So no tool change is employed in machining of these pockets, although more than one cutter may also be used for machining a pocket.

The number and the type of cutter(s) to be employed for machining a pocket depends on many conflicting parameters and constraints. For example, if only a single cutter (*whose diameter is equal to the minimum distance imposed by boundaries of the pocket*) is used to cut a large-volume pocket including small throats between its features, machining time is considerably increased, although tool changing time is saved. However, if more than one single cutter (*whose diameters are reasonably large*) are used for the same pocket, in this case, machining time is decreased since chip removal rate is increased by the large cutters. Notice that, the tool changing time should also be considered. It is worth to notice that the second method has another disadvantage such that the radial depth of cut is continuously changed due to unmachined regions left by other cutters in the pocket. This may cause some impact problems and also difficulties in tool path calculation; so in programming.

There are mainly three factors that affect the tool selection for *arbitrarily shaped pockets with islands*. These are; geometrical constraints imposed in the pocket, fillet radii of the pocket, available cutters.

In the first stage of tool selection module, the orientations due to pocket entities or the regions of the pocket which cannot be machined with the available tools are searched. If there exist such characteristics in the pocket as shown in Figure 8.3, the user is warned to modify the pocket geometry.

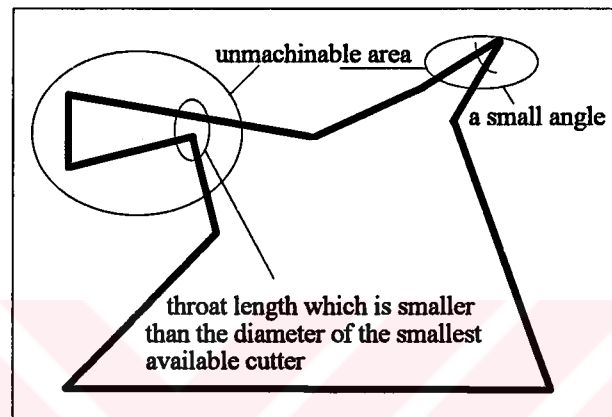


Figure 8.3. Geometrical Constraint (Throat) and Unmachinable Regions

The minimum distance between the pocket boundaries and islands' boundaries, or between the islands' boundaries is then calculated. This minimum distance value is used to determine the diameter of the candidate cutter to be used for machining the pocket. The diameter of the cutter should be equal or less than this minimum distance value. A suitable cutter is then selected for the pocket. Since the data base of a pocket includes segments of arc and straight lines, analytical geometry basics are used in the calculation of the minimum distance, by controlling the following;

- distance between a point and a straight line
- distance between two arc segments
- distance between a point and an arc segment
- distance between a straight line and an arc segment

The equation that can be used for calculating the minimum distance of a point to an arc segment, is shown in Figure 8.4.

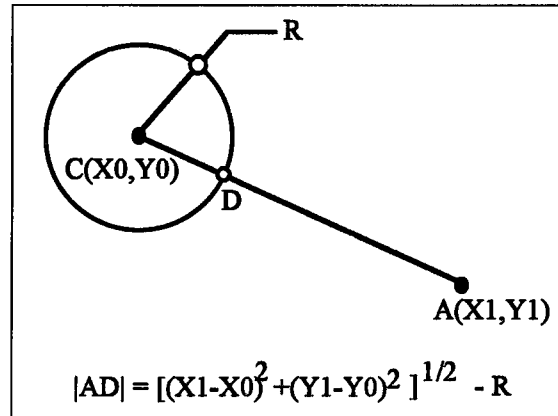


Figure 8.4. Calculation of the Distance of a Point to an Arc Segment

Tool Path Generation for Pockets with Islands

In this study, a hybrid methodology is developed by compensating the disadvantages of both stair-casing and contouring methods. In this hybrid method, *grooves* are first machined around the internal pocket boundary and the external islands' boundaries using the *contouring* scheme. By this way, the surface quality problems are eliminated and a safe working area that can easily be machined with *stair-casing* scheme is left in the pocket. In order to eliminate the second disadvantage of the stair-casing method, the tool path segments are grouped according to pocket geometry and a bi-directional cutting scheme is used when generating the part program based on the tool paths.

It is necessary to make offsetting processes in calculating the coordinates to be traced by the cutter while generating the tool paths, especially while using contouring scheme. These offsets depends on the types of entities like arc and line etc., constituting the boundaries of the pocket or islands in the pocket. They are automatically calculated by the program. As an example, the calculation of the positioning offsets ΔX and ΔY for a convex corner which is made up of a line and an arc is illustrated in Figure 8.5. A detailed information on these offsets can be found in Reference [181].

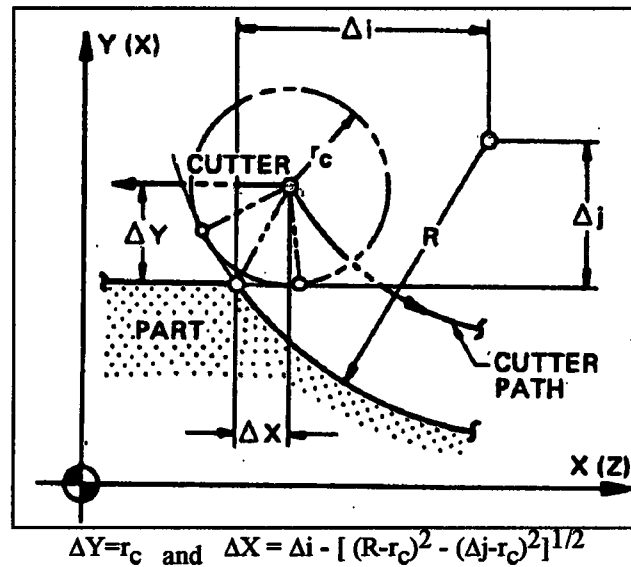


Figure 8.5. Positioning of a Cutter [181]

Tool Path Simulation for Pockets

A graphical simulation program is prepared to control the calculated tool paths. Tool paths which are considered to be a rough description of a part program, are shown on the screen within the pocket. Machining animation (after the CNC codes are generated) can also be performed in order to see whether there is an unmachined area left in the pocket or not.

In the tool path simulation, the cutter location files which are generated in the tool path generation module and stored in the hard-disk are retrieved and the paths to be traced by the cutter are drawn within the pocket on the screen. The colours of the paths for different movements are changed for easy visualisation.

In the pocket machining animation, the cutting processes, rapid traverse movements are animated on the screen in accordance with the generated CNC codes. To animate a chip removal process; the area of the material being machined is painted in the background colour of screen, while the cutting tool is repeatedly redrawn along a tool path according to the related CNC codes. In the simulation of rapid traversing, only the cutting tool is repetitively redrawn on the screen between two coordinates.

CNC Code Generation for Pockets with An Example

A post-processor is developed to interpret the developed tool paths based on the controller unit of a selected CNC machine tool, hence to obtain the CNC codes to be used for machining the arbitrarily shaped pockets with islands. In this study, the CNC codes are generated based on the controller unit (Fanuc) of TAKSAN TMC - 700 VMC.

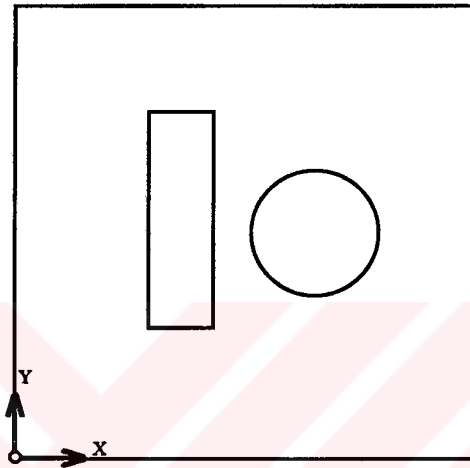


Figure 8.6. An Example Pocket with Two Islands

Table 8.3. Pocket Boundary - Data Base for The Pocket Shown in Fig. 8.6.

Segment	Type	x1	y1	x2	y2	r
C	L	0	0	200	0	0
C	L	200	0	200	200	0
C	L	200	200	0	200	0
C	L	0	200	0	0	0

Table 8.4. Island Boundary - Data Base for The Pocket Shown in Fig. 8.6.

No	Segment	Type	x1	y1	x2	y2	r
1	I	L	60	60	60	150	0
1	I	L	60	150	80	150	0
1	I	L	80	150	80	60	0
1	I	L	80	60	60	60	0
2	I	A	110	100	0	360	20

T.C. YÜKSEKÖĞRETİM KURULU
BİLİMİN VE TEKNOLOJİNİN
KURUMSALLAŞTIRILMASI
KURUMU

A step-by-step explanation of the CNC code generation module prepared for pockets is given through an illustrative example below. An example part and its data bases are given in Figure 8.6, Table 8.3 and Table 8.4.

Table 8.3 is for pocket boundaries (contours) and includes type of segments and geometry definitions (x1, y1, x2, y2, r). The parameter L defines a line while I denotes an island. The parameters x1, y1 and x2, y2 in the geometry definitions describe the starting and finishing coordinates of line entities, respectively. For arcs, these stand for the centre coordinates in x and y axes, start angle, end angle, respectively. The next column r is used to define the radius of an arc or a circle. The structure of Table 8.4 is similar to that of Table 8.3. However, the islands are given identification numbers for processing, since a pocket may have more than one island.

When the program is executed, the pocket shown in Figure 8.6 is drawn on the monitor by using its data bases given in Table 8.3 and Table 8.4. The minimum distance between the entities like arcs or lines etc., (also between the elements of these entities like a point) is then calculated by using these data bases and taking all possibilities into account. For the part given as example, the minimum distance is found between two islands as *10 mm* as shown in Figure 8.7.

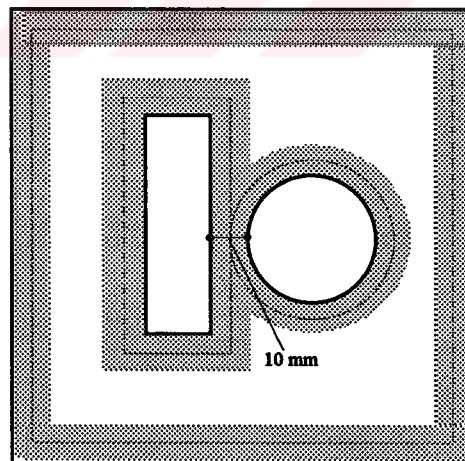


Figure 8.7. Minimum Distance Between Two Islands and Cutting Around Boundaries

The cutter that has the nearest and a smaller value of diameter to the minimum distance value (10 mm) is selected as the tool to be used for machining the example

pocket with island, from the tooling library prepared by using the manuals of different tool manufacturers. If such a cutting tool does not exist in the tool library, the user is given a warning such that this pocket cannot be machined with the available cutters and under these constraints.

The next stage is the generation of the tool paths for pre-machining of pocket and islands' boundaries by using the selected cutter with the *contouring programming* scheme as shown in Figure 8.7. All offsets and compensations of the cutter are automatically calculated by the program and the coordinates that are to be traced with the selected cutter are stored in the tool path file opened for the example part.

The tool paths for machining the regions left after contouring, are then determined by using the *stair-casing programming* scheme as shown in Figure 8.8. The tool path segments are initially determined in a sequential order (1,6,2,5,3,4) as shown in Figure 8.8. These segments are then grouped and ordered in order to reduce the rapid traversing and the coordinates that are to be traced with the selected cutter are appended to the tool path file previously opened for the example part. Finally, the required tool path for finishing operations are appended to this file.

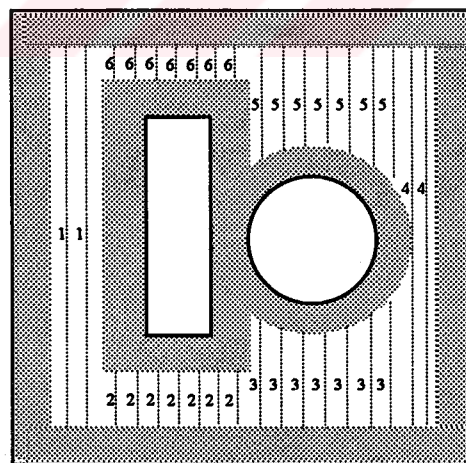


Figure 8.8. Segments of Tool Paths

The tool path (for only rough machining) generated for the example part is given in Table 8.5. This tool path can be interpreted by a post-processor to generate the part program based on a machine tool controller.

Table 8.5. Codes Generated for the Example Pocket shown in Fig. 8.6 (rough cutting)

#	G	X	Y	Z
1	00	5	5	
2	01			-2
3	01	195	5	
4	01	195	195	
5	01	5	195	
6	00			2
7	00	55	55	
8	01			-2
9	01		155	
10	01	85		
11	01		55	
12	01	55		
13	00			2
14	00	110	75	
15	01			-2
16	02	85	100	
17	02	110	125	
18	02	135	100	
19	02	85	100	
20	00			2
21	00	15	10	
22	01			-2
23	01		190	
24	01	25		
25	01		10	
26	01	35		
27	01		190	
28	01	45		
29	01		10	
30	01	55		
31	01		50	
32	01	65		
33	01		10	
34	01	75		
35	01		50	
36	01	85		
37	01		10	
38	01	95		
39	01		72.5	
40	01	105	70	
41	01		10	
42	01	115		
43	01		70	
44	01	125	72.5	
45	01		10	
46	01	135	0	
47	01		81	
48	00			2
49	00	185	10	
50	01			-2
51	01		190	
52	01	175		
53	01		10	
54	01	165		
55	01		190	
56	01	155		
57	01		10	
58	01	145		
59	01		190	
60	01	135		

continuing from the previous page.				
61	01		119	
62	01	125	127.5	
63	01		190	
64	01	115		
65	01		130	
66	01	105		
67	01		190	
68	01	95		
69	01		127.5	
70	01	85	160	
71	01		190	
72	01	75		
73	01		160	
74	01	65		
75	01		190	
76	01	55		
77	01		160	
78	00			2
79	00	0	0	



CHAPTER IX

APPLICATIONS

9.1. INTRODUCTION

The use and execution of the modules of OPPS-PRI developed in this thesis are discussed with practical examples and demonstrations in this chapter. The simplified flowchart of the OPPS-PRI and interactions of the system modules with each other are also presented.

9.2. OPPS-PRI: An Optimised Process Planning System for Prismatic Parts

OPPS-PRI is a process planning system implemented on a PC which aims in the first place to integrate CAD and CAM with its interfaces taking *optimality* in each stage of planning endeavour into account. Artificial intelligence techniques are generally utilised for optimising the process planning functions. It is originally developed for prismatic parts and implemented for a VMC available in the Mechanical Engineering Department at University of Gaziantep. A “*DFM (Design for Manufacturing)*” utility is also provided within the system. The developed package is composed of sixteen modules which can be clustered into five main sections. The constituents of each sections were given in previous sections. For each of the modules forming the sections of whole system, a computer program was prepared, except the ones for the verification and down-loading of CNC codes for which commercial

packages are adopted. The programs were written in C language. The modules are combined under a supervisory program developed using Visual Basic language. An overall structure of the OPPS-PRI is given in Figure 9.1.

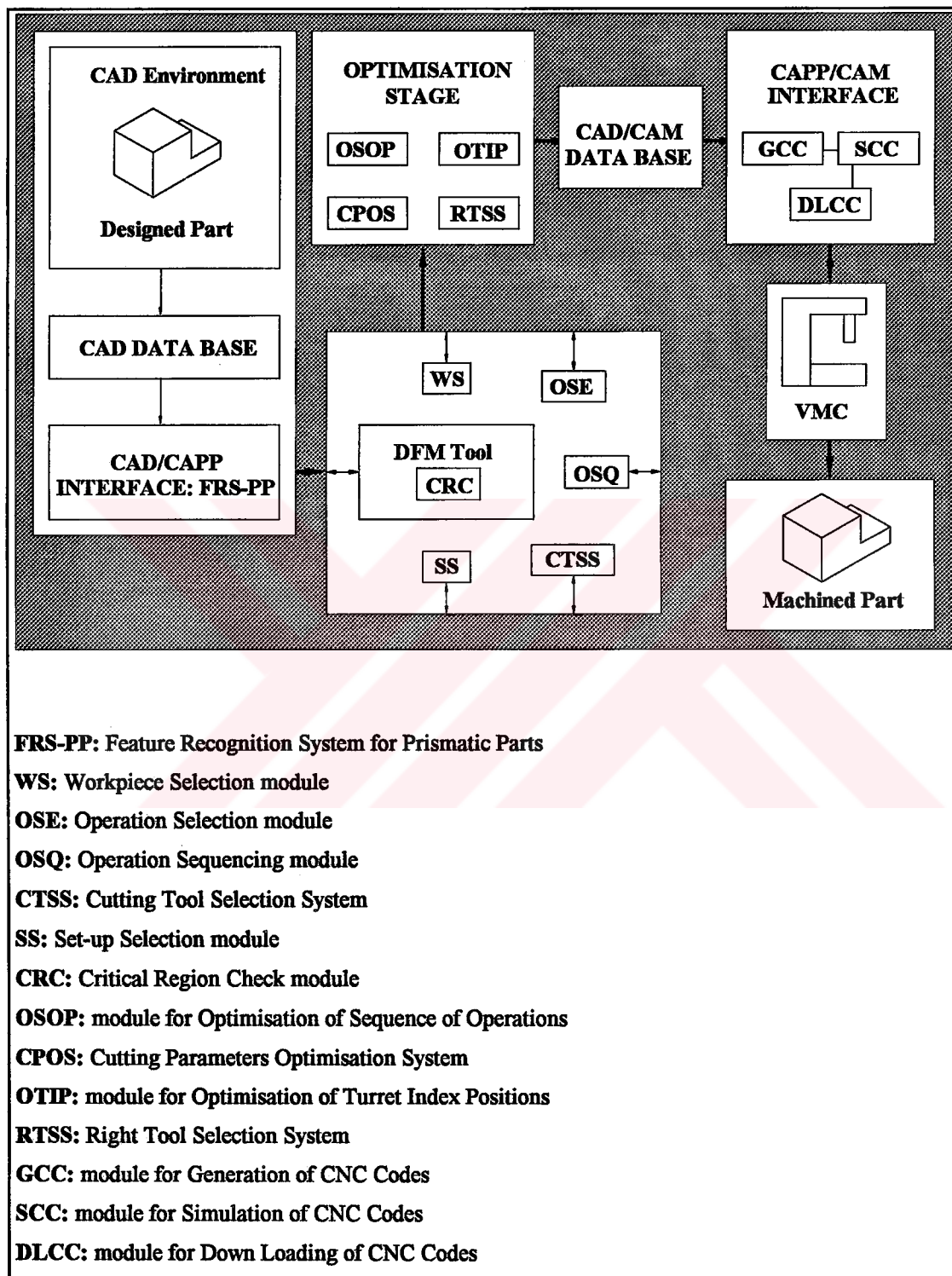


Figure 9.1. The Overall Structure of the OPPS-PRI

9.3. EXECUTION OF THE OPPS-PRI

OPPS-PRI which can automatically be installed onto hard-disk, can be executed by typing `opps-pri` followed by Enter key or running `opps-pri.exe` in the *Windows* environment. When the program is executed, an opening screen including a user-friendly pop-up/pull down menu is immediately displayed on the monitor as shown in Figure 9.2. The opening screen of OPPS-PRI has seven menus, under the headings of, FILE, CAD-INTERFACE, PLAN, OPTIMISATION, DFM-TOOL, CAM-INTERFACE, and ABOUT. They can be selected by the use of mouse when the mouse pointer is moved on the menu headings (highlighting) or entering the underlined characters specified for each menu heading from the keyboard. The system can then be easily driven by pulling-down menu items from the corresponding menus or by selecting suitable keys from the keyboard and giving the necessary inputs as requested. Pull-down menus of OPPS-PRI together with their contents (menu items) are illustrated from Figure 9.3 to Figure 9.10.

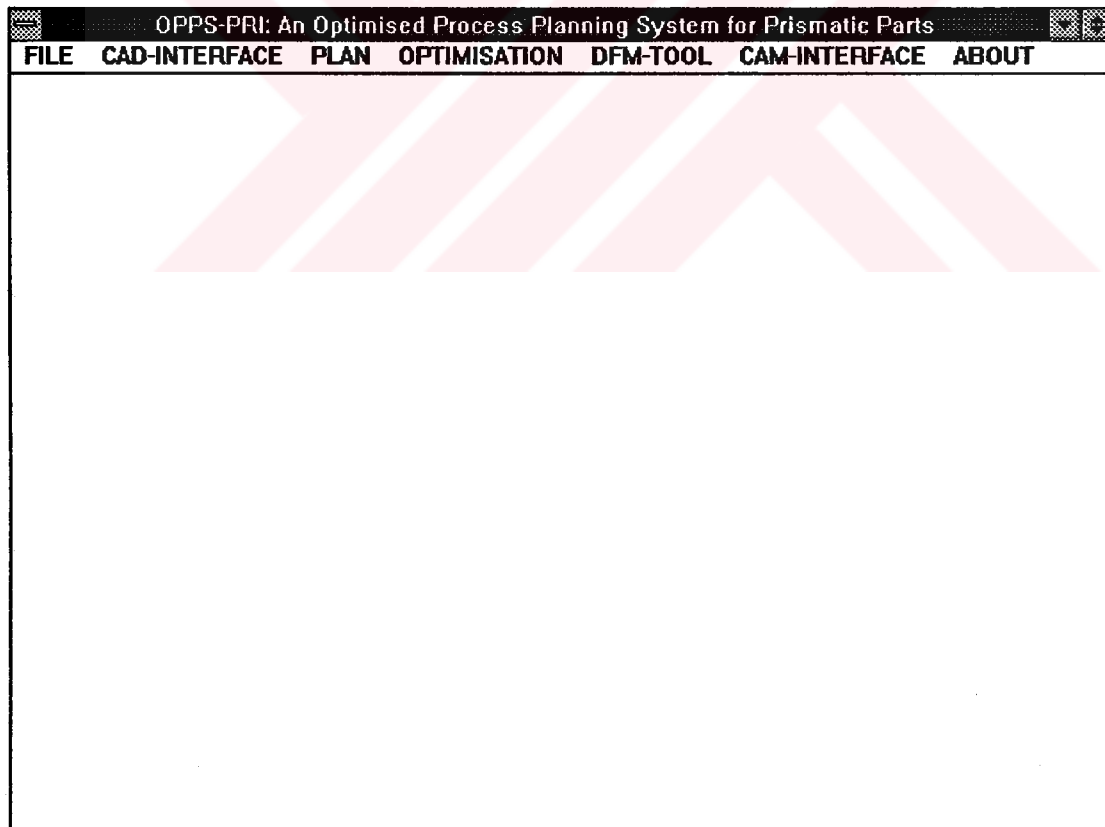


Figure 9.2. Main Menu of the OPPS-PRI on the Opening Screen

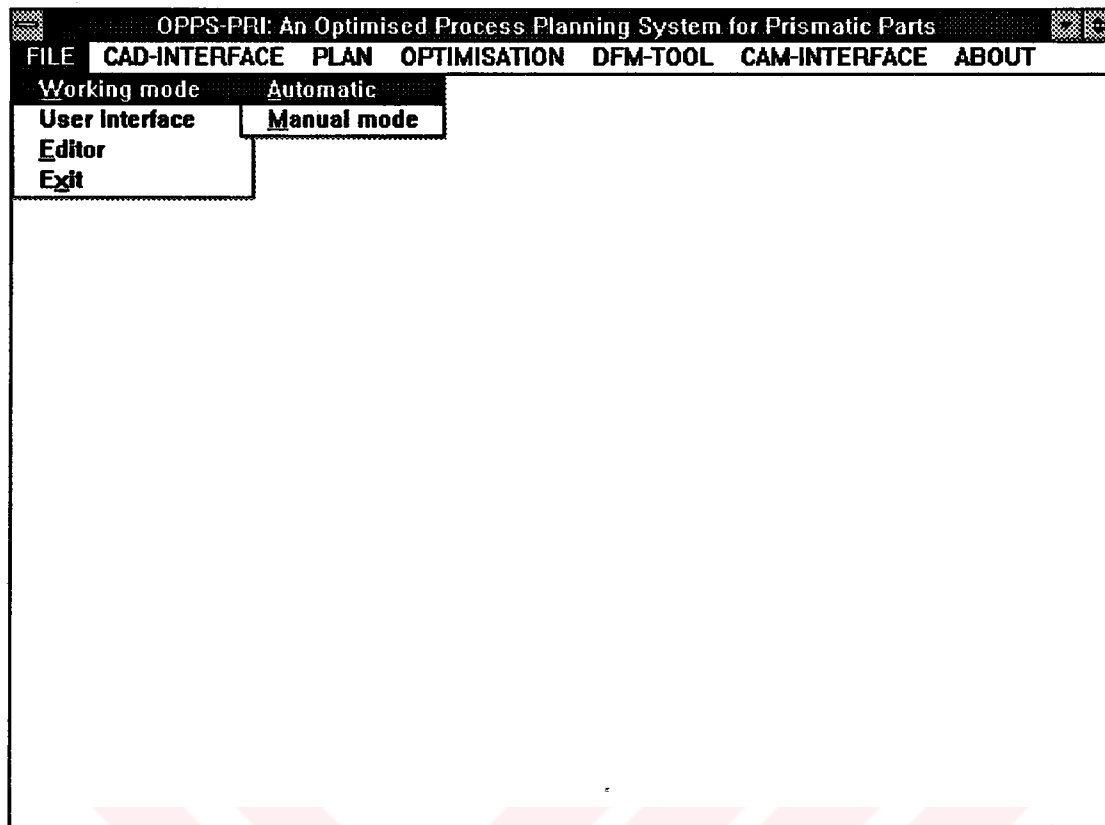


Figure 9.3. FILE Pull-down Menu

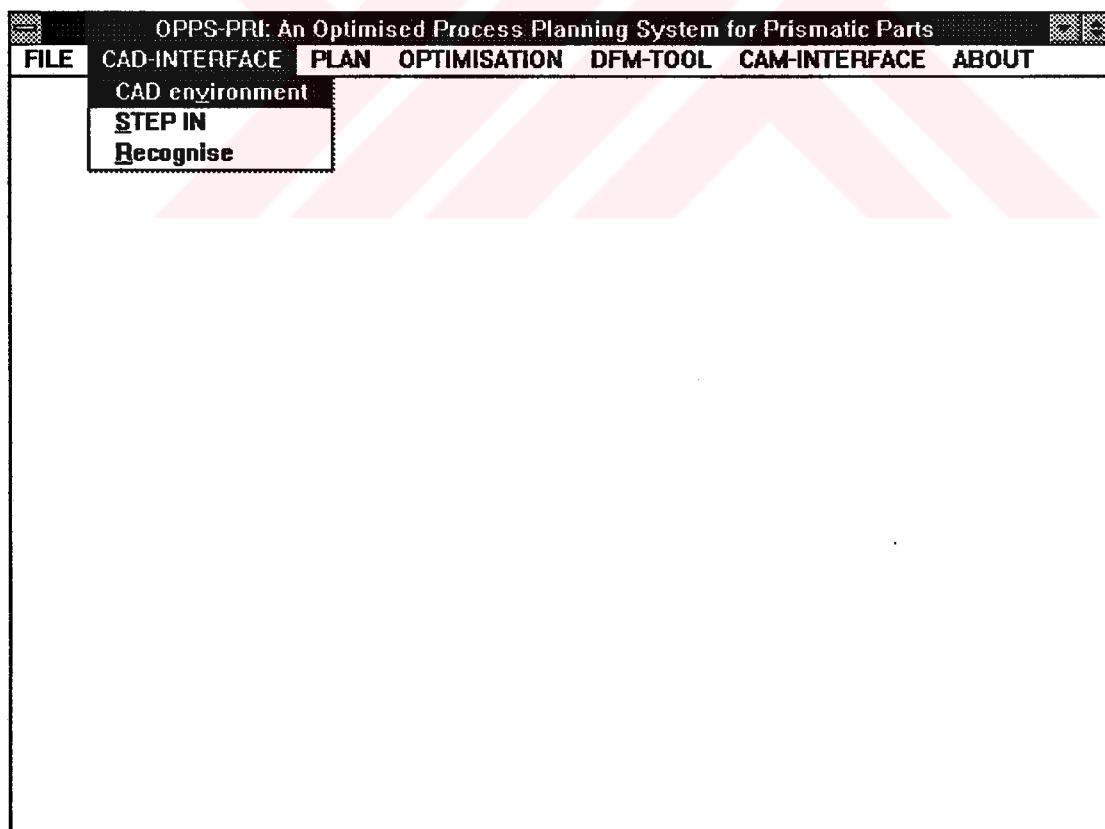


Figure 9.4. CAD Interface Pull-down Menu

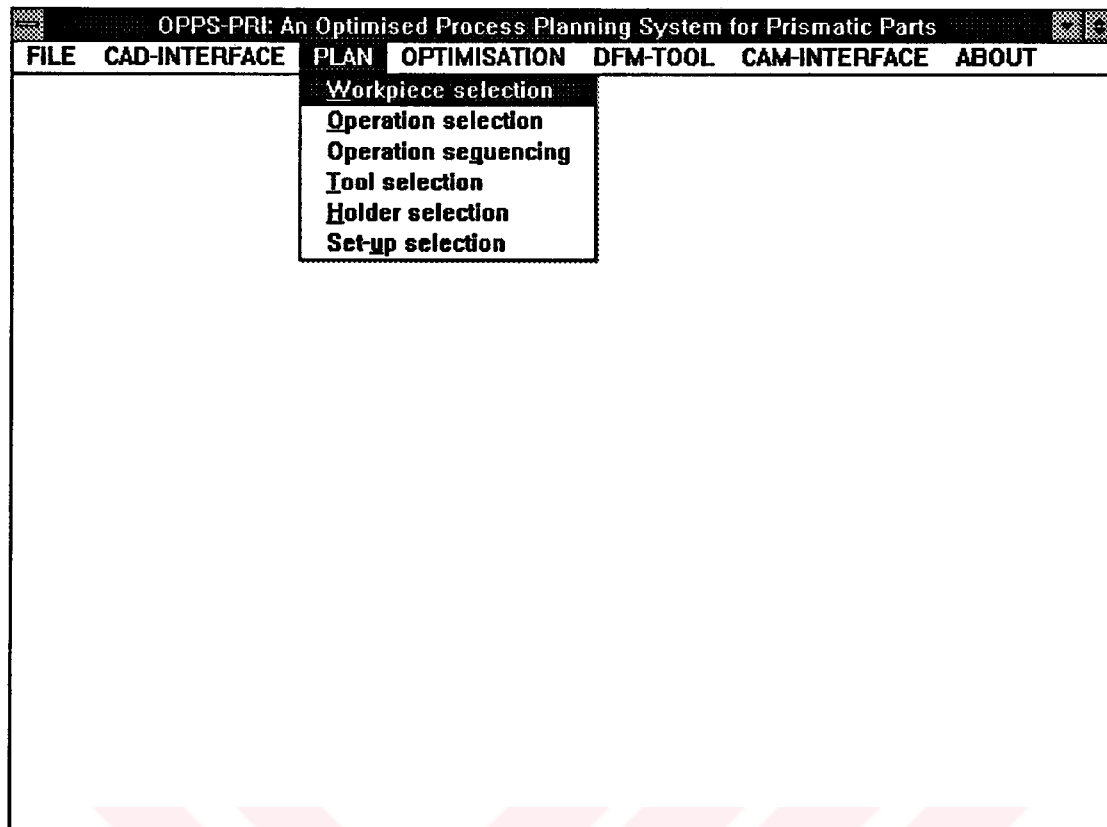


Figure 9.5. PLAN Pull-down Menu

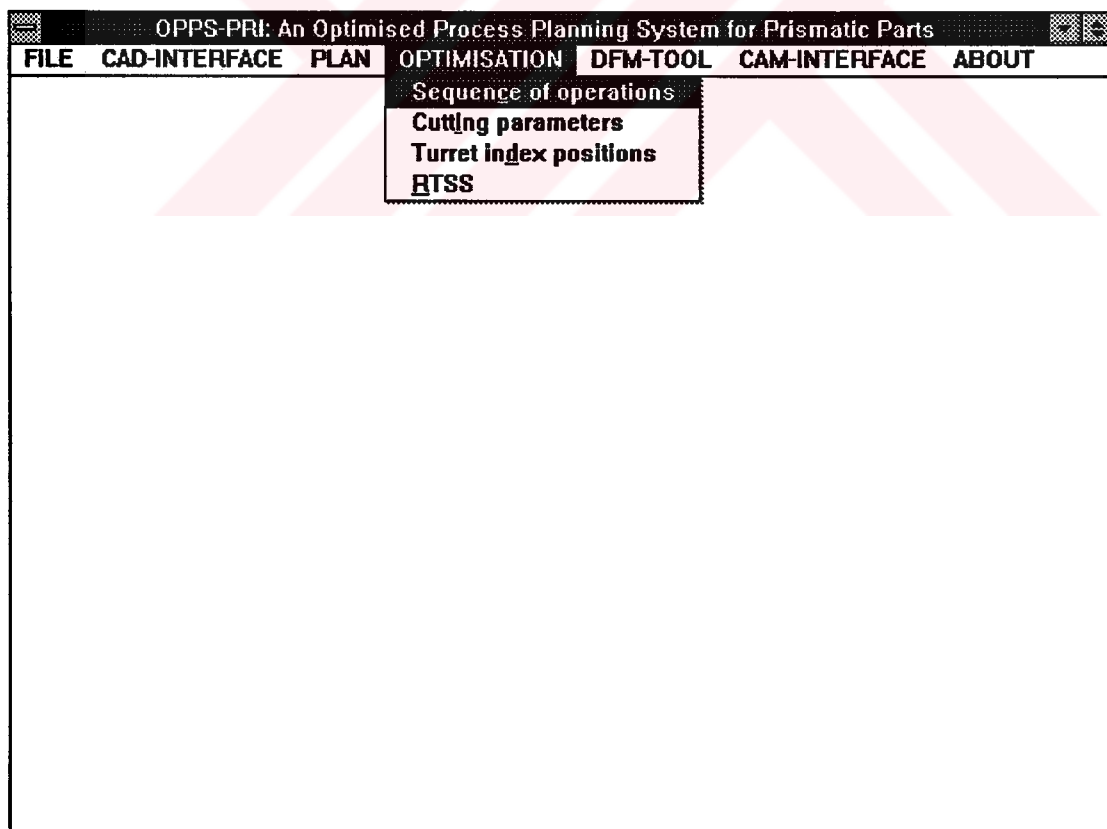


Figure 9.6. OPTIMISATION Pull-down Menu



Figure 9.7. DFM-TOOL Pull-down Menu

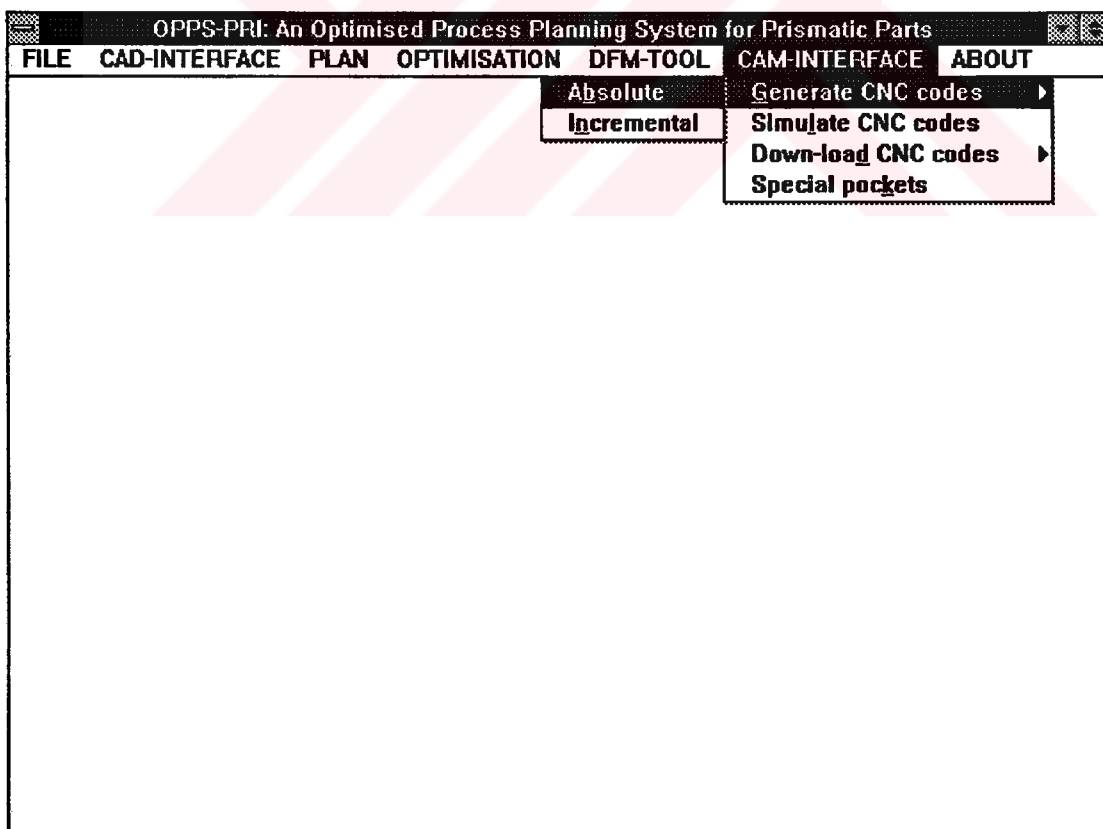


Figure 9.8. CAM-INTERFACE Pull-down Menu (Code Generation Item Is Selected)

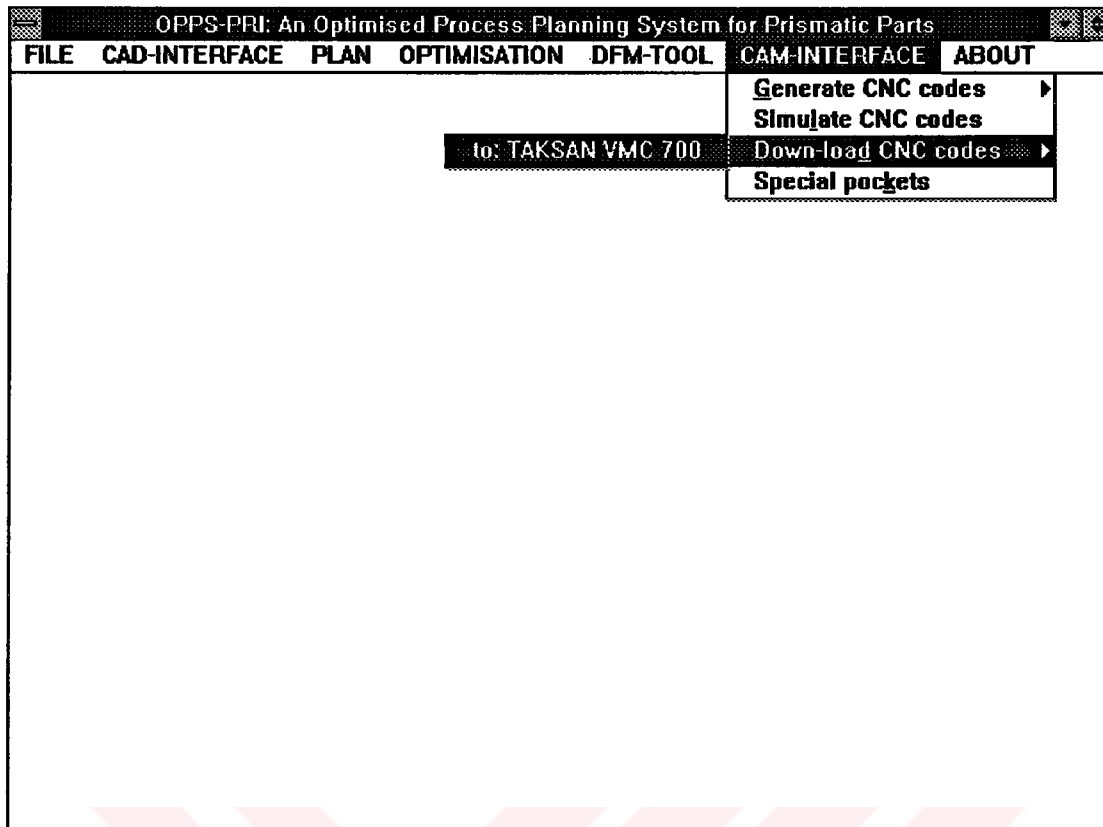


Figure 9.9. CAM-INTERFACE Pull-down Menu (Code Transfer Item Is Selected)

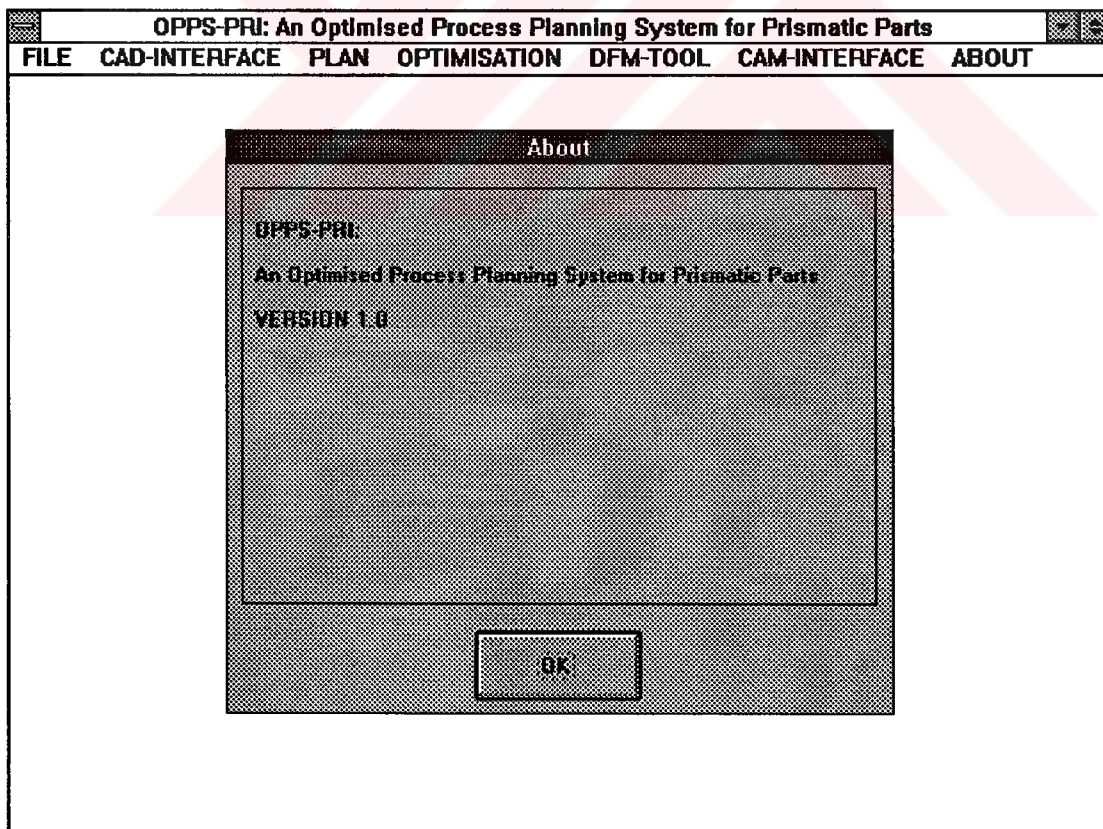


Figure 9.10. Effect of ABOUT Pull-down Menu on the Screen

FILE pull-down menu includes four menu items; Working mode, User interface, Editor, and Exit by which a user can select the working procedure of the system, communicates with the system, can call a district program (i.e., Borland's editor or DOS editor) for editing/manipulating files, and finally can quit the system, respectively. Through Working mode, the user can choose either automatic or manual mode of working procedure to be followed by the system. If the option "*automatic*" is selected in this menu, a unique file name and a pre-determined working order of the system modules are given to the system. Using these inputs, the system automatically determines all the processes, parameters and tools etc. (up to CAM stage) without interacting with the user. However, if the mode "*manual*" is selected, system modules might be used alone, and the user can prepare a process plan or generate a part programme step by step, as he/she wants. In this mode, the user determines his/her own working procedure, such that a system module (i.e., one of the DFM tools) can be skipped over or can be used at different stages of the planning cycle.

CAD-INTERFACE pull-down menu contains the starting modules in the CAD/CAM integration cycle which form the first unit of the OPPS-PRI's structure. It includes three menu items; CAD environment, STEP IN and Recognise. CAD environment menu item is used to call a CAD package (i.e. Pro/Engineer) to design a component. If a STEP output for a component design is already available, this can be input to the system through STEP IN menu item. Feature recognition module is activated by selecting Recognise menu item.

PLAN pull-down menu is composed of six modules responsible for carrying out five usual process planning practices, namely; workpiece selection, operation selection, operation sequencing, tool selection, holder selection and set-up selection. These modules form the second unit of the OPPS-PRI.

OPTIMISATION pull-down menu involves four modules for the optimisation of sequence of operations, cutting parameters, turret index positions and RTSS. RTSS is one of the two modules prepared for the optimisation of sequence of machining

operations using different optimisation criteria, in this pull-down menu. These four modules organise the third unit of the system.

DFM-TOOL pull-down menu includes the Design for Manufacturing (DFM) tool that forms the fourth unit of the OPPS-PRI. It can be used in the CAD/CAM integration cycle through Check critical regions menu item.

CAM-INTERFACE pull-down menu is used for the generation, verification and down-loading of CNC programs by forming the fifth unit of the OPPS-PRI. CNC code generation for arbitrarily shaped pockets with islands can be accomplished through Special pockets menu item. Both absolute and incremental part programs can be generated by selecting Absolute or Incremental through Generate CNC codes menu item from CAM-INTERFACE pull-down menu, as shown in Figure 9.8. In each option, the user is required to give a file name for a CAD/CAM data base (prepared by the up-stream modules of the OPPS-PRI) for which a part program is to be generated and warned for possible errors. The part programs are stored in the text files which can easily be printed through FILE pull-down menu. When the generation process is finished, the system returns automatically to the main menu. The verification of the generated part programs can be performed by calling the simulation module through Simulate CNC codes menu item from CAM-INTERFACE pull-down menu as shown in Figure 9.9.

9.4. CASE STUDY

The use of system modules of OPPS-PRI is illustrated through a sample part in this section. The example component is shown in Figure 9.11. Notice that the faces of holes are not shown in this figure, since the representation of passageways are somewhat different from others. Their representation will be demonstrated in the forthcoming paragraphs. The part can be modelled on solid modelling environment like Pro/Engineer and then information regarding the component can be obtained in the STEP standard format. This data can further be post-processed to secure a formatted B-Rep data structure that can be used as the main input to the feature recognition module. If the formatted B-Rep file for a component is readily available by the user, there is no need to use any post-processor.

The B-Rep data base and STEP data corresponding to the example component are given in Table 9.1 and *Appendix F*, respectively.

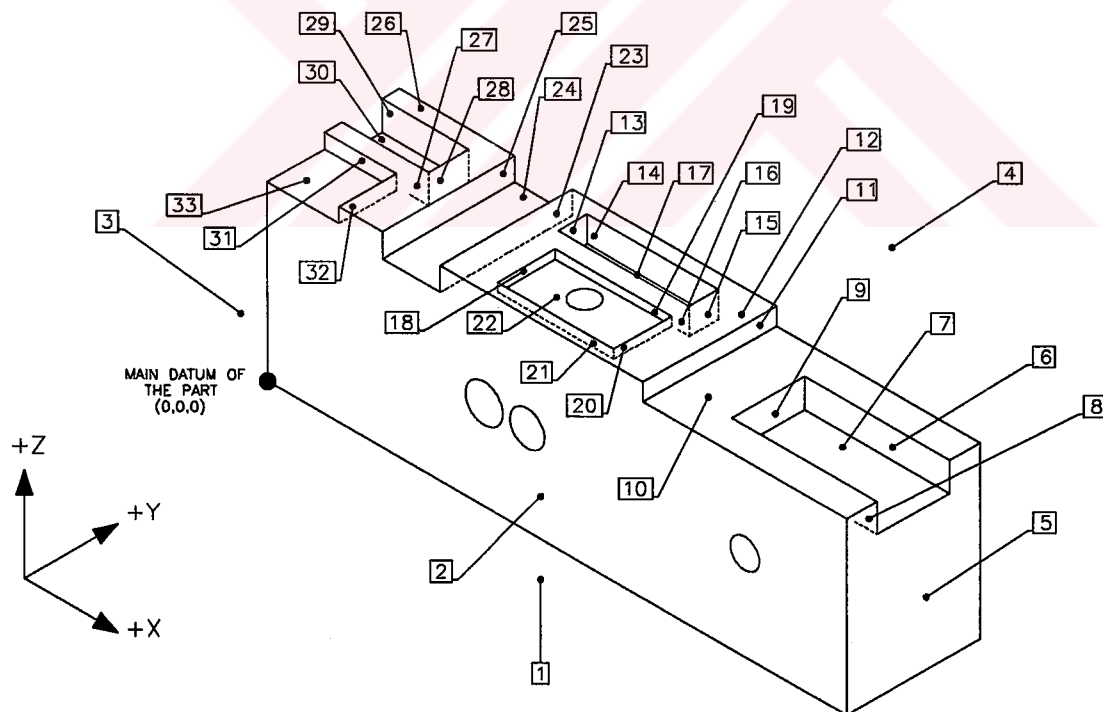


Figure 9.11. The Example Component for Case Study

Table 9.1. The B-Rep Data Base of the Example Component Shown in Fig. 9.11.

Face No	Edge No	v1x	v1y	v1z	v2x	v2y	v2z
1	1	0	0	0	0	46	0
1	2	0	46	0	200	46	0
1	3	200	46	0	200	0	0
1	4	200	0	0	0	0	0
2	1	0	0	0	0	0	60
2	2	0	0	60	25	0	60
2	3	25	0	60	25	0	65
2	4	25	0	65	40	0	65
2	5	40	0	65	40	0	57
2	6	40	0	57	60	0	57
2	7	60	0	57	60	0	65
2	8	60	0	65	130	0	65
2	9	130	0	65	130	0	59
2	10	130	0	59	200	0	59
2	11	200	0	59	200	0	0
2	12	200	0	0	0	0	0
3	1	0	0	0	0	0	60
3	2	0	0	60	0	20	60
3	3	0	20	60	0	20	65
3	4	0	20	65	0	26	65
3	5	0	26	65	0	26	55
3	6	0	26	55	0	40	55
3	7	0	40	55	0	40	65
3	8	0	40	65	0	46	65
3	9	0	46	65	0	46	0
3	10	0	46	0	0	0	0
4	1	0	46	0	0	46	65
4	2	0	46	65	40	46	65
4	3	40	46	65	40	46	57
4	4	40	46	57	60	46	57
4	5	60	46	57	60	46	65
4	6	60	46	65	130	46	65
4	7	130	46	65	130	46	59
4	8	130	46	59	200	46	59
4	9	200	46	59	200	46	0
4	10	200	46	0	0	46	0
5	1	200	0	0	200	0	59
5	2	200	0	59	200	10.5	59
5	3	200	10.5	49	200	35.5	49
5	4	200	35.5	49	200	35.5	59
5	5	200	35.5	59	200	46	59
5	6	200	46	59	200	46	0
5	7	200	46	0	200	0	0
6	1	150	35.5	49	150	35.5	59
6	2	150	35.5	59	200	35.5	59
6	3	200	35.5	59	200	35.5	49
6	4	200	35.5	49	150	35.5	49
7	1	150	10.5	49	150	35.5	49
7	2	150	35.5	49	200	35.5	49
7	3	200	35.5	49	200	10.5	49
7	4	200	10.5	49	150	10.5	49
8	1	150	10.5	49	150	10.5	59
8	2	150	10.5	49	200	10.5	59
8	3	200	10.5	59	200	10.5	49
8	4	200	10.5	49	150	10.5	49
9	1	150	10.5	49	150	10.5	59
9	2	150	10.5	59	150	35.5	59
9	3	150	35.5	59	150	35.5	49
9	4	150	35.5	49	150	10.5	49
10	1	130	0	59	130	46	59
10	2	130	46	59	200	46	59
10	3	200	46	59	200	35.5	59
10	4	200	35.5	59	150	35.5	59
10	5	150	35.5	59	150	10.5	59
10	6	150	10.5	59	200	10.5	59
10	7	200	10.5	59	200	0	59
10	8	200	0	59	130	0	59
11	1	130	0	59	130	0	65

continuing from the previous page							
11	2	130	0	65	130	46	65
11	3	130	46	65	130	46	59
11	4	130	46	59	130	0	59
12	1	1	60	0	65	60	46
12	2	2	60	46	65	130	46
12	3	3	130	46	65	130	0
12	4	4	130	0	65	60	0
12	1	5	75	5	65	75	25
12	2	6	75	25	65	115	25
12	3	7	115	25	65	115	5
12	4	8	115	5	65	75	5
12	1	9	70	31	65	70	41
12	2	10	70	41	69	115	41
12	3	11	115	41	65	115	31
12	4	12	115	31	65	70	31
13	1	70	31	56	70	31	65
13	2	70	31	65	70	41	65
13	3	70	41	65	70	41	56
13	4	70	41	56	70	31	56
14	1	70	41	56	70	41	65
14	2	70	41	65	115	41	65
14	3	115	41	65	115	41	56
14	4	115	41	56	70	41	56
15	1	115	31	56	115	31	65
15	2	115	31	65	115	41	65
15	3	115	41	65	115	41	56
15	4	115	41	56	115	31	56
16	1	70	31	56	70	31	65
16	2	70	31	65	115	31	65
16	3	115	31	65	115	31	56
16	4	115	31	56	70	31	56
17	1	70	31	56	70	41	56
17	2	70	41	56	115	41	56
17	3	115	41	56	115	31	56
17	4	115	31	56	70	31	56
18	1	75	5	62	75	5	65
18	2	75	5	65	75	25	65
18	3	75	25	65	75	25	62
18	4	75	25	62	75	5	62
19	1	75	65	62	75	25	65
19	2	75	25	65	115	25	65
19	3	115	25	65	115	25	62
19	4	115	25	62	75	25	62
20	1	115	5	62	115	5	65
20	2	115	5	65	115	25	65
20	3	115	25	65	115	25	62
20	4	115	25	62	115	5	62
21	1	75	5	62	75	5	65
21	2	75	5	65	115	5	65
21	3	115	5	65	115	5	62
21	4	115	5	62	75	5	62
22	1	1	75	5	62	75	25
22	2	2	75	25	62	115	25
22	3	3	115	25	62	115	5
22	4	4	115	5	62	75	5
22	1	5	80	10	62	80	20
22	2	6	80	20	62	90	20
22	3	7	90	20	62	90	10
22	4	8	90	10	62	80	10
23	1	60	0	57	60	0	65
23	2	60	0	65	60	46	65
23	3	60	46	65	60	46	57
23	4	60	46	57	60	0	57
24	1	40	0	57	40	40	57
24	2	40	46	57	60	46	57
24	3	60	46	57	60	0	57
24	4	60	0	57	40	0	57
25	1	40	0	57	40	0	65
25	2	40	0	65	40	46	65
25	3	40	46	65	40	46	57
25	4	40	46	57	40	0	57

continuing from the previous page							
26	1	0	20	65	0	26	65
26	2	0	26	65	30	26	65
26	3	30	26	65	30	40	65
26	4	30	40	65	0	40	65
26	5	0	40	65	0	46	65
26	6	0	46	65	40	46	65
26	7	40	46	65	40	0	65
26	8	40	0	65	25	0	65
26	9	25	0	65	25	20	65
26	10	25	20	65	0	20	65
27	1	0	26	55	0	26	65
27	2	0	26	65	30	26	65
27	3	30	26	65	30	26	55
27	4	30	26	55	0	26	55
28	1	30	26	55	30	26	65
28	2	30	26	65	30	40	65
28	3	30	40	65	30	40	55
28	4	30	40	55	30	26	55
29	1	0	40	55	0	40	65
29	2	0	40	65	30	40	65
29	3	30	40	65	30	40	55
29	4	30	40	55	0	40	55
30	1	0	26	55	0	40	55
30	2	0	40	55	30	40	55
30	3	30	40	55	30	26	55
30	4	30	26	55	0	26	55
31	1	0	20	60	0	20	65
31	2	0	20	65	25	20	65
31	3	25	20	65	25	20	60
31	4	25	20	60	0	20	60
32	1	25	0	60	25	0	65
32	2	25	0	65	25	20	65
32	3	25	20	65	25	20	60
32	4	25	20	60	25	0	60
33	1	0	0	60	0	20	60
33	2	0	20	60	25	20	60
33	3	25	20	60	25	0	60
33	4	25	0	60	0	0	60
Additional Information							
Represented Feature			# Total Faces (No)		# Total Edges	# Total Vertices	
Blind Circular Depression			5 (34,35,36,37,38)		12	8	
Circular Passage			4 (39,40,41,42)		12	8	
Circular Passage			4 (43,44,45,46)		12	8	
Circular Passage			4 (47,48,49,50)		12	8	
F (FACES)		50	VALIDATION OF MODEL USING EULER'S LAW: $F - E + V - L = 2 (B - G)$ $50 - 135 + 90 - 9 = 2 (1 - 3)$ $- 4 = - 4$ OK.				
E (EDGES)		135					
V (VERTICES)		90					
L (INNER LOOPS)		9					
B (BODIES)		1					
G (GENIUSES)		3					
Other Technological Information							
Surface Roughness (µm)			Face	Face 7	Face 10	Face 30	
			2	2	2	1	
Fillet Radius (mm)			Faces 31-32			Faces 19-20	
			10			10	

The circular holes on the part can be approximated by cubic holes to satisfy Euler's law for polyhedral objects as shown in Figure 9.12. The relation matrix which is automatically derived for the example part and the list of extracted features by the feature recognition module with their codes are presented in Table 9.2 and Table 9.3, respectively. Figure 9.13 shows the extracted features from the relation matrix.



Table 9.2. Relation Matrix Derived for The Example Component

195

Table 9.3. The List of Extracted Feature Types and Their Codes

No	Type	Code														
1	Face	Recognised by geometric reasoning														
2	Blind Step (B-STP)	0	3	0	0	2	2	0	2	2	0	2	2			
3	Blind Slot (B-SLT)	0	2	0	0	2	2	0	2	2	0	2	1			
4	Thru Slot (T-SLT)	0	4	0	0	2	2	0	1	3	0	1	3			
5	Face Slot (F-SLT)	0	1	0	0	3	1	0	3	1	0	3	1	0	3	0
6	Rect. Pocket (R-PKT)	0	1	0	0	3	1	0	3	1	0	3	1	0	3	0
7	Thru Step (T-STP)	0	4	0	0	1	3	0	1	3						
8	Blind Slot (B-SLT)	0	2	0	0	2	2	0	2	2	0	2	1			
9	Blind Hole (B-HOL)	0	1	0	0	3	1	0	3	1	0	3	1	0	3	0
10	Thru Hole (T-HOL)	1	2	0	0	2	2	0	2	2	0	2	2	0	2	2
11	Thru Hole (T-HOL)	1	2	0	0	2	2	0	2	2	0	2	2	0	2	2
12	Thru Hole (T-HOL)	1	2	0	0	2	2	0	2	2	0	2	2	0	2	2

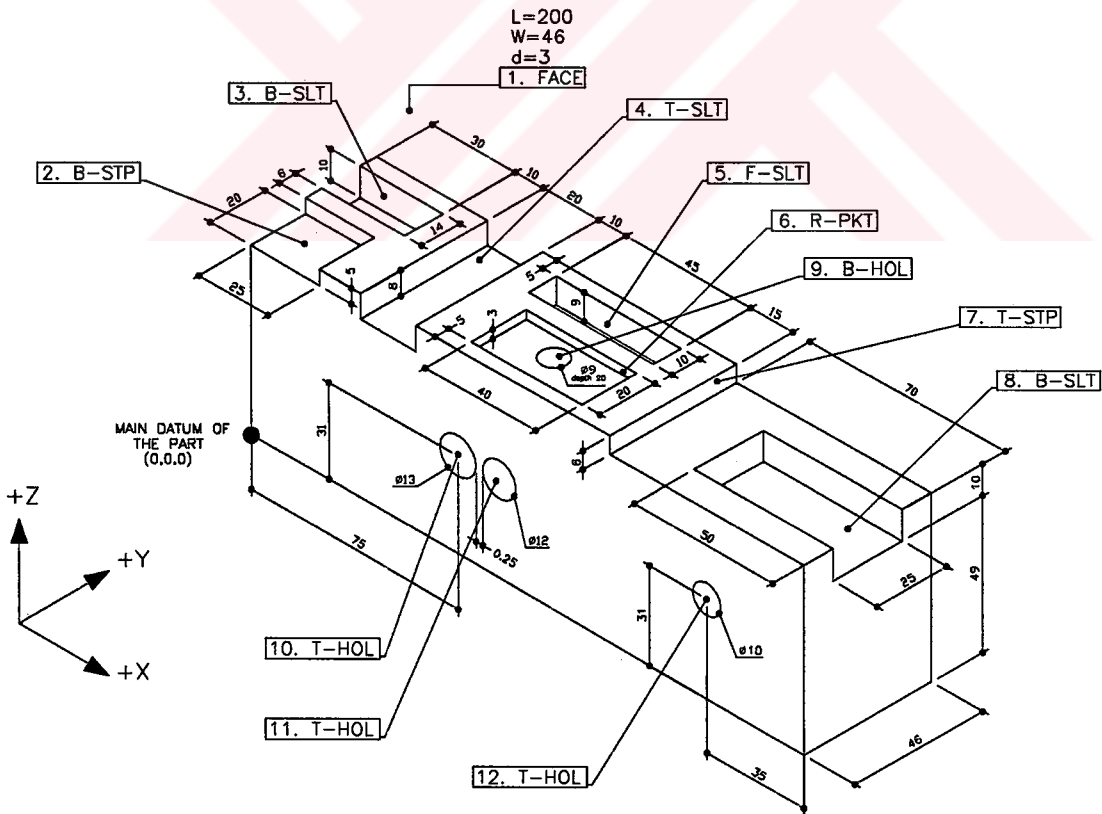


Figure 9.13. Extracted Features on The Component

Table 9.4. Output From the Feature Recogniser

Fea. No	Feature ID	Distance to main datum			Feature Attributes			Approach Directions						SR	FR
		d-x	d-y	d-z	d	w	l	1	2	3	4	5	6	μm	mm
F1	FACE	0	0	65	-	46	200	1	0	0	0	0	0	-	-
F2	B-STP	0	0	60	5	20	25	1	0	0	0	1	1	-	10
F3	B-SLT	0	26	55	10	14	30	1	0	0	0	0	1	-	-
F4	T-SLT	40	0	57	8	20	46	1	0	0	0	0	0	-	-
F5	F-SLT	70	31	56	9	10	45	1	0	0	0	0	0	-	-
F6	R-PKT	75	5	62	3	20	40	1	0	0	0	0	0	-	10
F7	T-STP	130	0	59	6	70	46	1	0	1	0	0	0	-	-
F8	B-SLT	150	10.5	49	10	25	50	1	0	1	0	0	0	-	-
F9	B-HOL	95	15	62	20	9	-	1	0	0	0	0	0	1.7	-
F10	T-HOL	75	0	31	46	13	-	0	1	0	0	1	0	1.7	-
F11	T-HOL	87.75	0	31	46	12	-	0	1	0	0	1	0	1.7	-
F12	T-HOL	165	0	31	46	10	-	0	1	0	0	1	0	1.7	-

SR column represents *Surface Roughness* and FR column stands for *Fillet Radius*

Table 9.4 shows the output of the feature recognition module to down-stream process planning modules. The *workpiece size selection module* is the first process planning module to be executed. The maximum width, length and height of the finished component can easily be calculated by using the information on the first feature (face), which are 46 x 200 x 65 (in mm), respectively. Upon inputting the machining allowances, the size of the raw stock is calculated. The user is then required to select a workpiece material (i.e., 10L50). From the available stocks, it is found that the smallest raw stock size is 46 x 200 x 68 (in mm), and W252 (as shown in Table 9.5) is assigned as the workpiece to be used. The depth of cut for the first feature (Face) is then automatically found (i.e. 3 mm).

Table 9.5. Raw Stock Information

Code	Material Type	Width (mm)	Length (mm)	Height (mm)
W252	10L50	46	200	68

At this stage, it is recommended to perform the set-up planning which can be performed by using the information provided in Table 9.4. The resulting optimal set-up plans are tabulated in Table 9.6. It is worth noticing that since there is no geometrical tolerance imposed on the component, the set-up sequencing is performed in such a way that the set-ups containing the smallest number of features are left to the end of the set-up sequences.

Table 9.6. Optimal Set-up Plans for the Example Part

Optimum Set-up Plan No	# Set-ups	Orientation	Features in set-up
OS-1	S1	AD - 1	F1, F2, F3, F4, F5, F6, F7, F8, F9
	S2	AD - 5	F10, F11, F12
OS-2	S1	AD - 1	F1, F3, F4, F5, F6, F8, F9
	S2	AD - 5	F2, F7, F10, F11, F12
OS-3	S1	AD - 1	F1, F2, F3, F4, F5, F6, F7, F8, F9
	S2	AD - 2	F10, F11, F12

After the features are aggregated into set-ups, the next stage in the process planning is the determination of operations corresponding to the features and their sequences for each set-up. Only the machinability rules are considered in determining the operation sequences. Hence, the operation sequences obtained in this module are rough but reasonable. However, some of the repeated operations like *center drilling* are gathered such that they will follow each other. The resulting sequences of operations for the example part are given in Table 9.7. The modified sequence of operations for the second set-up is also presented in Table 9.8.

Table 9.7. The Sequence of Operation Determined for The Sample Part

Set-up No	Operation Sequence No	Operation	Corresponding Feature
S1	O1	Face Milling	FACE (F1)
S1	O2	Step Milling	T-STP (F7)
S1	O3	Slot Milling	B-SLT (F8)
S1	O4	Slot Milling	T-SLT (F4)
S1	O5	Slot Milling	B-SLT (F3)
S1	O6	Slot Milling	F-SLT (F5)
S1	O7	Step Milling	B-STP (F2)
S1	O8	Pocket Milling	R-PKT (F6)
S1	O9	Center Drilling	B-HOL (F9)
S1	O10	Twist Drilling	B-HOL (F9)
S2	O11	Center Drilling	T-HOL (F10)
S2	O12	Twist Drilling	T-HOL (F10)
S2	O13	Center Drilling	T-HOL (F11)
S2	O14	Twist Drilling	T-HOL (F11)
S2	O15	Center Drilling	T-HOL (F12)
S2	O16	Twist Drilling	T-HOL (F12)

Table 9.8. Modified Sequence of Operations for The Second Set-up

Set-up No	Operation Sequence No	Operation	Corresponding Feature
2	11	Center Drilling	T-HOL (F10)
2	12	Center Drilling	T-HOL (F11)
2	13	Center Drilling	T-HOL (F12)
2	14	Twist Drilling	T-HOL (F10)
2	15	Twist Drilling	T-HOL (F11)
2	16	Twist Drilling	T-HOL (F12)

Cutting tools together with their holders and other auxiliary accessories are then proposed by executing the *tool selection* and *holder selection modules* of the OPPS-PRI. The alternative cutting tools proposed by the *cutting tool selection module* are presented in Table 9.9. The information on the corresponding tooling and specifications are also given in Table 9.10 and Table 9.11, respectively.

Table 9.9. Cutting Tool(s) versus Operations

Operations→ Tools ↓	O1 (F1)	O2 (F7)	O3 (F8)	O4 (F4)	O5 (F3)	O6 (F5)	O7 (F2)	O8 (F6)	O9 (F9)	O10 (F9)	O11 (F10)	O12 (F11)	O13 (F12)	O14 (F10)	O15 (F11)	O16 (F12)
T1	1	1														
T2					1											
T3						1	1	1								
T4		1		1												
T5		1	1													
T6																1
T7										1						
T8															1	
T9									1		1	1	1			
T10														1		

Table 9.10. Tooling Information

Cutter	Type	Cutter	Adaptor	Insert
T1	Modern	ISCAR F90SD-D63-22-12	BT 40 KBM 22	SDTM 1205 PDR-HQ
T2	Solid	-	BT40 PB 444 E	-
T3	Solid	-	BT 40 VT 10	-
T4	Modern	ISCAR E90A-D20-CS20	BT 40 VT 20	APKT 1003 PDTR-76
T5	Modern	ISCAR E90SP-D25-W25-10	BT40 VT 25	SPMT 100408 TR-HQ
T6	Solid	-	BT 40 PB 444 E-10	-
T7	Solid	-	BT 40 PB 444 E-9	-
T8	Solid	-	BT 40 VT 12	-
T9	Solid	-	BT 40 MM B12 with (0-10mm mandren)	
T10	Solid	-	BT 40 MM B16 with (4-16mm mandren)	

Table 9.11. Specifications of Cutters

Tool No	Tool Type	Quality	Dia. (mm)	Cutting Length (mm)	# of Teeth	Shank Diameter (mm)	Helix Angle (°)	al (°)	ac (°)
T1	Face mill	Carbide	63	12	6	22	-	-	-
T2	End mill	HSS	14	26	4	12	45	0	5
T3	End mill	HSS	10	24	3	10	45	0	5
T4	End mill	Carbide	20	10	3	20	-	-	-
T5	End mill	Carbide	25	10	2	25	-	-	-
T6	Twist drill	HSS	10	88	-	10			
T7	Twist drill	HSS	9	82	-	9			
T8	Twist drill	HSS	12	100	-	12			
T9	Center drill	HSS	-	24	-	4			
T10	Twist drill	HSS	13	100	-	13			

The next step is the optimisation. The sequence of operations, cutting parameters and the positions of cutters on the wheel of automatic tool changer can be optimised through the menu items in OPTIMISATION pull-down menu. For the optimisation of sequence of operations, there are two optional modules to be used. One of them uses GA while the other is a heuristic based system (RTSS). However, it is preferable to optimise the sequence of operations first by the top-most item (through Sequence of operations menu item) in the OPTIMISATION pull-down menu, and then to use the RTSS. By this way, optimal sequence of operations can be found based on the safety, minimum tool change and combined criteria. Table 9.12 presents the reward-penalty matrix (for the first set-up) prepared for use with the GA based optimisation strategy. As can be seen in this table, Operation 1 has a priority over all of the operations, Operation 7 has a preference over Operation 3, and finally Operation 1 and Operation 10 has a precedence over Operation 8. Optimal alternative sequences of operations found in the first set-up by the GA based system are given in Table 9.13. Although there are alternatives sequences for the second set-up, an optimal sequence is presented in Table 9.14. Notice that, there may be a optimal sequence among the alternatives proposed by the GA based optimisation module which can also satisfy the minimum tool change constraint.

Table 9.12. Reward/Penalty Matrix of the Sample Part (First Set-up Operations)

Oprs.	O1 (F1)	O7 (F2)	O5 (F3)	O4 (F4)	O6 (F5)	O8 (F6)	O2 (F7)	O3 (F8)	O9 (F9)	O10 (F9)
O1(F1)	∞	20	20	20	20	20	20	20	20	20
O2(F2)	-10	∞	5	5	5	5	5	5	5	5
O3(F3)	-10	5	∞	5	5	5	5	5	5	5
O4(F4)	-10	5	5	∞	5	5	5	5	5	5
O5(F5)	-10	5	5	5	∞	5	5	5	5	5
O6(F6)	-10	5	5	5	5	∞	5	5	85	85
O7(F7)	-10	5	5	5	5	5	∞	-100	5	5
O8(F8)	-10	5	5	5	5	5	110	∞	5	5
O9(F9)	-10	5	5	5	5	-75	5	5	∞	-50
O10 (F9)	-10	5	5	5	5	-75	5	5	60	∞

Table 9.13. Optimum Sequence of Operations in The First Set-up Based on "Safety"

Alternative	1	2	3	4	5	6	7	8	9	10	Fitness
1	O1	O7	O5	O4	O6	O9	O10	O8	O2	O3	-180
2	O1	O5	O4	O6	O7	O2	O3	O9	O10	O8	-180
3	O1	O7	O9	O10	O8	O2	O3	O6	O4	O5	-180
4	O1	O9	O10	O6	O7	O8	O3	O5	O4	O2	-180

Table 9.14. Optimum Sequence of Operations in The Second Set-up

<i>11</i>	<i>12</i>	<i>13</i>	<i>14</i>	<i>15</i>	<i>16</i>
O11	O12	O13	O14	O15	O16

Input to the RTSS is the priority matrix shown in Table 9.14 which is actually a different representation of the reward-penalty matrix as shown in Table 9.15. Table 9.16 shows the operations and corresponding cutters to be used. A single cutter is then assigned for each operation as illustrated in Table 9.17.

Table 9.15. Priority Matrix (for The First Set-up)

Operations→ ↓	O1 (F1)	O2 (F7)	O3 (F8)	O4 (F4)	O5 (F3)	O6 (F5)	O7 (F2)	O8 (F6)	O9 (F9)	O10 (F9)	UM	RM
O1(F1)		1	1	1	1	1	1	1	1	1	1	1
O2 (F7)			1								2	2
O3 (F8)											3	10
O4 (F4)											2	10
O5 (F3)											2	10
O6 (F5)											2	10
O7 (F2)											2	10
O8 (F6)									1	1	2	8
O9 (F9)										1	9	9
O10 (F9)											10	10

Table 9.16. Tool(s) versus Operations (The First Set-up)

Operations→ Tools ↓	O1 (F1)	O2 (F7)	O3 (F8)	O4 (F4)	O5 (F3)	O6 (F5)	O7 (F2)	O9 (F9)	O10 (F9)	O8 (F6)
T1	1	1								
T2					1					
T3						1	1			1
T4		1		1						
T5		1	1							
T7									1	
T9								1		

Table 9.17. Tool versus Operations (The First Set-up)

Operations→ Tools ↓	O1 (F1)	O2 (F7)	O3 (F8)	O4 (F4)	O5 (F3)	O6 (F5)	O7 (F2)	O9 (F9)	O10 (F9)	O8 (F6)
T1	1	1								
T2					1					
T3						1	1			1
T4				1						
T5			1							
T7									1	
T9								1		

Table 9.18. Optimal Block Structure: Optimal Sequence of Operations with Their Assigned Tools for Minimum Tool Change (The First Set-up)

Operations→ Tools ↓	O1	O2	O3	O4	O5	O9	O10	O8	O6	O7
T1	1	1								
T2					1					
T3								1	1	1
T4				1						
T5			1							
T7							1			
T9						1				

By taking the UM and RM values and the selected tools into account, operations which use the same tool are grouped so that the number of tool change is minimised. The optimum sequence of operations for the first set-up is shown in Table 9.18. The number of tool change is reduced from 7 to 6 with this optimisation. Notice that the sequence found by the RTSS also satisfies the safety requirements. The selected tools for the operations of the second set-up and sequence of those operations within the set-up are shown in Table 9.19. As a summary, the selected tools for all of the operations (for both set-ups) are presented in Table 9.20.

Table 9.19. Tool versus Operations (The Second Set-up)

Operations→ Tools ↓	O11 (F10)	O12 (F11)	O13 (F12)	O14 (F10)	O15 (F11)	O16 (F12)
T6						1
T8					1	
T9	1	1	1			
T10				1		

Table 9.20. Cutting Tools Assigned to Operations (for Both Set-ups)

Operation Seq.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Operation No.	O1	O2	O3	O4	O5	O9	O10	O8	O6	O7	O11	O12	O13	O14	O15	O16
Cutting Tools	T1	T1	T5	T4	T2	T9	T7	T3	T3	T3	T9	T9	T9	T10	T8	T6

Determination of optimal positions of cutters on the ATC wheel can be performed by selecting Turret index positions menu item in the OPTIMISATION pull-down menu. The user is required to specify the tool capacity, type of ATC movement and indexing time between two adjacent slots. In this case

study, the target machine is assumed to be TAKSAN TMC-700 VMC which has a bi-directional ATC capacity of 16 and an indexing time of 0.69 seconds. The proposed alternative position sets for the problem are tabulated in Table 9.21. The resulting configuration on the ATC is depicted in Figure 9.14.

Table 9.21. Proposed Position Sets

Itr. No	# Iterations / Elapsed Time	Positions on ATC										Fitness - # of Unit Rotations	Fitness - Total Index Time
		1	2	3	4	5	6	7	8	9	10		
1	1 / 5 sec.	T3	T7	T4	T5	T1	T2	T9	T10	T8	T6	20	13.80 s.
2	2 / 12 sec.	T8	T6	T10	T3	T9	T2	T7	T1	T4	T5	19	13.11 s.
3	3 / 15 sec.	T1	T5	T4	T7	T3	T9	T2	T10	T8	T6	15	10.35 s.
4	20 / 50 sec.	T1	T5	T4	T3	T2	T7	T9	T10	T8	T6	15	10.35 s.
5	30 / 80 sec.	T1	T5	T4	T2	T7	T3	T9	T10	T8	T6	13	8.97 s.

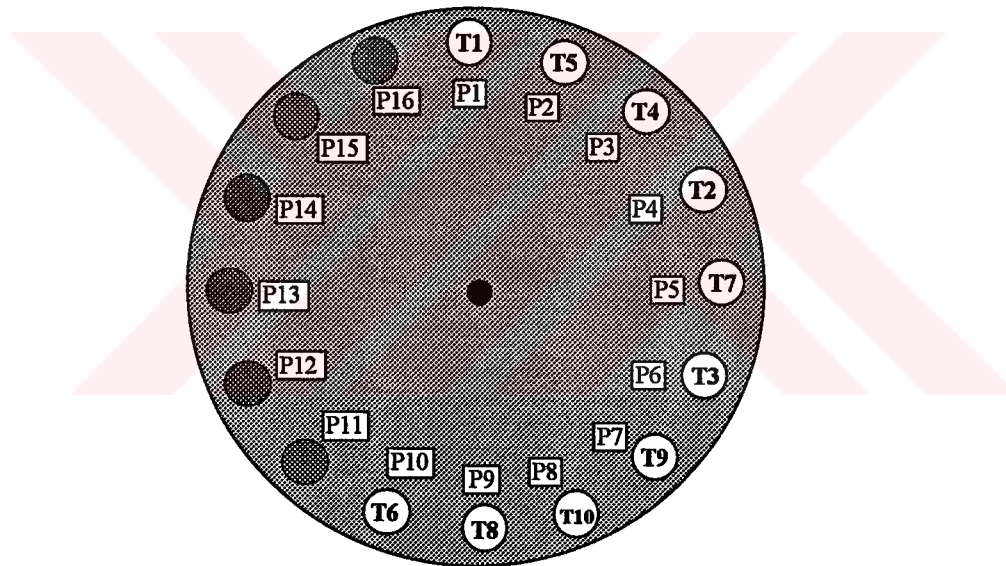


Figure 9.14. Tool Arrangement on the 16-Station ATC of TAKSAN TMC-700 VMC

Optimisation of the cutting parameters are then performed by executing CPOS through Cutting parameters menu item in the OPTIMISATION pull-down menu. For the case study, *minimum production time* is selected as the objective function. The machining operations required to produce the example part and their related parameters are summarised in Table 9.22. Optimum cutting parameters (except those for drilling operations) to be used for machining operations are found by the CPOS with 25 iterations in the GA and presented in Table 9.23.

Table 9.22. Machining Operations and Their Requirements

Opr. No.		Operation Type	Tool No	Total depth of cut (mm)	Distance to be travelled (mm)	Surface Roughness (μm)	Fillet Radius (mm)
1	O1	Face milling	T1	3	273	2	-
2	O2	Step milling	T1	6	238	2	-
3	O3	Slot milling	T5	10	80	2	-
4	O4	Slot milling	T4	8	67.5	-	-
5	O5	Slot milling	T2	10	42	1	-
6	O9	Center drilling	T9	-	-	-	-
7	O10	Twist drilling	T7	20	28	1.7	-
8	O8	Pocket milling	T3	3	80	-	10
9	O6	Slot milling	T3	9	35	-	-
10	O7	Step milling	T3	5	80	-	10
11	O11	Center drilling	T9	2	7	-	-
12	O12	Center drilling	T9	2	7	-	-
13	O13	Center drilling	T9	2	7	-	-
14	O14	Twist drilling	T10	46	52	1.7	-
15	O15	Twist drilling	T8	46	52	1.7	-
16	O16	Twist drilling	T6	46	52	1.7	-

Table 9.23. Cutting Parameters for The Example Part

Operation Seq./ No		Pass	depth of cut (mm)	V (m/min)	f (mm/tooth)	N (rpm)	f (mm/min)
1	O1	1	3	24.8	0.480	125	360
2	O2	1 of 1	3	25.6	0.384	129	298
2	O2	2 of 1	3	192.1	0.449	970	2613
2	O2	1 of 2	3	25.6	0.384	129	298
2	O2	2 of 2	3	192.1	0.449	970	2613
3	O3	1	3	64.0	0.256	814	416
3	O3	2	3	64.0	0.256	814	416
3	O3	3	3	64.0	0.256	814	416
3	O3	4	1	192.2	0.263	2447	1287
4	O4	1	3	49.6	0.248	789	587
4	O4	2	3	49.6	0.248	789	587
4	O4	3	2	60.3	0.298	959	857
5	O5	1	3	38.4	0.284	873	992
5	O5	2	3	38.4	0.284	873	992
5	O5	3	3	38.4	0.284	873	992
5	O5	4	1	105.0	0.288	2387	2750
6	O9	1	2	-	-	1000	50
7	O10	1	20	-	-	250	35
8	O8	1	3	51.2	0.268	1630	1310
9	O6	1	3	51.2	0.262	1630	1280
9	O6	2	3	51.2	0.262	1630	1280
9	O6	3	3	51.2	0.262	1630	1280
10	O7	1	2.5	54.8	0.260	1744	1360
10	O7	2	2.5	54.8	0.260	1744	1360
11	O11	1	2	-	-	1000	50
12	O12	1	2	-	-	1000	50
13	O13	1	2	-	-	1000	50
14	O14	1	46	-	-	250	50
15	O15	1	46	-	-	250	50
16	O16	1	46	-	-	250	50

Before generating the part programs, the example part can be controlled whether there exist a critical region between the features, or not. This process can be fulfilled by selecting Check Critical Regions menu item in the DFM-TOOL pull-down menu. The torque required to machine *Feature 11* (Operation 15) is about 20 N.m. The distance between the borders of two holes (*Feature 10* and *Feature 11*) is 0.25 mm. τ is calculated as 346 MPa by using Equation 6.4. S_{sy} of the workpiece material (10L50) is 290 MPa which is smaller than τ . Therefore, *Feature 11* cannot be machined safely after *Feature 10* under the specified machining conditions. The system offers several type of remedies to solve the problem like *material change*, *using modified cutting parameters* or *shift of features*. It is preferable to use the last option (if possible), since both the first and second options affect many process planning variables and so require re-calculations. However, if it is forbidden to shift the features on the component, the other options can also be used.

Using the *shift of features* type of remedy, it is found that the minimum safe distance between those features (*Feature 10* and *Feature 11*) should be 0.3 mm. The distance between the features is then automatically increased to 1 mm by using a safety factor of 3.33 as shown in Table 9.24. The geometrical data base of the component is immediately refreshed.

Table 9.24. Modification of The Critical Distance Between The Features

Features	Minimum Distance (mm)	Factor of Safety	Modified Distance (mm)
F10 - F11	0.25	3.33	1

The next stage is to generate part programs. Part program generation using either incremental or absolute programming scheme can be accomplished through Generate CNC codes menu item in the CAM-INTERFACE pull-down menu. CNC code generation module generates part programs by processing the CAD/CAM data base prepared by all upstream modules of OPPS-PRI. The CAD/CAM data base contains all the information necessary to generate a part program; from operation sequences to cutting parameters. The CAD/CAM data base of the example component shown in Figure 9.11 includes all the information contained in Table 9.5, Table 9.6, Table 9.10, Table 9.11, Table 9.20, Table 9.21, Table 9.22 and Table 9.23,

in a more compact form. In order not to make duplication, the CAD/CAM data base of the worked part (shown in Figure 9.11) is not given here. The structure of the CAD/CAM data base is similar to that given in Table 8.1.

It should also be noticed that since the part can be machined in 2 set-ups, for each set-up, a different workpiece zero is used. The workpiece zeros to be used in programming of the example part are shown in Figure 9.15. The generated part program in absolute coordinates for the example component is given in Table 9.25.

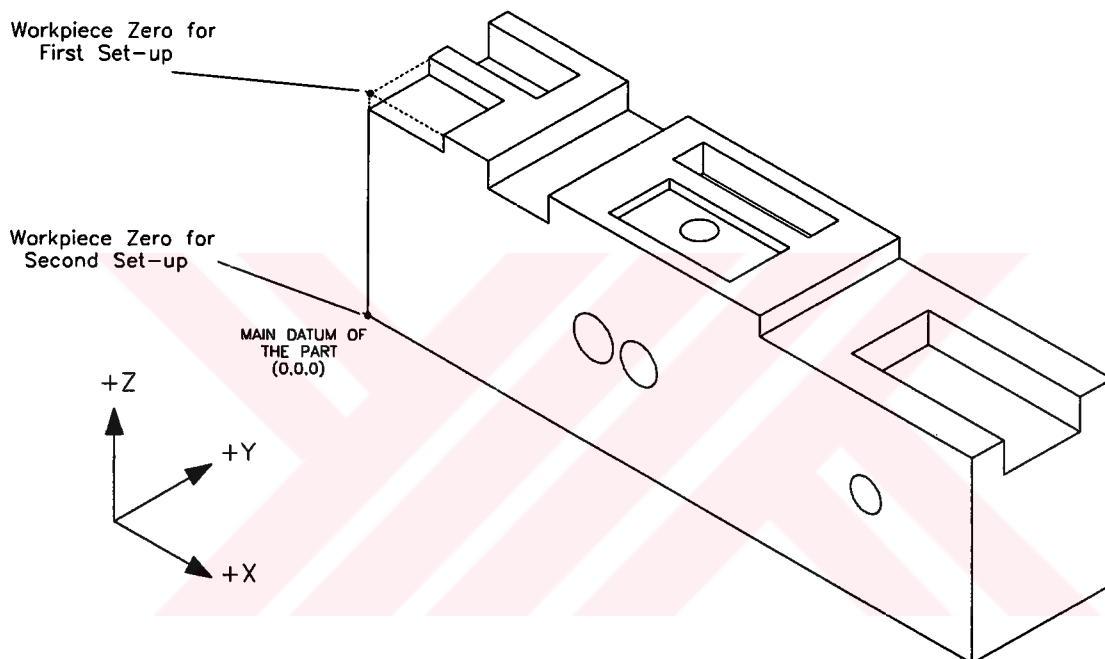


Figure 9.15. Workpiece Zeros for The Example Part

Table 9.25. Absolute Part Program Generated for Example Part

```

%
O0002
N10 G91 G28 X0 Y0 Z0
N20 M03 S125
N30 G00 G90 G43 G54 X-36.5 Y23 Z10 H1
N40 G00 Z0
N50 G01 X236.5 F360
N60 G00 Z10
N70 S129
N80 G00 G90 G43 G54 X173.5 Y-36.5
N90 G00 Z-3
N100 G01 Y82.5 F298
N110 G00 X161.5

```

continuing from the previous page

N120 S970
N130 G01 Y-36.5 F2613
N140 G00 X173.5 Z-6
N150 S129
N160 G01 Y82.5 F298
N170 G00 X161.5
N180 S970
N190 G01 Y-36.5 F2613
N200 G00 Z10
N210 G91 G28 X0 Y0 Z0
N220 M05
N230 M06 T2 /* Tool 5 in Position 2 */
N240 M03 S814
N250 G00 G90 G43 G54 X217.5 Y23 Z10 H5
N260 G00 Z-9
N270 G01 X162.5 F416
N280 G00 Z-4
N290 G00 X217.5
N300 G00 Z-12
N310 G01 X162.5
N320 G00 Z-7
N330 G00 X217.5
N340 G00 Z-15
N350 G01 X162.5
N360 G00 Z-10
N370 G00 X217.5
N380 G00 Z-16
N390 S2447
N400 G01 X162.5 F1287
N410 G00 Z10
N420 G91 G28 X0 Y0 Z0
N430 M05
N440 M06 T3 /* Tool 4 in Position 3 */
N450 M03 S789
N460 G00 G90 G43 G54 X50 Y-15 Z10 H4
N470 G00 Z-3
N480 G01 Y61 F587
N490 G00 Z-6
N500 G01 Y-15
N510 G00 Z-8
N520 S959
N530 G01 Y61 F857
N540 G00 Z10
N550 G91 G28 X0 Y0 Z0
N560 M05
N570 M06 T4 /* Tool 2 in Position 4 */
N580 M03 S873
N590 G00 G90 G43 G54 X-12 Y33 Z10 H2
N600 G00 Z-3
N610 G01 X23 F992
N620 G01 Z-6 F100
N630 G01 X-12 F992
N640 G00 Z-9
N650 G01 X23
N660 G01 Z-10 F100
N670 S2387
N680 G01 X-12 F2750
N690 G00 Z10
N700 G91 G28 X0 Y0 Z0

continuing from the previous page

N710 M05
N720 M06 T7 /* Tool 9 in Position 7 */
N730 M03 S1000
N740 G00 G90 G43 G54 X95 Y15 Z10 H9
N750 G00 Z5
N760 G01 Z-2 F50
N770 G00 Z10
N780 G91 G28 X0 Y0 Z0
N790 M05
N800 M06 T5 /* Tool 7 in Position 5 */
N810 M03 S250
N820 G00 G90 G43 G54 X95 Y15 Z10 H7
N830 G00 Z5
N840 G01 Z-23 F35
N850 G00 Z10
N860 G91 G28 X0 Y0 Z0
N870 M05
N880 M06 T6 /* Tool 3 in Position 6 */
N890 M03 S1630
N900 G00 G90 G43 G54 X80 Y10 Z10 H3
N910 G00 Z1
N920 G01 Z-3 F100
N930 G01 X110 F1310
N940 G01 Y20
N950 G01 X80
N960 G01 Y10
N970 G00 Z10
N980 G00 X75 Y36
N990 G00 Z1
N1000 G01 Z-3 F100
N1010 G01 X110 F1280
N1020 G01 Z-6 F100
N1030 G01 X75 F1280
N1040 G01 Z-9 F100
N1050 G01 X110 F1280
N1060 G00 Z10
N1070 G00 X-10 Y5
N1080 G00 Z-2.5
N1090 S1744
N1100 G01 X20 F1360
N1110 G01 Y15
N1120 G01 X-10
N1130 Y5
N1140 G00 Z-5
N1150 G01 X20
N1160 G01 Y15
N1170 G01 X-10
N1180 G01 X20
N1190 G01 Y0
N1200 G01 X-10
N1210 G00 Z10
N1220 G91 G28 X0 Y0 Z0
N1230 M05
N1240 M06 T7 /* Tool 9 in Position 7 */
N1250 M01
2nd. SET-UP
N1260 M3 S1000
N1270 G00 G90 G43 G55 X75 Y31 Z10 H11 /* H11-Tool length of Tool 9 for the 2nd set-up */
N1280 G00 Z5

continuing from the previous page

N1290 G01 Z-2 F50
N1300 G00 Z10
N1310 G00 X88.5 Y31
N1320 G00 Z5
N1330 G01 Z-2 F50
N1340 G00 Z10
N1350 X165 Y31
N1360 G00 Z5
N1370 G01 Z-2 F50
N1380 G00 Z10
N1390 G91 G28 X0 Y0 Z0
N1400 M05
N1410 M06 T8 /* Tool 10 in Position 8 */
N1420 M03 S250
N1430 G00 G90 G43 G55 X75 Y31 Z10 H10
N1440 G00 Z5
N1450 G01 Z-51 F50
N1460 G00 Z10
N1470 G91 G28 X0 Y0 Z0
N1480 M05
N1490 M06 T9 /* Tool 8 in Position 9 */
N1500 M03 S250
N1510 G00 G90 G43 G55 X88.5 Y31 Z10 H8
N1520 G00 Z5
N1530 G01 Z-51 F50
N1540 G00 Z10
N1550 G91 G28 X0 Y0 Z0
N1560 M05
N1570 M06 T10 /* Tool 6 in Position 10 */
N1580 M03 S250
N1590 G00 G90 G43 G55 X165 Y31 Z10 H6
N1600 G00 Z5
N1610 G01 Z-51 F50
N1620 G00 Z10
N1630 G91 G28 X0 Y0 Z0
N1640 M05
N1650 M02
%

The generated part programs can be verified by selecting Simulate CNC codes menu item in the CAM-INTERFACE pull-down menu. To avoid repetition of the work, CNCEZ simulation software is used as an auxiliary tool in the verification of CNC part programs in this study. After the verification, the part programs can directly be down-loaded to the machine tool controller unit (by picking Down load CNC codes menu item in the CAM-INTERFACE pull-down menu) through the RS-232 serial interface, and then machining of the component starts.

CHAPTER X

DISCUSSION AND CONCLUSIONS

10.1. INTRODUCTION

Evolution of the developed CAD/CAM package and problems faced in the development of the system are discussed in Section 10.2. Conclusions and contributions of the work are given in Section 10.3. Section 10.4 is devoted to the recommendations for future research.

10.2. DISCUSSION

Since the use of computers to assist process planning activities was first discussed by Niebel in 1965 [101], more than 30 years elapsed. In last 30-40 years, CAD and CAM technologies progressed independently in the field of mechanical engineering. A lot of geometric modelling systems have been developed to automate CAD works while a huge amount of NC programming systems have been put into practice to automate CAM facilities. After a while, two independent “*islands of automation*” were obtained. Bridging and integrating these two islands was recognised as a fundamental problem to realise a combined CIM system. Particular emphasis has been given to automation and integration of CAD/CAM systems for machined parts during the last decade. Several integration approaches such as GT (Group Technology) have been proposed, combining CAD and CAM data bases and

automated process planning (CAPP) [182]. Among these, CAPP has been accepted as the key technology to integrate CAD and CAM systems, and as a result it has been an active research area for 30 years. Many researches have been carried out on CAPP and numerous papers have been published in the last two decades [105].

It is difficult to apply a uniform approach to automate the process planning, since it has a diverse nature. Therefore, the process planning activity can be broken down into various phases (subactivities) like; operation selection, operation sequencing, tool selection and set-up planning [33]. The number of those subactivities determines degree of divergence and capability of a process planning system.

Two primary interfaces are required in order to integrate a computer aided process planning system with CAD and CAM are;

- a CAD to CAPP interface (feature recogniser) which has the ability to accept part information from CAD systems and to translate CAD language to CAPP language
- a CAPP to CAM interface (CNC code generator) which has the ability to pass the process information to CAM systems

The successful development of an integrated CAD/CAPP/CAM system is therefore dependent on these two interfaces together with capability and performance of the CAPP modules.

The idea behind the development of a process planning system for prismatic parts, problems faced in the development of the system as well as phases of the work are presented in the following paragraphs.

10.2.1. Initiation of the Work

At the beginning, there were a lot of objectives of this work, as they were discussed in Chapter 1. However, one of the main ideas behind this thesis was to establish a step to CAD/CAM integration, and to provide a home-made and domestic process planning package with its two interfaces to the small-sized manufacturing

industry. The initiation of this thesis was also due to the experience gained from a previous work on a process planning system for 2D rotational parts called OPPS-ROT [43] which was developed within a Ph.D. study at the Mechanical Engineering Department in the University of Gaziantep. It was a PC based and commercial package free system developed by using C language. It was developed by using pure conventional programming techniques and could only conceive 2D components modeled on its own CAD tool integrated into the system. Hence, front end of the CAD/CAM integration was closed to other CAD environments. Tool selection and workpiece handling modules of the OPPS-ROT was relatively weak. However, it has been used for reasonably-complex rotational parts with success for two years. OPPS-ROT has found its place as a reference example in the literature.

Motivation of the work was mainly given to the development of a CAD/CAPP/CAM package for prismatic parts that can serve as one of the corner-stones of an intelligent and integrated manufacturing system to be developed further.

In the near future, the system developed in this thesis: OPPS-PRI and the OPPS-ROT could be the two main components of a CAD/CAM package for small-scale manufacturing industry. However, both systems require some modifications.

10.2.2. The Need for This Work

In spite of the fact that numerous papers were published on the process planning and CAD/CAM integration to date, there have not been so many process planning researches for prismatic components in the literature as compared with those for the turning processes. This is mainly due to problems in geometrical representation of the 3D prismatic parts which have often complex shapes and also complex nature of cutting mechanism in milling.

It has been also recognised from the literature that the studies on the process planning are not saturated and some of the modules were less emphasised. Although the most of the emphasis is given to the "*feature recognition*", the problems involved in this module were not completely identified and solved. Little work has been done

on the automated selection of set-ups for the prismatic parts. There is almost no work that attempts to develop a concurrent engineering strategy with the inclusion of DFM utilities into a typical CAD/CAM cycle. For generative process planning systems to be successful, some of the most difficult problems to be solved is the workpiece and set-up selection as well as the detection of feature interactions. Pocket machining and optimisation of turret indexing positions have not been successfully addressed in the process planning. Most of the systems reported in the literature are not modular and not open to any modifications. The utilization of STEP standard which can be used to translate both technological and geometrical information between CAD and CAM is rare. This work in the first place attempts to find practical solutions to those problems encountered with a process planning system.

10.2.3. Specification of the Tools Employed in the Thesis

Machine Tool and Controller Unit

It is obvious that if a work is performed for only theoretical objectives without any actual implementation, the benefits obtained can not be directly reflected to the practical area. So, early definition of the practical side, actual conditions, actual machining environment, available tools etc., are extremely important for the realisation of the goals of a project. Most of the works performed on the CAD/CAM integration including process planning have not taken this situation into account. Many works have been carried out without considering available machine and tools of a factory, and even without a previously determined target. For example, some of the process planning systems developed for machining prismatic parts as well as rotational parts offer a *machine tool selection module* in which it is assumed that the company has a large variety of machine tools of different sizes and powers. They provide a selection mechanism which determines a specific machine tool for each operation by considering estimated or calculated cutting conditions or the dimension of the workpiece, etc., among the alternatives, and prepare a process plan employing the selected machine tools. However, if the company does not have a large variety of machine tools in its portfolio, as in the case of small-scale industries, the machine tool selection facility becomes a redundant process. Many small-scale machining industry

has only one single CNC Vertical Machining Center. A few of them and medium-sized industry may have an extra CNC turning center on the shop floor. The reason for the limited CNC machine tool utilisation in the small-to-medium scale machining industry is that the initial cost of these machines is notably high; changing from 100.000 USD to 500.000 USD.

In this thesis, machine tool selection facility is not included in the process planning cycle. Instead, the user can select and specify a target machine (available on the shop-floor), and the system can produce process plans and part programs for the specified machine tool. The developed process planning system is currently capable of producing part programs for a 3-axis VMC. However, since the system is designed modular, a new CNC milling machine can easily be described into the system.

As everyone knows, a CNC system is composed of three main units; a machine tool, a controller unit, and program of instructions. If a well constructed CNC machine tool can not provide a flexible programming interface (due to poor controller unit), expected CNC advantages can not be realised. Vice versa is also true. The specification of machine controller unit (MCU) *which can be used as the "base" for the format of part programs to be generated* is therefore also an important consideration. However, machine tools are generally equipped with their standard control unit provided by MCU producers in agreement. Therefore, the company or the user has usually no chance to select the type of MCU. The MCU coming with the machine tools is to be used.

Another problem with the MCUs is that CNC industry has so far been unable to come up with a compatible coding standard. The problem is that each machine tool/controller unit is different and machine readable codes (G and M codes) running the equipment are non-standard. Each control uses different codes and functions specific to a model. Therefore, the same G word may represent different functions on different machines. Part programs generated for a specific control type can not be used for another, even if the machine is the same kind and size, because of those incompatible codes. Besides, the number of CNC words supported by the machine tool control determines the capabilities of the machine. For instance, although there

are more than two hundred G words standardised, many CNC machines may not support even a few of them. The problems concerning the non-standard CNC codes can be overcome by specifying the properties of different machine tool controllers into the system and presenting a simple selection mechanism at the beginning of the process planning session.

However, it is well worth pointing out that there was not so many alternative CNC milling machines to be selected as the target machine for this study. TAKSAN TMC 700 was available and it is readily used for testing the resulting part programs.

TAKSAN TMC-700 VMC is fitted with the FANUC O-MC CNC controller unit and offers a wide range of G words, M words, canned cycles and options. The X, Y, Z axes strokes of the TAKSAN VMC are 700, 550, 600 mm, respectively. It has a 12-station ATC. Although VMCs have disadvantages concerning chip disposal, they are less expensive choice, and therefore more popular than HMCs. The current work is focused on the process planning of prismatic parts that can be machined on a 3-axis VMC.

Software and Hardware

The introduction of powerful microprocessors, computers as well as the CNC machine tools into industry has given a new perspective to manufacturing processes. The new systems and concepts like IMS, CIM, CAD, CAPP, CAM, CAI, GT have been considered and employed to improve product quality while decreasing the cost by optimising and integrating the individual tasks of design, planning and manufacturing facilities. However, many of these techniques were not significantly efficient due to inadequacy of the existing computer hardware and software.

It is actually important to select the right working environment to develop a CAD/CAM package as in the case of selecting the right software to perform CAD/CAM applications. A good working environment will help CAD/CAM software developers improve the quality of their works, program jobs more quickly, and handle more complex jobs. The choice of hardware makes a vital difference to the speed of

developments. A more powerful computer is more expensive initially, but it is a better value in the long run. Software will perform better and computer will function for a large period time. A number of factors have to be taken into consideration when selecting a software developing tool. For example, it must be powerful and easy to use.

Other vital elements in the development of a CAD/CAM package is the hardware. Computers and their peripherals are necessary to develop a software. Different software and hardware components were available that run on most PC based computers as well as workstations under UNIX operating systems and a VAX main-frame system to be used for this work.

Recently, the trend in CAD/CAM packages has been toward PCs. They are now beginning to rival the workstations in performance and cost. Many of the popular CAD/D companies have developed their software to run on PCs, too. Therefore, the software developed in this work is also PC based.

In this work, following hardware and software are utilised;

- Pentium 133 processor PC which has 32 MB RAM with its other peripherals
- Silicon Graphics Indy workstation which has 64 MB RAM
- Compilers of C and Visual Basic programming language
- Pro/Designer module of Pro/Engineer Professional Engineering Software
- CNCEZ simulation software
- TAKSAN TMC 700 VMC with RS-232 Interface

Inclusion of AI

The 1990's produced a remarkable growth in the area of Artificial Intelligence (AI) that is accepted as to be one of the most promising key technologies for the factories of the future. In recent years, numerous programs and programming packages have been developed to solve manufacturing problems and many papers

have reported the successful applications of AI to engineering design and manufacturing planning.

Knowledge-based systems also known as expert systems, neural networks, genetic algorithms, fuzzy logic are some of the examples of AI tools. Unfortunately these tools, for instance, their compilers or shells are not easily found. Of course, one can also implement AI strategy or tool for different purposes by writing her/his own computer program, but this is somewhat difficult and a tedious task. For example, writing a neural network algorithm for a complex problem is a difficult process including a lot of trial and error methods for establishing the structure of the network. It is also impossible to write such an algorithm without using an object-oriented programming language.

The expert systems are the most famous AI tools. They are the first attempt to simulate human behaviours and thinking process in industry. In some old books, AI is equivalent to only expert systems. Initially, one of the principal objectives of this work was to develop a fully expert system for process planning. However, the tools for building expert systems are not sufficiently developed and are difficult to apply. Several thousand rules and huge software systems should be contained (since the process planning is very inclusive and divergent) in order to build such a structurally-expert system for process planning, which are difficult to use on conventional computer systems. A partially expert system *which can take out of the responsibility of the experienced manager in the shop floor* is developed in this work. It has been enriched with other AI tools to optimise the ability of some process planning modules.

The basic difference between a conventional program and expert system is actually the way in which knowledge is presented and processed. The expert system consists of a knowledge data base containing explicit knowledge of a human expert in a specialised domain, and an inference engine which can access the knowledge base to come to a decision for a described problem. In principle, any algorithm and its associated data contain expert knowledge about a limited domain [3].

The algorithm and language of programming are the two predominant factors in the design of a software [183]. The computer programming languages have been actually accepted as CIM technologies serving the objective of flexibility [184]. It is obvious that there is a direct relationship between programming language and performance of the programs. Alting and Zhang [105] have made a literature survey on the programming languages for CAPP procedures and seen that, PASCAL, C, LISP, and PROLOG are now being used increasingly in that area. Among them, PASCAL and C have a special importance because of their close relationship with UNIX operation system used commonly for machining centres, PROLOG and LISP occupy more memory space and have low speed. So, they are generally used for the steps requiring a highly logical programming before the CNC code generation in the CAPP procedures. In this study C is selected as the main programming language, since it is a fast and powerful language and offers a handy environment to work, and has superior graphical capabilities for simulation processes [185]. The availability of compilers and manuals of C played also an important role in this selection.

10.3. CONTRIBUTIONS AND CONCLUDING REMARKS

In this study, a Computer Aided Process Planning (CAPP) package for prismatic parts is developed together with its two interfaces (CAD/CAPP interface and CAPP/CAM interface) to integrate CAD/CAM facilities. The package is able to recognise a model and to generate a machining program for producing the model on the machine tool. The methodologies used for complex process planning functions in an integrated CAD/CAPP/CAM environment have been established. The developed package is composed of five main sections;

- Section 1: provides a modelling platform (CAD tool) with a feature recogniser (CAD/CAPP interface) based on STEP (STandard for the Exchange of Product) standard,
- Section 2: includes the process planning works such as “workpiece size selection”, “tool selection”, “cutting parameters selection” and “set-up planning”
- Section 3: involves optimisation of process planning events

- Section 4: offers a utility on “Design for Manufacturing”
- Section 5: is the CAPP/CAM interface that consists of the generation, verification and down-loading of CNC codes.

The main objective of this research was to develop an intelligent and integrated CAD/CAM system for shop-floor use that can be used by an average operator and produce fully (globally) optimised results (process plans and part programmes). For this purpose, in this work, an attempt has been made to include the impact and potential of Artificial Intelligence (AI) in process planning applications and to optimise all events (parameters) in an integrated CAD/CAM environment. Genetic algorithms were extensively used together with conventional programming techniques in the development of process planning and optimisation facilities in order to include profits of AI techniques into the system.

The following functions of CAD/CAM integration are fulfilled by this work;

- a CAD/CAPP interface (feature recogniser) based on STEP standard,
- a CAPP system, including the applications like; “workpiece size selection”, “tool selection”, “cutting parameters selection” and “set-up planning”,
- CAPP/CAM interface, including both the automatic generation and verification of CNC part programs together with tool paths,

Most important significant features of the developed system are as follows;

- extensive use of GAs for optimisation of process parameters
- for automatic tool selection and part program generation for arbitrarily shaped pockets with islands, a modified methodology was proposed and implemented. The developed strategy can also be used for virtual pockets (i.e., for complex slots)
- process planning tasks were individually optimised
- global optimisation of process plans was accomplished
- feature recognition (CAD/CAM interface) was based on STEP standard.
- a “design for manufacturing” utility was provided
- the entire system was implemented on a Vertical Milling Center

- tool selection module is capable of selecting alternative tools for machining operations
- part programs were directly transmitted to the machine tool through RS 232-C serial communication interface

The first generation of process planning systems were stand alone with limited links to CAD and CAM, and has employed a static approach not able to meet advances in manufacturing technology. The developed CAD/CAM package has a modular structure. Therefore, it can further be extended and modified and could be integrated with a previously developed process planning package (OPPS-ROT) designed for rotational parts [43]. Any CAD system capable of generating standard STEP files can serve as the front end (CAD environment) to the developed CAD/CAM package.

The experience gained from this work can easily be applied to other domains of machining, i.e., for process planning of sheet metals. By making small modifications, the developed package can also be implemented for a HMC. The most usable result of the system is a part program to be used for machining of the model created in the CAD stage. The generated part programs have been tested satisfactorily on the TAKSAN TMC-700VMC and positive results have been obtained.

Recent surveys on CAD/CAM systems have revealed an increasing trend for small and medium sized industries to employ CAPP systems for their manufacturing tasks. Lower cost of these systems will probably increase this trend. The results obtained from this study can provide an additional degree of flexibility for the planning and manufacturing resources in machining industry. With the developed package, it would be possible to reduce the times spent in the planning of machining, part programming and verification, and to contribute to the success of the manufacturing industry. This will lead to increased utilisation of CNC machine tools and maximisation of CNC productivity which are commonly considered as ultimate goals of CIM.

10.4. RECOMMENDATIONS FOR FURTHER STUDY

The fast moving world of software and hardware developments makes it nearly impossible to predict trends in the long-term development of integrated CAD/CAM systems. However, the work presented in this thesis can be extended to the following fields;

- future research should take *real time considerations* into account,
- an *adaptive control system* for cutting parameters (feed, speed, depth of cut, etc.) might be implemented. Hence, on-line optimisation of cutting parameters would be possible. This will maximise CNC productivity, extend tool life and minimise the operator supervision,
- *chatter* is announced as one of the most critical parameters to be taken into consideration in mass production. So, the installation of on-line chatter control systems in CAD/CAM packages will increase the product quality considerably,
- quality control is a vital element in CIM. The 21st century is expected to be a “*Century dedicated to Quality*”. Therefore, integration of *Computer Aided Inspection* systems with other CAD/CAM applications has a higher importance,
- *thinking mechanism* of experienced process planners should be investigated in order to reflect their knowledge to process planning more effectively,
- current process planning systems should continuously be enhanced with AI techniques. The advantages and disadvantages of some of the AI techniques will be amalgamated in order to develop an intelligent and powerful CAD/CAM system; a *Neuro-Fuzzy Expert CAD/CAM system*. However, these packages should be used by average mechanical operators,
- further process planning works should include “*Reverse Engineering*”. CNC code generation systems can also be incorporated with the image processing technology. From a scanner or photograph image of the workpiece, the CNC part programs can automatically be generated,
- CAPP should be integrated and furnished with *additional CIM functions* like inspection, production planning, other than CAD and CAM
- inclusion of more *compound/complex features* including non-planar surfaces could enhance the utility of CAD/CAM systems,

- another potential area for research can be developing a process planning system (hopefully AI based) which concentrates on the manufacture of dies considering that the manufacturing environment has, *not only one single target machine* like VMC, but also a grinding machine, an EDM machine, etc.
- one of the ultimate goal of manufacturing and software industry should be *developing a domestic MCU* for CNC machine tools,
- CAPP systems *should be affordable* to the small and medium-scale manufacturing companies,
- tolerancing issues should be considered in the process planning,
- future works should be placed in the concurrent engineering context,
- high speed machining is becoming popular and plays an important role for the reduction of the lead time and production cost in machining [186, 187]. Therefore, further process planning work may consider “*High Speed Machining*” (e.g., in the determination cutting conditions).



LIST OF REFERENCES

1. Bell, R., 1996, "Planetary Speech of the 1st National Symposium on Intelligent Manufacturing Systems", Sapanca, Sakarya, Turkey.
2. Harrington, J., 1973, Computer Integrated Manufacturing, R.E. Kreiger, Huntington, NY, USA.
3. Rembold, N., et.al., 1994, Computer Integrated Manufacturing and Engineering, Addison-Wesley, UK.
4. Gülesin, M. and Jones, R.M., 1993, "Cutting Tool Selection for Process Planning Using Expert System Techniques", 26th International Symposium on Automotive Technology and Automation, pp. 359-366.
5. Kayacan, M.C., 1995, "Computer Aided Process Planning for Rotational Parts", Ph.D. Thesis, University of Gaziantep,.
6. Dereli, T., 1994, "Computer Aided Part Program Generation and Simulation of Turning Processes", M.Sc. Thesis, University of Gaziantep, June,.
7. Wolfe, P.M., 1985, "Computer-Aided Process Planning Is Link Between CAD and CAM", Industrial Engineering, August, pp. 72-77.
8. Ham, I. and Lu, S.C.Y, 1988, "Computer-Aided Process Planning: The Present and the Future", Annals of the CIRP, Vol. 37, No. 2, pp. 591-601.
9. ———, 1994, Modern Metal Cutting - A Practical Handbook, Sanvik Coromant AB, Sweden.

10. Fallböhmer, P., et.al., 1995, "Survey of the Die and Mold Manufacturing Industry-Practices in Germany, Japan, and US", Journal of Materials Processing Technology, Vol. 59, pp. 158-168.
11. Altan, T., et.al, 1993, "Advanced Techniques for Die and Mold Manufacturing", Annals of the CIRP, Vol.42, No.2, pp. 707-716.
12. Murray, J.L. and Yue, Y., 1993, "Automatic Machining of 2.5D Components with the ACIS Modeller", International Journal of Computer Integrated Manufacturing, Vol. 6, No. 1-2, pp. 94-104.
13. Grayer, A.R., 1976, "A Computer Link Between Design And Manufacture", Ph.D. Dissertation, University Of Cambridge, UK.
14. Shah, J., et.al., 1991, "Survey Of Feature Based Process Planning And NC Programming Techniques", Computer-Aided Engineering Journal, pp. 25-33.
15. Kakino, Y., et al., 1977, "A New Method of Part Description for Computer Aided Production Planning", Advances in Computer Manufacturing, pp.197-213.
16. Kyprianou, L.K., 1980, "Shape Classification in Computer Aided Design", Ph.D. Dissertation, University of Cambridge, UK.
17. Jakubowski, R.R., 1982, "Syntactic Characterisation of Mechanical Part Shapes", Cybernetics, Vol.13, pp.1-24.
18. Staley, S.M., et.al., 1983, "Using Syntactic Pattern Recognition to Extract Feature Information From Solid Geometric Data Base", Computers in Mechanical Engineering, Vol. 2, pp. 61-66.
19. Li, R.K. and Bedworth, D.D., 1988, "A Framework for the Integration of Computer Aided Design and Computer Aided Process Planning", Computers and Industrial Engineering, Vol.14, No.4, pp.395, 413.

20. Srinivasan, R. and Liu, C.R, 1988, "Extraction of Manufacturing Details From Geometric Models", Computers and Industrial Engineering, Vol. 9, pp.125-133.
21. Filiz, İ.H., et. al. (including Dereli, T.), 1994, "ROMOD: A Product Modelling System for Rotational Parts", 6th International Machine Design and Production Conference, pp.271-279, September 21-23, METU, Ankara.
22. Woo, T.C., 1984, "Feature Extraction by Volume Decomposition", Proceedings of the Conference on CAD/CAM in Mechanical Engineering, Cambridge, pp. 76-94.
23. Woo, T.C., 1984, "Interfacing Solid Modelling to CAD and CAM: Data Structures and Algorithms for Decomposing A Solid", IEEE Computer, Vol. 17, pp. 44-49.
24. Joshi, S. and Chang, T.C., 1988, "Graph Based Heuristics for Recognition of Machined Features from 3D Solid Model", Computer-Aided Design, Vol. 20, pp. 58-66.
25. Gossard, D. and Sakurai, H., 1990, "Recognizing Shape Features in Solid Models", IEEE Computers Graphics and Applications, Vol. 10, No. 5, pp. 22-32.
26. De Floriani, L., 1987, "Graph Based Approach to Object Feature Recognition", Proceeding of the 3rd ACM Symposium on Computational Geometry, Canada.
27. Gindy, N.N.Z., et.al., 1993, "Feature-based Component Model for Computer Aided Process Planning Systems", International Journal of Computer Integrated Manufacturing, Vol. 6, No. 1-2, pp. 20-26.
28. Henderson, M.R. and Musti S., 1984, "Automated Group Technology Part Coding from A Three-Dimensional CAD Database", Transactions of the ASME, Vol. 110, pp. 278-28.

29. Henderson, M.R. and Anderson, D.C., 1984, "Computer Recognition and Extraction of form Features: A CAD/CAM Link", Computers in Industry, Vol. 4, No. 5, pp. 315-325.
30. Prabhakar, S. and Henderson, M.R., 1994, "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.
31. Chen, C.L.P. and LeClair, S.R., 1994, "Integration of Design and Manufacturing: Solving Setup Generation and Feature Sequencing Using an Unsupervised Learning Approach", Computer-Aided Design, Vol. 26, No. 1, pp. 59-75.
32. Maropoulos, P.G., 1995, "Review of Research in Tooling Technology, Process Modelling and Process Planning, Part II: Process Planning", Computer Integrated Manufacturing Systems, Vol.8, No.1, pp.13-20.
33. Kusiak, A., 1991, "Process Planning: A Knowledge-Based and Optimization Perspective", IEEE Transactions on Robotics and Automation, Vol. 7, No. 3, pp. 257-266.
34. —————, 1990, CAM-I Proceedings: Features Symposium, P-90-PM-02.
35. Shah, J., et.al., 1994, "Introduction to Feature Based Manufacturing", Advances in Feature Based Manufacturing (Eds. Shah, J., et al.), Elsevier, The Netherlands, pp. 1-11.
36. Mayer, R.J., et al., 1992, "An Integrated Manufacturing Planning Assistant - IMPA", Journal of Intelligent Manufacturing, Vol. 3, No. 2, pp. 109-122.
37. Yue, Y., 1993, "Process Planning for 2.5D Components", Ph.D. Thesis, Heriot-Watt University, UK.
38. Gülesin, M. and Jones, R.M., 1993, "Stock Selection and Blank Part Modelling in An Expert Process Planning System", Proceedings of 6th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, pp. 14-17.

39. Champati, S., et al., 1996, "Automated Operation Sequencing in Intelligent Process Planning: A Case-Based Reasoning Approach", International Journal of Advanced Manufacturing Technology, Vol. 12, pp. 21-36.
40. Chang, T.C., 1985, An Introduction to Automated Process Planning System, Prentice-Hall Inc., Englewood Cliffs, New Jersey, USA.
41. Hinduja, S. and Huang, H., 1989, "OP-PLAN: An Automated Operation Planning System for Turning Components", Proceedings of the Institution of Mechanical Engineers, Vol. 203, pp. 145-158.
42. Lewis, W. and Bartlef, E., 1982, Tool Oriented Automatic Process Planning", Proceedings of the 23th International MATADOR Conference, Manchester, UK.
43. Kayacan, M.C., et.al. (including Dereli, T.), 1996, "OPPS-ROT: An Optimised Process Planning System For Rotational Parts", Computers in Industry, Vol. 32, pp. 181-195.
44. Eskicioğlu, H. and Davies, B.J., 1981, "An Interactive Process Planning System for Prismatic Parts (ICAPP)", International Journal of Machine Tools and Design Research, Vol. 21, No. 3/4, pp. 193-206.
45. Joshi, S., et.al., 1988, "Expert Process Planning System with Solid Model Interface", International Journal of Production Research, Vol. 26, No. 5, pp. 863-885.
46. Gülesin, M. and Jones, R.M., 1993, "Operation Sequencing in an Expert Process Planning and Fixturing System", Proceedings of the ICES'93: The International Conference on Engineering Software, pp.45-50.
47. Filiz, İ.H., et.al. (including Dereli, T.), 1996, "Computer Aided Determination Of Sequence Of Operations For Prismatic Parts", Proceedings of 7th International Conference on Machine Design and Production, METU, 11-13 September, pp.103-112.

48. Ssemakula, M.E., et.al, 1988, "Optimal Process Sequencing in CAPP Systems", Proceedings of the Autofact' 88 Conference, Vol. 22, pp. 15-27.
49. Rho, M., et.al, 1992, "An Integrated Cutting Tool Selection and Operation Sequencing Method", Annals of the CIRP, Vol.41, No.1, pp.517-520.
50. Lin, C.J. and Wang, H.P, 1993, "Optimal Operation Planning and Sequencing: Minimization of Tool Changeovers", International Journal of Production Research, Vol. 31, No. 2, 1993, pp. 257-266.
51. Koulamas, C., 1993, "Operation Sequencing and Machining Economics", International Journal of Production Research, Vol. 31, No. 4, pp. 957-975.
52. Vosniakos, L.C. and Davies, B.J, 1993, "Knowledge-Based Selection and Sequencing of Hole-Making Operations for Prismatic Parts", International Journal of Advanced Manufacturing Technology, Springer-Verlag Ltd., Vol. 8, pp. 9-16.
53. Kamhavi, H.N., et. al., 1996, "Feature Sequencing in the Rapid Design System using a Genetic Algorithm", Journal of Intelligent Manufacturing, Vol. 7, pp. 55-56.
54. Maropoulos, P.G. and Hinduja, S.,1991. "Automatic Tool Selection for Rough Turning", International Journal of Production Research, Vol. 29, No. 6, pp 1185-1204.
55. Hinduja, S. and Barrow, G., 1993, "SITS - A Semi-intelligent Tool Selection System for Turned Components", Annals of the CIRP, Vol. 42, No.1.
56. Domazet, D., 1990, "The Automatic Tool Selection with The Production Rule Matrix Method", Annals of the CIRP, Vol.. 39, No. 1, pp. 497-500.
57. Giusti F, Santochi M, Dini G, 1986, "COATS: An Expert Module for Optimal Tool Selection", Annals of the CIRP, Vol. 35, No. 1, pp. 337-340.

58. Hayes, C. and Wright, P., 1988, "Automating Process Planning: Using Feature interactions to guide search", Journal of Manufacturing Systems, Vol. 8, No.1, pp. 1-14.
59. Chang, T.C., 1990, Expert Process Planning for Manufacturing, Addison-Wesley, Reading, MA, 1990.
60. Sakurai, H., 1992, "Automatic Setup Planning and Fixture Design for Machining", Journal of Manufacturing Systems, Vol. 11, No. 1, pp. 30-37.
61. Zhang, Y.F., et.al., 1995, "A Hybrid Approach for Set-up Planning", International Journal of Advanced Manufacturing Technology, Vol.10, pp. 183-190.
62. Boerma, J.R. and Kals, H.J.J., 1988, "FIXES: A System for Automatic Selection of Set-ups and Design of Fixtures", Annals of the CIRP, Vol. 37, No. 1, pp. 443-446.
63. Wiendahl, E.H. and Fu, Z., 1992, "Computer-Aided Analysis and Planning of Set-up Process", Annals of the CIRP, Vol. 41, No. 1, pp. 497-500.
64. Delbressine, F.L.M., et.al., 1993, "On the Automatic Generation of Set-ups Given a Feature-Based Design Representation", Annals of the CIRP, Vol. 42, No.1, pp. 527-530.
65. Yue, Y. and Murray, J.L., 1994, "Validation, Workpiece Selection and Clamping of Complex 2.5D Components", Advances in Feature Based Manufacturing (Eds. Shah, J., et al.), Elsevier, The Netherlands, pp. 185-213.
66. Manessa, R.J. and DeVries, W.R., 1989, "Locating Point Synthesis in Fixture Design", Annals of the CIRP, Vol. 38, No. 1, pp. 165-169.
67. Roy, U. and Sun, P.L., 1994, "Selection of Preliminary Locating and Clamping Positions on a Workpiece for an Automatic Fixture Design System", Computer Aided Manufacturing Systems, Vol. 7, No. 3, pp. 161-172.

68. Young, R.I.M. and Bell, R., 1991, "Fixturing Strategies and Geometric Queries in Set-up Planning", International Journal of Production Research, Vol. 29, No.3, pp 537-550.
69. Kals, H.J.J., et.al, 1978, "A Computer Aid in the Optimization of Turning Conditions in Multi-Cut Operations", Annals of the CIRP, Vol. 27, No.1, pp. 465-469.
70. Crookall, J.R. and Venkataramani,N., 1971, "Computer Optimisation of Multipass Turning", International Journal of Production Research, Vol. 9, No. 2, pp. 247-259.
71. Lambert, P.K. and Walvekar, A.G., 1978, "Optimisation of Multipass Machining Operations", International Journal of Production Research.
72. Hitomi, K., 1971, "Optimisation of Multistage Machining System: Analysis of Optimal Machining Conditions for the Flow-Type Machining System", Transactions of the ASME, pp. 498-506.
73. Hitomi, K., 1976, "Analysis of Production Models - Part 2: Optimisation of a Multistage Machining System", AIIE Transactions, Vol. 8, No. 1, pp. 106-112.
74. Hitomi, K., 1977, "Optimisation of Multistage Production Systems with Variable Production Times and Costs", International Journal of Production Research, Vol. 15, No. 6, pp. 583-597.
75. Hitomi, K., 1979, Manufacturing Systems Engineering, Taylor & Francis, London, pp. 184-193.
76. Rao, S.S and Hati, S.K., 1978, "Computerized Selection of Optimum Machining Conditions for Flow-Type Multistage Machining System", Journal of Engineering for Industry, Vol. 100, pp. 356-362.
77. Iwata, K., et.al., 1972, "A Probabilistic Approach to the Determination of the Optimum Cutting Conditions", Journal of Engineering for Industry, Vol. 94.

78. Somlo, J. and Nagy, J., 1977, "A New Approach to Cutting Data Optimization", Advances in Computer Aided Manufacture, North-Holland.
79. Groover, M.P., 1975, "Monte Carlo Simulation of the Machining Economics Problem", Transactions of the ASME, Vol. 97, pp. 931-938.
80. El Hakim, M.A., et. al., 1983, "Computer Program for the Optimization of the Machining Variables", Computers and Industrial Engineering, Vol. 7, pp. 317-336.
81. Brewer, R.C., 1978, "On the Economics of Basic Turning Operations", Transactions of the ASME, Vol. 80, pp. 1498-1958.
82. Van Houten, F.J.A.M., et. al., 1981, "The Development of a Technological Processor as a Workpiece Programming System", Annals of the CIRP, Vol. 30, pp. 363-368.
83. Devor, R.E, et.al., 1983, "Control of Surface Error in End Milling", Proceedings of NAMRC, pp. 356.
84. Ermer, S. and Kromodihardjo, S., 1981, "Optimisation of Multipass Turning with Constraints", Transactions of the ASME, 80-WA/Prod-22.
85. Tipnis, V.A. and Friedman, M.Y., 1976, "Cutting Rate-Tool Life Characteristic Functions for Metarial Removal Processes, Part 1:Theory", Transactions of the ASME, Vol. 75, pp. 481-486.
86. Tipnis, V.A. and Friedman, M.Y., 1976, "Cutting Rate - Tool Life Characteristic Functions for Metarial Removal Processes, Part 2: Verification & Applications", Transactions of the ASME, Vol. 75, pp. 487-495.
87. Hinduja, S., et.al., 1985, "Calculation of Optimum Cutting Conditions for Turning Operations", Proceedings of the Institution of Mechanical Engineers, Vol. 199, No. 82, pp. 81-92.

88. Arsecularatne, J.A., et.al., 1992, "Optimum Cutting Conditions for Turned Components", Proceedings of the Institution of Mechanical Engineers, Vol. 206, pp. 15-30.
89. Agapiou, J.S., 1992, "The Optimisation of Machining Operations Based on a Combined Criterion, Part 2: Multipass Operations", Journal of Engineering for Industry, Vol. 114, pp. 508-513.
90. Wang, J., 1993, "Constrained Optimisation of Rough Peripheral and End Milling Operations", Ph.D. Thesis, University of Melbourne.
91. Wang, H.P. and Wysk, R.A., 1987, "Intelligent Reasoning for Process Planning", Computers in Industry, No.8, pp. 293-309.
92. Joseph, A.T., et.al, 1990, "Automatic Generation of NC Programs for Turned Components from CAD Product Models", Proceedings of the 28th International MATADOR Conference, pp 289-294.
93. Wang, H.P. and Lin, C.A., 1987, "Automated Generation of NC Part Programs for Turned Parts Based on 2-D Drawing Files", International Journal of Advanced Manufacturing Technology, Vol. 2, No. 3, pp. 23-36.
94. Chin, S.C., 1988, "Developing A Feature Based Knowledge System For CAD/CAM Integration", Computers and Industrial Engineering, Vol. 15, No. 1-4, pp 34-40.
95. Ssemakula, M.E., 1990, "Process Planning System in the CIM Environment", Computers and Industrial Engineering, Vol. 19, No. 1-4, pp. 452-456.
96. Shyu, J. and Chen, Y.W., 1987, "A Mini CIM System for Turning", Annals of the CIRP, Vol. 36, No. 1, pp. 277-280.
97. Boer, C.R., et.al., 1991, "CAPP/CAM Expert System for a High Productivity, High Flexibility CNC Turning Center", Annals of the CIRP, Vol. 39, No. 1, pp. 481-483.

98. Opas, J. and Mantyla, M., 1994, "Techniques for Automatic Part Program Generation", Advances in Engineering Software, Vol. 20, No. 2-3, pp.141-155.
99. Filiz, İ.H., et.al.(including Dereli, T.), 1994, "Automatic Generation of CNC Part Programs for Rotational Parts", Proceedings of 6th International Conference on Machine Design and Production, pp. 281-290.
100. Filiz, İ.H. and Dereli, T., 1996, "Tornalama İşlemleri İçin Otomatik Parça Programı Türetimi Metodları, 'Gerekçeleme' ve Bir Uygulama", Proceedings of 1st Symposium on Intelligent Manufacturing Systems, pp.489-505.
101. Niebel, B.W., 1965, "Mechanical Process Selection for Planning New Designs", ASME Paper, No. 737.
102. Schenk, D.E., 1966, "Feasibility of Automated Process Planning", Ph.D. Thesis, Purdue University, West Lafayette, Indiana, USA.
103. Dong, J., et al, 1995, "Intelligent Feature Extraction for Concurrent Design and Manufacturing", pp. 301-323, Design and Implementation of Intelligent Manufacturing Systems, From Experts Systems, Neural Networks, to Fuzzy Logic (Editors: H.R.Parsaei and M. Jamshidi), Prentice Hall, USA.
104. Nolen, J., 1989, Computer Automated Process Planning for World Class Manufacturing, Marcel Dekker Inc.
105. Altıng L. and Zhang, H., 1989, "Computer Aided Process Planning: State of the Art Survey", International Journal of Production Research, Vol. 27, No. 4, pp. 553-585.
106. Eversheim, W. and Schultz, J., 1985, "CIRP Technical Reports: Survey of Computer Aided Process Planning Systems", Annals of the CIRP, Vol. 34, No. 2, pp. 607-613.
107. Gupta, T., 1990, "An Expert System Approach in Process Planning: Current Development and Its Future", Computers and Industrial Engineering, Vol. 18, No. 1, pp. 69-80.

108. Link, C.H., 1976, "CAPP - CAM-I Automated Process Planning System", Proceedings of the 13th NC Society Annual Meeting and Technical Conference, CAM-I, Arlington, pp. 401-408.
109. Schaffer, G., 1980, "GT via Automated Process Planning", American Machinist, Vol. 124, No. 5, pp. 119-122.
110. OIR, MULTIPLAN, 1983, Organisation for Industrial Research, Waltham MA.
111. Tulkoff, J., 1981, "Lockheed's GENPLAN", Proceedings of the 18th NC Society Annual Meeting and Technical Conference, pp. 417-421.
112. Eversheim, W. and Holz, B., 1982, "Computer Aided Programming of NC-Machine Tools by Using the System AUTAP-NC", Annals of the CIRP, Vol. 31, No. 1.
113. Parks, C.M., et al, 1989, "Computer Aided Process Planning with CMPP", Computers and Industrial Engineering, Vol. 17, No. 1-4, pp. 246-251.
114. Descotte, Y. and Latombe, J.C., 1981, "GARI: A Problem Solver That Plans How to Machine Mechanical Parts", Proceedings of the 7th Int. Joint Conference on Artificial Intelligence, Canada.
115. Nau, D.S. and Chang, T.C., 1985, "A Knowledge Based Approach to Process Planning", Proceedings of the Symposium on Computer Aided/Intelligent Process Planning, pp. 65-72.
116. Davies, B.J and Darbyshire, I.L, 1984, "The Use of Expert Systems In Process Planning", Annals of the CIRP, Vol. 33, No. 1.
117. Matsushima, K., et al., 1985, "The Integration of CAD and CAM by Application of Artificial Intelligence Techniques", Annals of the CIRP, Vol. 34, No. 1, pp. 329-332.
118. Lenau, T. and Alting, L., 1986, "XPLAN-An Expert Process Planning System", The 2nd International Expert Systems Conference, London, UK.

119. Iwata, K. and Fukuda, Y., 1987, "KAPPS: Know-how and Knowledge Assisted Production Planning System in the Machining Shop", 19th CIRP International Seminar on Manufacturing Systems, USA.
120. Feigenbaum, E.A., 1977, "The Art of Artificial Intelligence: I, Themes and Case Studies of Knowledge Engineering", Proceedings of 5th International Joint Conference on Artificial Intelligence, pp. 1014-1029.
121. ———, 1990, KES PS Manual, Software Architecture and Engineering, Inc.
122. Huang, S.H. and Zhang, H.C., 1995, "Neural-Expert Hybrid Approach for Intelligent Manufacturing: A Survey", Computers in Industry, Vol. 26, pp. 107-126.
123. Badiru, A.B., 1992, Expert Systems Applications in Engineering and Manufacturing, Prentice-Hall, NJ, USA.
124. Zupan, J. and Gasteiger, J., 1993, Neural Networks for Chemists, VCH Publishers, Germany.
125. ———, 1992, NeuroShell2 Software On-line Help Files.
126. Lipmann, R., 1987, "An Introduction to Computing with Neural Nets", IEEE Acoust. Speech Signal Process (ASSP) Magazine, Vol. 4, No. 2, pp. 4-22.
127. Kalpakjian, S., 1997, Manufacturing Processes for Engineering Materials, 3rd Ed., Addison-Wesley, USA.
128. Von Altrock, C., 1995, Fuzzy Logic & NeuroFuzzy Applications Explained, Prentice Hall, USA.
129. Mizomoto, M., et al., 1979, "Some Methods of Fuzzy Reasoning", Advances in Fuzzy Set Theory and Applications, (Ed. Gupta, M.M. et al.), pp. 117-136.
130. Zadeh, L.A., 1965, "Fuzzy Sets", Information and Control, Vol. 8, pp. 338-353.
131. Ross, J., 1995, Fuzzy Logic with Engineering Applications, McGraw-Hill, USA.

132. Zhang, H.C. and Huang, S.H., 1994, "A Fuzzy Logic Approach to Process Plan Selection", International Journal of Production Research, Vol. 32, No. 6, pp. 1265-1279.
133. Goldberg, E.D., 1989, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, USA.
134. Lee, C.Y. and Choi, J.Y., 1995, "A Genetic Algorithm for Job Sequencing Problems with Distinct Due Dates and General Early-Tardy Penalty Weights", Computers Operations Research, Vol. 22, No. 8, pp. 857-869.
135. Stacey, D., 1994, "Intelligent Systems Architecture: Design Techniques", Artificial Neural Networks for Intelligent Manufacturing (Edited by Dagli, C.H.), Chapman&Hall, UK, pp. 17-38.
136. Filiz, İ.H. and Dereli, T., 1997, Introduction to Computer Aided Design/Drafting and AutoCAD, Univesity of Gaziantep, Printing Office, Gaziantep, Turkey.
137. Gomes, A.J.P. and Teixeira, J.C.G., 1991, "Form Feature Modelling in a Hybrid CSG/BREP Scheme", Computers & Graphics, Vol. 15, No. 2, pp. 217-229.
138. Allada, V. and Anand, S., 1995, "Feature-Based Modeling Approaches for Integrated Manufacturing: State-of-the-art Survey and Future Directions", International Journal of Computer Integrated Manufacturing, Vol. 8, No. 6, 411-440.
139. —————, 1997, Pro/Engineer Guide, Release 18.0, Catalogue.
140. Hannam, R., 1997, Computer Integrated Manufacturing: From Concepts to Realisation, Addison-Wesley Longman, UK.
141. Dereli, T. and Filiz, İ.H., 1997, "Bilgisayar Destekli Tasarım ve Üretim Sistemlerinde Bilgi Değişimi ve Standartları", Mühendis ve Makina, Vol. 38, No. 452, pp. 11-19.
142. McMahon, C. and Browne, J., 1995, CADCAM: From Principles to Practice, Addison-Wesley, UK.

143. Gülesin, M., 1997, "The STEP Standard and Data Exchange Between CAD/CAM Systems", Proceedings of the 1st National Conference on Machine Design Theory and Modern Manufacturing Methods (MATİK), Gazi University, pp. 281-294.
144. —————, 1994, "Industrial Automation Systems and Integration - Product Data Representation and Exchange", International Standard, ISO 10303-21.
145. —————, 1997, Pro/Interface Guide, Pro/Engineer, Release 18.0, Catalogue.
146. —————, 1993, Webster's Ninth Collegiate Dictionary, Merriam-Webster, USA.
147. Filiz, İ.H., et.al. (including Dereli, T.), 1996, "FRS-PP: A Feature Recognition System for Prismatic Parts", Proceedings of 7th International Conference on Machine Design and Production, METU, 11-13 September, pp.113-122.
148. Yıldız, M., 1998, Development of An Automatic Feature Recognition System for Prismatic Parts, M.Sc. Thesis, University of Gaziantep.
149. Venuvinod, P.K. and Yuen, C.F., 1994, "Efficient Automated Geometric Feature Recognition through Feature Coding", Annals of the CIRP, Vol. 43, No. 1, pp. 413-416.
150. Kusiak, A., 1990, Intelligent Manufacturing Systems, Prentice-Hall, USA.
151. Furth, B.L., 1988, "Automated Process Planning", NATO Advanced Study Series - Computer Aided Manufacturing, Vol. 49, pp. 37-53.
152. —————, 1994, Catalogue, 4001 GB Milling, Hertel AG., Germany.
153. ———, 1980, Machining Data Handbook, Machinability Data Center, Metcut Inc.
154. —————, 1993, Milling - Main Catalogue, Seco Milling Cutters and Inserts, Seco Tools AB, Sweden.
155. —————, 1994, Catalogue, 400 GB Milling Technique, Hertel AG., Germany.

156. —————, 1995, Catalogue, Takımsaş Kesici Takım ve Makina Sanayi Tic. A.Ş., İstanbul, Turkey.
157. —————, 1997, Catalogue, Eroğlu Makina Sanayi Tic. A.Ş., Bursa, Turkey.
158. Filiz, İ.H., 1995, NC and CNC Fundamentals, Univesity of Gaziantep, Printing Office, Gaziantep, Turkey.
159. ElMaraghy, H.A., “Evolution and Feature Perspectives of CAPP”, Annals of the CIRP, Vol. 42, No. 2, 1993, pp. 739-753.
160. Sohlenius, G., 1992, “Concurrent Engineering”, Annals of the CIRP, Vol. 41, No. 2, pp. 645-655.
161. Kusiak, A., 1993, Concurrent Engineering: Automation, Tools, and Techniques, Wiley Interscience, USA.
162. Dereli, T., et.al, 1997, “Automatic Determination of Critical Regions Between Interacting Features on Prismatic Components”, Proceedings of the 13th ISPE/IEE International Conference on CAD/CAM, ROBOTICS & Factories of the Future, Vol. 1, pp. 396-401, Colombia.
163. Dere, M., et.al.(including Dereli, T.), 1997, “A Feature-Based ‘Design For Manufacturing System’ To 3-axis CNC Cavity Machining”, Proceedings of the 22nd International Conference on Computers and Industrial Engineering, pp. 671-674, Egypt.
164. Beer, F.P. and Johnston, Jr., E.R., 1992, Mechanics of Materials, McGraw-Hill, Singapore.
165. Dereli, T., et.al, 1997, “Determining Optimal Sequence of Machining Operations Using Conflicting Processing Criteria and A Genetic Algorithm”, Proceedings of the 13th ISPE/IEE International Conference on CAD/CAM, ROBOTICS & Factories of the Future, Vol. 1, pp. 319-326, Colombia.
166. Little, J.D.C, et.al., 1963, “An Algorithm for the Travelling Salesman Problem”, Operations Research, pp. 972-989.

167. Lin, S. and Kernighan, B.W., 1973, "An Effective Heuristic Algorithm for the Travelling Salesman Problem", Operations Research, pp. 498-516.
168. Norback, J.P., and Love, R.F., 1977, "Geometric Approaches to Solving the Travelling Salesman Problem", Management Science, pp. 1208-1223.
169. Dereli, T., et.al., 1997, "Selection Of Alternative Cutting Tools For Part Features Of Prismatic Components And Their Effects On Operation Sequence", Proceedings of the 5th International Conference: Factory 2000, UK, pp. 170-177.
170. Dereli, T., et.al., 1998, "Determination of Optimal Turret-Index Positions of Cutting Tools by Using Genetic Algorithms", paper to be published in Proceedings of the 2nd International Symposium on Intelligent Manufacturing Systems, Turkey.
171. ———, 1996, Manual, HITEC Turning Center, Hitachi-Seiki Co., Ltd., Japan.
172. ———, 1993, Manual, Compact Universal CNC Lathe Hi-Eco10, Hwacheon, Taiwan, R.O.C.
173. ———, 1995, Manual, TAKSAN TMC-700 VMC, Kayseri, Turkey.
174. Ostafiev, V. and Globa, A., 1984, "Integrated End Milling Development", Annals of the CIRP, Vol. 33, pp. 29.
175. Sönmez, A.İ., et.al. (including Dereli, T.), "Dynamic Optimization of Multipass Milling Operations via Geometric Programming", paper to be published in International Journal of Machine Tools and Manufacture.
176. Rad-Tolouei, M. and Bidhendi, I.M., 1997, "On The Optimization of Machining Parameters for Milling Operations", International Journal of Machine Tools and Manufacture, Vol. 37, No. 1, pp. 1-16.
177. ———, 1995, Operator's Manual for Machining Center, Fanuc Series O-MC, Fanuc Ltd., Japan.
178. ———, 1997, Manual, Torcam Simulation Software, USA.

179. Dereli, T., et.al., 1997, “ Adalar’ İçeren Karmaşık Çevreli Cepler İçin Otomatik Takım Seçimi ve Takım Yolu Türetimi”, Proceedings of the 1st National Conference on Mechanical Engineering (MAMKON), İTÜ, pp.284-291.
180. Kurthacioğlu, M., 1995, “CNC Teknolojisi Makro Tekniği ile Cep (Paket) Frezeleme, Makina & Metal Teknolojisi, Vol. 3, No. 9, pp. 142-144.
181. Sava, M. and Pusztai, J., 1990, Computer Numerical Control Programming, Prentice Hall, USA.
182. Kwei-Li, R. and Bedworth, D.D., 1988, “A Framework for the Integration of Computer Aided Design and Computer Aided Process Planning”, Computers and Industrial Engineering, Vol. 14, No. 4, pp. 395-413.
183. Filiz, İ.H., et.al. (including Dereli, T.), 1993, “CNC Tornalama İşlemleri İçin C++ ile Hazırlanan Bir Simülasyon Yazılımı”, Proceedings of the 1st International Symposium on Computer Applications, pp. 63-70, Konya.
184. Kochan, A. and Cowan, D., 1986, Implementing CIM, IFS Publications, UK.
185. Stevens, R., 1992, The C Graphics Handbook, Academic Press Inc., USA.
186. Nakagawa, T., 1997, “High Speed Milling for Die and Mold Making”, Proceedings of International Conference and Exhibition on Design and Production of Dies and Molds, CIRP, İstanbul, pp. 9-14.
187. Ahmetoğlu, M., et.al., 1997, “Modern Die Design and Manufacturing - Application of Process Modeling in Stamping and Machining”, Proceedings of International Conference and Exhibition on Design and Production of Dies and Molds, CIRP, İstanbul, pp. 1-7.

APPENDICES



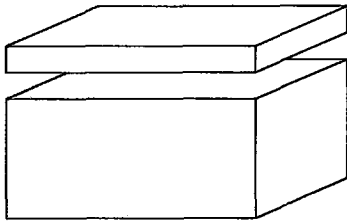
APPENDIX A

WHOLE B-REP FILE FOR THE PART AS SHOWN IN FIGURE 4.12

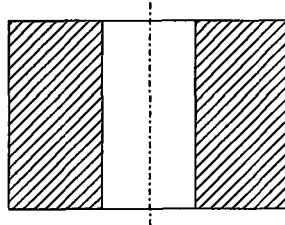
FaceNo	EdgeNo	v1x	v1y	v1z	v2x	v2y	v2z
1	0	105	10	50	15	50	-1
2	1	0	0	0	0	50	0
2	2	0	50	0	150	50	0
2	3	150	50	0	150	0	0
2	4	150	0	0	0	0	0
3	1	0	0	0	0	0	60
3	2	0	0	60	30	0	60
3	3	30	0	60	30	0	45
3	4	30	0	45	50	0	45
3	5	50	0	45	50	0	60
3	6	50	0	60	80	0	60
3	7	80	0	60	80	0	50
3	8	80	0	50	150	0	50
3	9	150	0	50	150	0	30
3	10	150	0	30	140	0	30
3	11	140	0	30	140	0	20
3	12	140	0	20	150	0	20
3	13	150	0	20	150	0	0
3	14	150	0	0	0	0	0
4	1	0	0	0	0	0	60
4	2	0	0	60	0	50	60
4	3	0	50	60	0	50	0
4	4	0	50	0	0	0	0
5	1	0	50	0	0	50	60
5	2	0	50	60	30	50	60
5	3	30	50	60	30	50	45
5	4	30	50	45	50	50	45
5	5	50	50	45	50	50	60
5	6	50	50	60	150	50	60
5	7	150	50	60	150	50	30
5	8	150	50	30	140	50	30
5	9	140	50	30	140	50	20
5	10	140	50	20	150	50	20
5	11	150	50	20	150	50	0
5	12	150	50	0	0	50	0
6	1	150	0	30	150	0	50
6	2	150	0	50	150	20	50
6	3	150	20	50	150	20	60
6	4	150	20	60	150	50	60
6	5	150	50	60	150	50	30
6	6	150	50	30	150	0	30
7	1	140	0	30	140	50	30
7	2	140	50	30	150	50	30
7	3	150	50	30	150	0	30
7	4	150	0	30	140	0	30
8	1	140	0	20	140	0	30
8	2	140	0	30	140	50	30
8	3	140	50	30	140	50	20
8	4	140	50	20	140	0	20
9	1	140	0	20	140	50	20
9	2	140	50	20	150	50	20
9	3	150	50	20	150	0	20
9	4	150	0	20	140	0	20
10	1	150	0	0	150	0	20
10	2	150	0	20	150	50	20
10	3	150	50	20	150	50	0

10	4	150	50	0	150	0	0
11	1	50	0	60	50	50	60
11	2	50	50	60	150	50	60
11	3	150	50	60	150	20	60
11	4	150	20	60	80	20	60
11	5	80	20	60	80	0	60
11	6	80	0	60	50	0	60
11	7	40	30	60	40	40	60
11	8	40	40	60	100	40	60
11	9	100	40	60	100	30	60
11	10	100	30	60	40	30	60
12	1	40	30	53	40	40	53
12	2	40	40	53	100	40	53
12	3	100	40	53	100	30	53
12	4	100	30	53	40	30	53
13	1	40	40	53	40	40	60
13	2	40	40	60	100	40	60
13	3	100	40	60	100	40	53
13	4	100	40	53	40	40	53
14	1	40	30	53	40	30	60
14	2	40	30	60	40	40	60
14	3	40	40	60	40	40	53
14	4	40	40	53	40	30	53
15	1	100	30	53	100	30	60
15	2	100	30	60	100	40	60
15	3	100	40	60	100	40	53
15	4	100	40	53	100	30	53
16	1	40	30	53	40	30	60
16	2	40	30	60	100	30	60
16	3	100	30	60	100	30	53
16	4	100	30	53	40	30	53
17	1	80	20	50	80	20	60
17	2	80	20	60	150	20	60
17	3	150	20	60	150	20	50
17	4	150	20	50	80	20	50
18	1	80	0	50	80	0	60
18	2	80	0	60	80	20	60
18	3	80	20	60	80	20	50
18	4	80	20	50	80	0	50
19	1	80	0	50	80	20	50
19	2	80	20	50	150	20	50
19	3	150	20	50	150	0	50
19	4	150	0	50	80	0	50
20	1	50	0	45	50	0	60
20	2	50	0	60	50	50	60
20	3	50	50	60	50	50	45
20	4	50	50	45	50	0	45
21	1	30	0	45	30	50	45
21	2	30	50	45	50	50	45
21	3	50	50	45	50	0	45
21	4	50	0	45	30	0	45
22	1	30	0	45	30	0	60
22	2	30	0	60	30	50	60
22	3	30	50	60	30	50	45
22	4	30	50	45	30	0	45
23	1	0	0	60	0	50	60
23	2	0	50	60	30	50	60
23	3	30	50	60	30	0	60
23	4	30	0	60	0	0	60

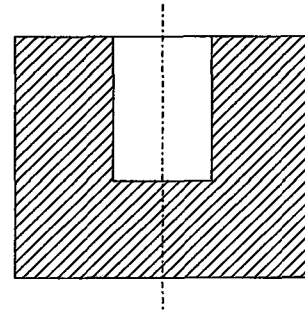
APPENDIX B
GEOMETRICAL REPRESENTATION OF FEATURES
CURRENTLY HANDLED BY THE OPPS-PRI



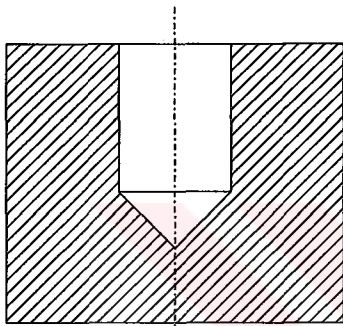
FACE



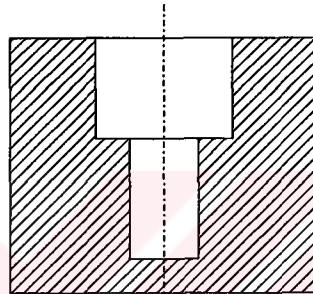
T-HOL



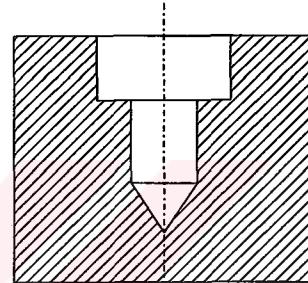
FBB-HOL



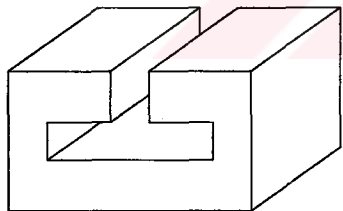
PBB-HOL



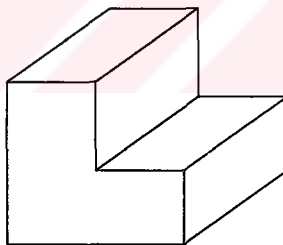
TC-BOR



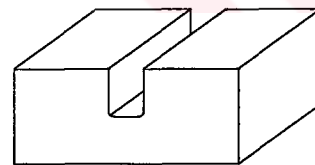
PBC-BOR



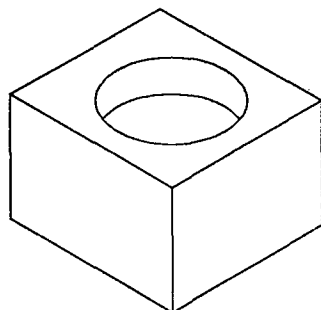
TSLOT



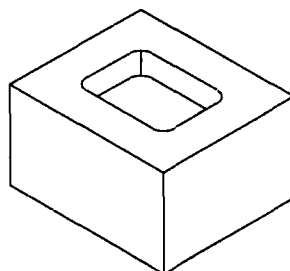
T-STP



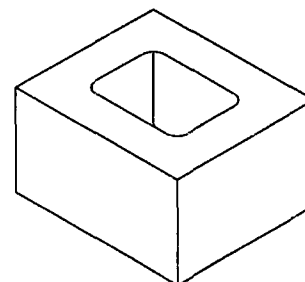
T-SLT



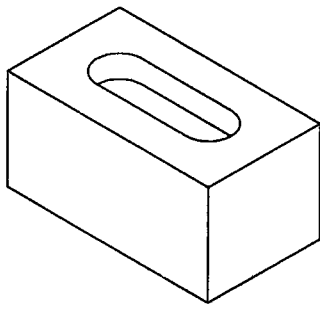
C-PKT



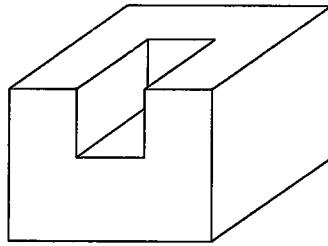
R-PKT



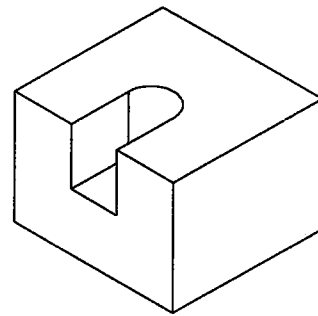
R-HOL



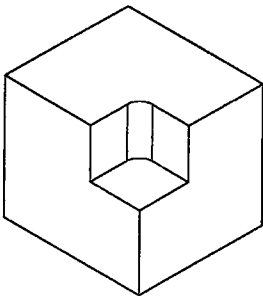
F-SLT



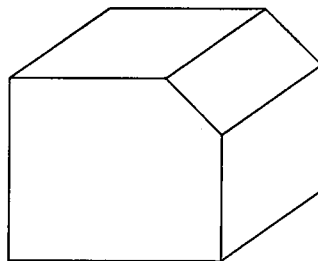
E-PKT



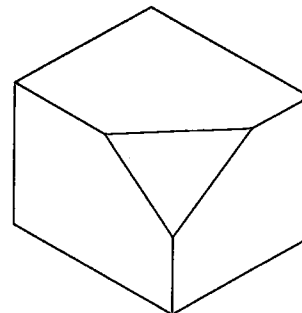
B-SLT



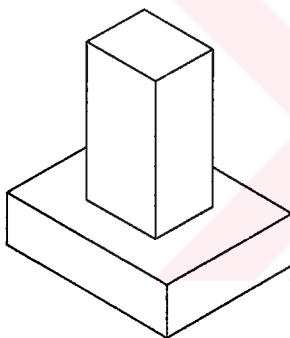
B-STP



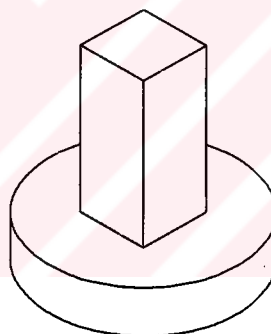
E-CUT



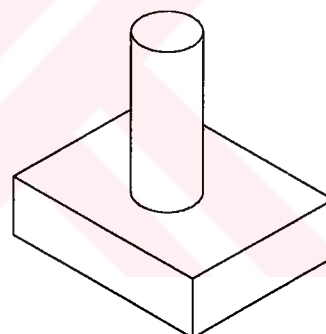
CE-CUT



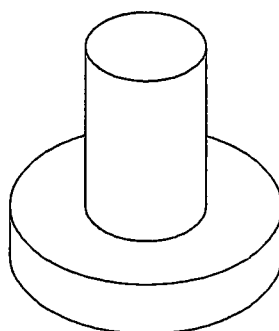
PB-PB



PB-CB

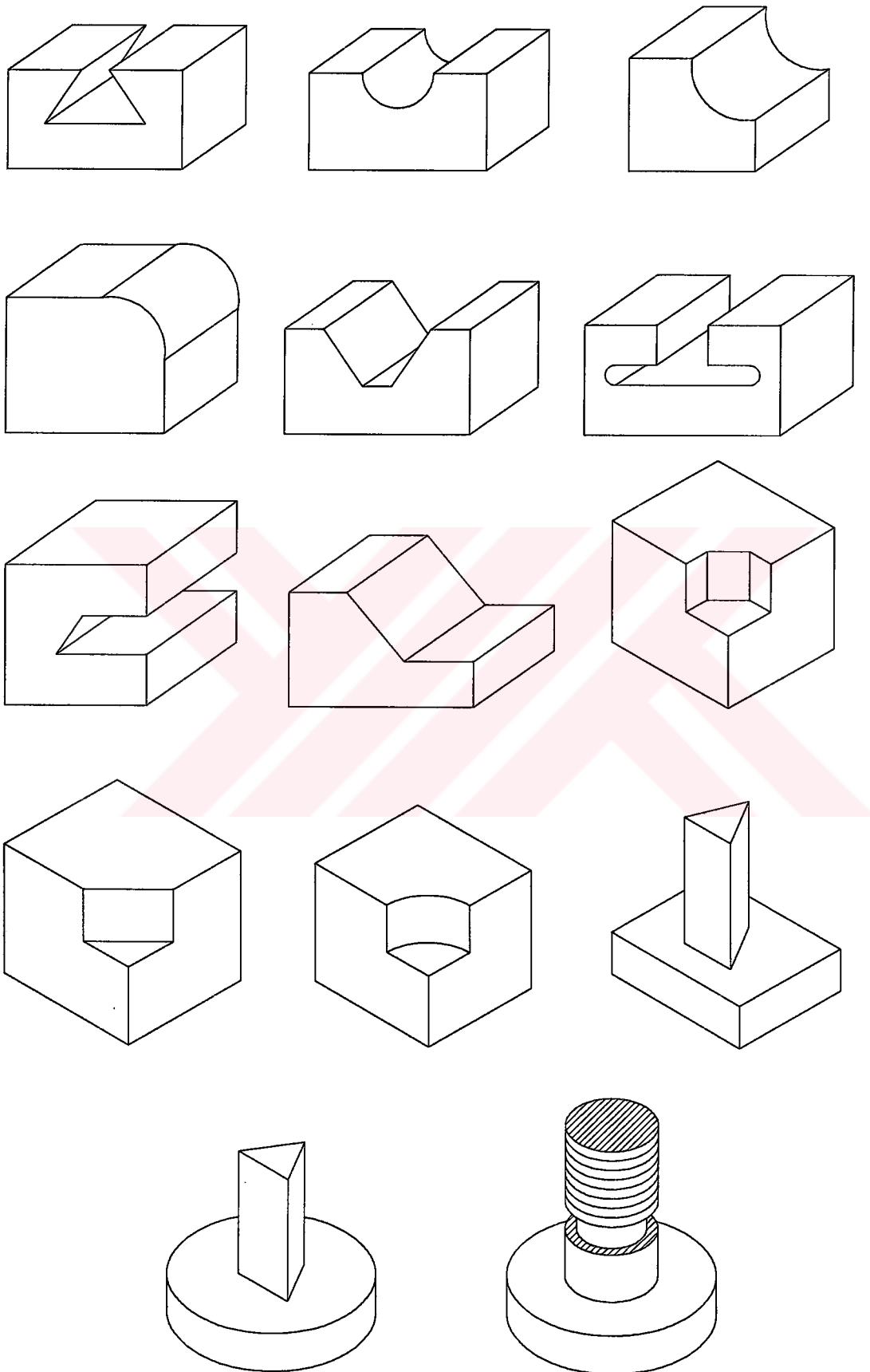


CB-PB



CB-CB

**OTHER TYPES OF FEATURES THAT CAN BE ADDED
TO THE SYSTEM WITH LITTLE MODIFICATIONS**



APPENDIX C

ISO CODING SCHEME FOR MILLING INSERTS

INDEXABLE INSERTS - MILLING

Extract from ISO 1832-1985

S	P	K	N	12	03	ED	R
1	2	3	4	5	6	7	9

1 Insert shape					
H	O	P	R	S	T
C	D	E	M	V	W
L	A	B	K		

2 Clearance angle on major cutting edge			
A	B	C	D
E	F	G	N
	Indicated for other clearance angles requiring specific description		
P	O		

4 Chipbreaker and clamp type	
N	R
F	A
M	G
W	T
Q	U
Special design and for unequal inserts	
X	

3 Tolerances			
Letter symbol	Tolerances, mm		
	m	s	I.C.
A ¹	±0,005	±0,025	±0,025
F ¹	±0,005	±0,025	±0,013
C ¹	±0,013	±0,025	±0,025
H	±0,013	±0,025	±0,013
E	±0,025	±0,025	±0,025
G	±0,025	±0,13	±0,025
J ¹	±0,005	±0,025	±0,05 ²
K ¹	±0,013	±0,025	±0,05 ²
L ¹	±0,025	±0,025	±0,05 ²
M	±0,08 ²	±0,13	±0,05 ²
N	±0,08 ²	±0,025	±0,05 ²
U	±0,13 ²	±0,13	±0,05 ²

1) These tolerance classes normally apply to inserts with parallel land.
2) The tolerance is dependent upon the insert size and should be indicated for each insert according to the standard tolerance for the corresponding size. See tables below.

LC: theoretical diameter of inscribed circle
s: insert thickness
m: see fig.

Insert shapes H, O, P, S, T, C, E, M, W, R			
Inscribed circle	Tolerances for m class M	Tolerances for m class U	Tolerances for I.C. class M, J, K, L
6,35	±0,08	±0,13	±0,05
9,525 (10)	±0,08	±0,13	±0,05
12,7 (12)	±0,13	±0,20	±0,08
15,875 (16)	±0,15	±0,27	±0,10
19,05 (20)	±0,15	±0,27	±0,10
25,4	±0,18	±0,38	±0,13

Insert shapes D		
Inscribed circle	Tolerances for m	Tolerances for I.C.
6,35	±0,11	±0,05
9,525	±0,11	±0,05
12,70	±0,15	±0,08
15,875	±0,18	±0,10
19,5	±0,18	±0,10

5 Cutting edge length, l mm	
Integers to be preceded by 0, eg. 0,52 mm indicated with 05	

6 Insert thickness, s mm	
01 s = 1,59	04 s = 4,76
11 s = 1,98	05 s = 5,56
02 s = 2,38	06 s = 6,35
03 s = 3,18	07 s = 7,94
13 s = 3,97	09 s = 9,52

7 Parallel land, clearance angle		
Parallel land	Radius, mm	
		00 - Sharp
A - 45°	A - 3°	02 - 0,2
D - 60°	B - 5°	04 - 0,4
E - 75°	C - 7°	08 - 0,8
F - 85°	D - 15°	12 - 1,2
P - 90°	E - 20°	16 - 1,6
Z - Others	F - 25°	20 - 2,0
	G - 30°	24 - 2,4
	N - 0°	32 - 3,2
	Z - Others	X - Others

MO - Round inserts

9 Feed direction	

Comparison cutting edge length in mm (pos. 5) to I.C. in inches								
	06	09	11	16	22	27	33	44
101				09	12	15	19	25
55°			07	11	15	19	23	31
80°			06	09	12	16	19	25
I.C.	5/32"	7/32"	1/4"	3/8"	1/2"	5/8"	3/4"	1"

The ISO code consists of nine symbols including 8 and/or 9 which are used only when required. In addition, the manufacturer may add further symbols joined to the ISO code through a hyphen (eg. -71 for the chipbreaker design).

APPENDIX D

THIN-WALLED HOLLOW SHAFTS

Consider a hollow cylindrical member of a noncircular section subjected to a torsional loading as shown in Figure A.1. While the thickness of the wall may vary within the transverse section, it is assumed that the section remains small compared to the other dimensions of the member.

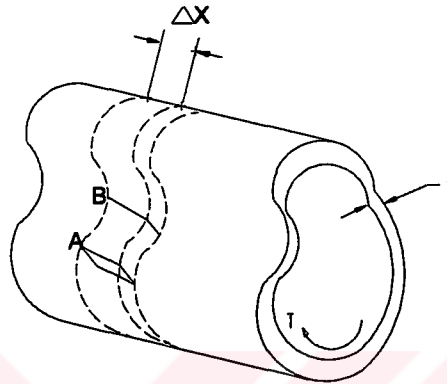


Figure A.1. A Small Section on a Hollow Cylindrical Member

The upper face of wall **AB** is detached from the member which is bounded by two transverse planes at a distance ΔX from each other, and by two longitudinal planes perpendicular to the wall. Since the portion **AB** is in equilibrium, the sum of the forces exerted on it in the longitudinal **X** direction must be zero as shown in Figure A.2.

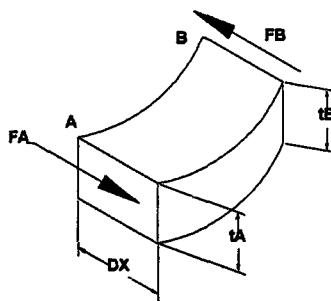


Figure A.2. Forces Exerted on the Small Section (Element)

The only forces involved are the shearing forces **FA** and **FB** exerted on the end of portion **AB**. Therefore; $\sum F_X = 0$, $F_A - F_B = 0$ (Equation A.1). **FA** can be expressed as the product of the longitudinal shearing stress τ_A on the small face at A

and of the area $t_A \Delta X$ of that face; $F_A = \tau_A (t_A \Delta X)$. Note that, while the shearing stress is independent of the X coordinate of the point considered, it may vary across the wall; thus τ_A represents the average value of the stress computed across the wall. Expressing F_B in a similar way and substituting for F_A and F_B into $F_A - F_B = 0$; following equations can be written; $\tau_A (t_A \Delta X) - \tau_B (t_B \Delta X) = 0$. These may be rewritten in the form of; $\tau_A t_A = \tau_B t_B$ (Equation A.2). Since A and B were chosen arbitrarily, the last equation expresses the product τt of the longitudinal shearing stress τ and of the wall thickness t is constant throughout the member. Denoting this product by q ; following relation is obtained $q = \tau t$ (Equation A.3). A small element from the wall portion AB is then detached. Since the upper and lower faces of this element are part of the free surface of the hollow member, the stresses on these faces are equal to zero. Thus the shearing stress at any point of a transverse section of the hollow member is parallel to the wall surface as shown in Figure A.3 and its average value computed across the wall satisfies Equation A.3.

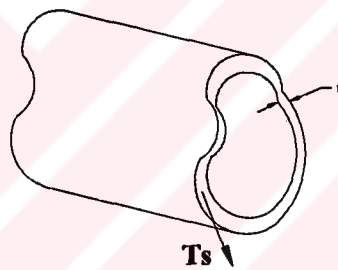


Figure A.3. Shearing Stress at Any Point of a Transverse Section

An analogy can be established between the distribution of the shearing stress τ in the transverse section of a thin walled hollow shaft and the distribution of the velocities v in the water flowing through a closed channel of unit depth and variable width. With the velocity v of the water varies from point to point on account of the variation in the width t of the channel, the rate of flow, $q = vt$, remains constant throughout the channel, just as τt in (Equation A.3). Because of this analogy, the product $q = \tau t$ is referred to the shear flow in the wall of the hollow body.

A relation can also be derived between the torque T applied to a hollow member and the shear flow q in its wall. A small element of the wall section, of length ds can be considered as shown in Figure A.4. The area of the element is $dA = t \cdot ds$, and the magnitude of the shearing force dF exerted on the element is;

$$dF = \tau dA = \tau (t \cdot ds) = (\tau t) ds = q ds \quad (\text{Equation A.4}).$$

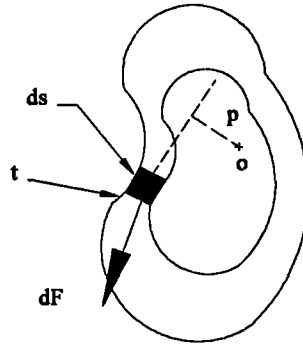


Figure A.4. A Small Element of the Wall Section

The moment dM_o of this force about an arbitrary point o within the cavity of the member may be obtained by multiplying dF by the perpendicular distance P from o to the line action of dF which yields; $dM_o = PdF = P(qds)$ (Equation A.5). However, the product Pds is equal to twice the area $d@$ of the triangle in Figure A.5.

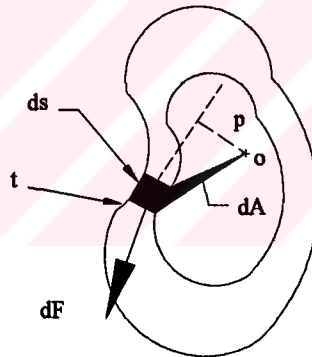


Figure A.5. Triangular Area

Thus, dM_o can be rewritten in the form; $dM_o = q(2 d@)$ (Equation A.6). Since the integral around the wall section of the left hand member of Equation A.6 represents the sum of the moments of all the elementary shearing forces exerted on the wall section, and since this sum is equal to the torque applied to the hollow member, T can be formulated as; $T = \int dM_o = \int q(2 d@)$. Since the shear flow q is a constant, this equation can be written as; $T = 2q@$ (Equation A.7), where $@$ is the area bounded by the center line of the wall cross section as shown in Figure A.6.

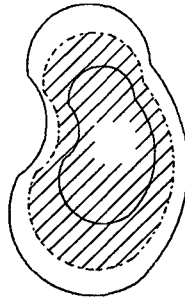


Figure A.6. Area @

The shearing stress τ at any given point of the wall may be expressed in terms of the torque T if we substitute for q from Equation A.3 into Equation A.7, and solved for τ , the following equation is obtained; $\tau = T / (2t@)$ (Equation A.8), where t is the wall thickness at the point considered and @ the area bounded by the center line. We recall that τ represents the average value of the shearing stress across the wall. However for elastic deformations the distribution of the stresses across the wall may be assumed uniform, and Equation A.8 will yield the actual value of the shearing stress at a given point of the wall.

APPENDIX E

G & M CODES

G code	Function
G00	Positioning
G01	Linear interpolation
G02	Circular interpolation/Helical interpolation CW
G03	Circular interpolation/Helical interpolation CCW
G04	Dwell, Exact stop
G05	High speed cycle machining
G09	Exact stop
G10	Data setting
G11	Data setting mode cancel
G15	Polar coordinates command cancel
G16	Polar coordinates command
G17	XpYp plane selection Xp: X axis or its parallel axis
G18	ZpXp plane selection Yp: Y axis or its parallel axis
G19	YpZp plane selection Zp: Z axis or its parallel axis
G20	Input in inch
G21	Input in mm
G22	Stored stroke check function on
G23	Stored stroke check function off
G27	Reference position return chuck
G28	Return the reference position
G29	Return from reference position
G30	2nd, 3rd and 4th reference position return
G31	Skip function
G33	Thread cutting
G37	Automatic tool length measurement
G39	Corner offset circular interpolation
G40	Cutter compensation cancel
G41	Cutter compensation left
G42	Cutter compensation right
G43	Tool length compensation + direction
G44	Tool length compensation - direction
G45	Tool offset increase
G46	Tool offset decrease
G47	Tool offset double increase
G48	Tool offset double decrease
G49	Tool length compensation cancel
G50	Scaling cancel
G51	Scaling
G52	Local coordinate system setting
G53	Machine coordinate system selection
G54	Workpiece coordinate system 1 selection
G55	Workpiece coordinate system 2 selection
G56	Workpiece coordinate system 3 selection
G57	Workpiece coordinate system 4 selection
G58	Workpiece coordinate system 5 selection
G59	Workpiece coordinate system 6 selection
G60	Single direction positioning
G61	Exact stop mode
G62	Automatic corner override
G63	Tapping mode

G64	Cutting mode
G65	Macro call
G66	Macro modal call
G67	Macro modal call cancel
G68	Coordinate rotation
G69	Coordinate rotation cancel
G73	Peck drilling cycle
G74	Counter tapping cycle
G75	Plunge grinding cycle (0-GSC)
G76	Fine boring cycle
G77	Direct constant-dimension plunge grinding cycle (0-GSC)
G78	Continuous-feed surface grinding cycle (0-GSC)
G79	Intermittent-feed surface grinding cycle (0-GSC)
G80	Canned cycle cancel/external operation cancel
G81	Drilling cycle, spot boring cycle or external operation function
G82	Drilling cycle or counter boring cycle
G83	Peck drilling cycle
G84	Tapping cycle
G85	Boring cycle
G86	Boring cycle
G87	Back boring cycle
G88	Boring cycle
G89	Boring cycle
G90	Absolute command
G91	Increment command
G92	Setting for work coordinate system or clamp at maximum spindle speed
G94	Feed per minute
G95	Feed per rotation
G96	Constant surface speed control
G97	Constant surface speed control cancel
G98	Return to initial point in canned cycle
G99	Return to R point in canned cycle
G107	Cylindrical interpolation
G150	Normal direction control cancel mode
G151	Normal direction control left side on
G152	Normal direction control right side on
G160	In-feed control function cancel (0-GSC)
G161	In-feed control function (0-GSC)

Code	Function
M00	Program stop
M01	Optional stop
M02	End of program
M03	Spindle rotation in CW
M04	Spindle rotation in CCW
M05	Spindle off (stop)
M06	Tool change
M08	Coolant on
M09	Coolant off
M30	End of program
M98	Subroutine call
M99	Continuation code

APPENDIX F

LISTING OF THE STEP FILE OF THE EXAMPLE COMPONENT AS SHOWN IN FIGURE 9.11

```

ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('','1');
FILE_NAME('TURKAY-TEZ2','1997-12-25T14:26:57','(dereli)',(''),
'PRO/ENGINEER BY PARAMETRIC TECHNOLOGY CORPORATION, 9709',
'PRO/ENGINEER BY PARAMETRIC TECHNOLOGY CORPORATION, 9709','');
FILE_SCHEMA(('CONFIG_CONTROL_DESIGN'));
ENDSEC;
DATA;
#1=DIRECTION('',(-1.E0,0.E0,0.E0));
#2=VECTOR('',#1,2.E2);
#3=CARTESIAN_POINT('',(0.E0,0.E0,0.E0));
#4=LINE('',#3,#2);
#5=DIRECTION('',(0.E0,0.E0,-1.E0));
#6=VECTOR('',#5,5.9E1);
#7=CARTESIAN_POINT('',(-2.E2,0.E0,0.E0));
#8=LINE('',#7,#6);
#9=DIRECTION('',(1.E0,0.E0,0.E0));
#10=VECTOR('',#9,7.E1);
#11=CARTESIAN_POINT('',(-2.E2,0.E0,-5.9E1));
#12=LINE('',#11,#10);
#13=DIRECTION('',(0.E0,0.E0,-1.E0));
#14=VECTOR('',#13,6.E0);
#15=CARTESIAN_POINT('',(-1.3E2,0.E0,-5.9E1));
#16=LINE('',#15,#14);
#17=DIRECTION('',(1.E0,0.E0,0.E0));
#18=VECTOR('',#17,7.E1);
#19=CARTESIAN_POINT('',(-1.3E2,0.E0,-6.5E1));
#20=LINE('',#19,#18);
#21=DIRECTION('',(0.E0,0.E0,1.E0));
#22=VECTOR('',#21,8.E0);
#23=CARTESIAN_POINT('',(-6.E1,0.E0,-6.5E1));
#24=LINE('',#23,#22);
#25=DIRECTION('',(1.E0,0.E0,0.E0));
#26=VECTOR('',#25,2.E1);
#27=CARTESIAN_POINT('',(-6.E1,0.E0,-5.7E1));
#28=LINE('',#27,#26);
#29=DIRECTION('',(0.E0,0.E0,-1.E0));
#30=VECTOR('',#29,8.E0);
#31=CARTESIAN_POINT('',(-4.E1,0.E0,-5.7E1));
#32=LINE('',#31,#30);
#33=DIRECTION('',(1.E0,0.E0,0.E0));
#34=VECTOR('',#33,4.E1);
#35=CARTESIAN_POINT('',(-4.E1,0.E0,-6.5E1));
#36=LINE('',#35,#34);
#37=DIRECTION('',(0.E0,0.E0,1.E0));
#38=VECTOR('',#37,6.5E1);
#39=CARTESIAN_POINT('',(0.E0,0.E0,-6.5E1));
#40=LINE('',#39,#38);
#41=CARTESIAN_POINT('',(-7.5E1,0.E0,-3.1E1));
#42=DIRECTION('',(0.E0,1.E0,0.E0));
#43=DIRECTION('',(-1.E0,0.E0,0.E0));
#44=AXIS2_PLACEMENT_3D('',#41,#42,#43);
#46=CARTESIAN_POINT('',(-7.5E1,0.E0,-3.1E1));

```

```

#47=DIRECTION(",(0.E0,1.E0,0.E0));
#48=DIRECTION(",(1.E0,0.E0,0.E0));
#49=AXIS2_PLACEMENT_3D(",#46,#47,#48);
#51=CARTESIAN_POINT(",-8.775E1,0.E0,-3.1E1));
#52=DIRECTION(",(0.E0,1.E0,0.E0));
#53=DIRECTION(",-1.E0,0.E0,0.E0));
#54=AXIS2_PLACEMENT_3D(",#51,#52,#53);
#56=CARTESIAN_POINT(",-8.775E1,0.E0,-3.1E1));
#57=DIRECTION(",(0.E0,1.E0,0.E0));
#58=DIRECTION(",(1.E0,0.E0,0.E0));
#59=AXIS2_PLACEMENT_3D(",#56,#57,#58);
#61=CARTESIAN_POINT(",-1.65E2,0.E0,-3.1E1));
#62=DIRECTION(",(0.E0,1.E0,0.E0));
#63=DIRECTION(",-1.E0,0.E0,0.E0));
#64=AXIS2_PLACEMENT_3D(",#61,#62,#63);
#66=CARTESIAN_POINT(",-1.65E2,0.E0,-3.1E1));
#67=DIRECTION(",(0.E0,1.E0,0.E0));
#68=DIRECTION(",(1.E0,0.E0,0.E0));
#69=AXIS2_PLACEMENT_3D(",#66,#67,#68);
#71=DIRECTION(",(0.E0,-1.E0,0.E0));
#72=VECTOR(",#71,4.6E1);
#73=CARTESIAN_POINT(",(0.E0,0.E0,0.E0));
#74=LINE(",#73,#72);
#75=DIRECTION(",(0.E0,0.E0,1.E0));
#76=VECTOR(",#75,1.E1);
#77=CARTESIAN_POINT(",(0.E0,-2.E1,-6.5E1));
#78=LINE(",#77,#76);
#79=DIRECTION(",(0.E0,-1.E0,0.E0));
#80=VECTOR(",#79,6.E0);
#81=CARTESIAN_POINT(",(0.E0,-2.E1,-6.5E1));
#82=LINE(",#81,#80);
#83=DIRECTION(",(0.E0,0.E0,1.E0));
#84=VECTOR(",#83,5.E0);
#85=CARTESIAN_POINT(",(0.E0,-2.6E1,-6.5E1));
#86=LINE(",#85,#84);
#87=DIRECTION(",(0.E0,-1.E0,0.E0));
#88=VECTOR(",#87,2.E1);
#89=CARTESIAN_POINT(",(0.E0,-2.6E1,-6.E1));
#90=LINE(",#89,#88);
#91=DIRECTION(",(0.E0,-1.E0,0.E0));
#92=VECTOR(",#91,6.E0);
#93=CARTESIAN_POINT(",(0.E0,0.E0,-6.5E1));
#94=LINE(",#93,#92);
#95=DIRECTION(",(0.E0,0.E0,1.E0));
#96=VECTOR(",#95,1.E1);
#97=CARTESIAN_POINT(",(0.E0,-6.E0,-6.5E1));
#98=LINE(",#97,#96);
#99=DIRECTION(",(0.E0,-1.E0,0.E0));
#100=VECTOR(",#99,1.4E1);
#101=CARTESIAN_POINT(",(0.E0,-6.E0,-5.5E1));
#102=LINE(",#101,#100);
#103=DIRECTION(",-1.E0,0.E0,0.E0));
#104=VECTOR(",#103,3.E1);
#105=CARTESIAN_POINT(",(0.E0,-2.E1,-6.5E1));
#106=LINE(",#105,#104);
#107=DIRECTION(",(0.E0,1.E0,0.E0));
#108=VECTOR(",#107,1.4E1);
#109=CARTESIAN_POINT(",-3.E1,-2.E1,-6.5E1));
#110=LINE(",#109,#108);
#111=DIRECTION(",-1.E0,0.E0,0.E0));

```

```

#112=VECTOR(",#111,3.E1);
#113=CARTESIAN_POINT(",(0.E0,-6.E0,-6.5E1));
#114=LINE(",#113,#112);
#115=DIRECTION(",(0.E0,-1.E0,0.E0));
#116=VECTOR(",#115,4.6E1);
#117=CARTESIAN_POINT(",(-4.E1,0.E0,-6.5E1));
#118=LINE(",#117,#116);
#119=DIRECTION(",(0.E0,1.E0,0.E0));
#120=VECTOR(",#119,2.E1);
#121=CARTESIAN_POINT",(-2.5E1,-4.6E1,-6.5E1));
#122=LINE(",#121,#120);
#123=DIRECTION(",(-1.E0,0.E0,0.E0));
#124=VECTOR(",#123,2.5E1);
#125=CARTESIAN_POINT",(0.E0,-2.6E1,-6.5E1));
#126=LINE(",#125,#124);
#127=DIRECTION",(0.E0,0.E0,1.E0));
#128=VECTOR(",#127,1.E1);
#129=CARTESIAN_POINT",(-3.E1,-2.E1,-6.5E1));
#130=LINE(",#129,#128);
#131=DIRECTION",(1.E0,0.E0,0.E0));
#132=VECTOR(",#131,3.E1);
#133=CARTESIAN_POINT",(-3.E1,-6.E0,-5.5E1));
#134=LINE(",#133,#132);
#135=DIRECTION",(0.E0,1.E0,0.E0));
#136=VECTOR(",#135,1.4E1);
#137=CARTESIAN_POINT",(-3.E1,-2.E1,-5.5E1));
#138=LINE(",#137,#136);
#139=DIRECTION",(-1.E0,0.E0,0.E0));
#140=VECTOR(",#139,3.E1);
#141=CARTESIAN_POINT",(0.E0,-2.E1,-5.5E1));
#142=LINE(",#141,#140);
#143=DIRECTION",(0.E0,0.E0,1.E0));
#144=VECTOR(",#143,1.E1);
#145=CARTESIAN_POINT",(-3.E1,-6.E0,-6.5E1));
#146=LINE(",#145,#144);
#147=DIRECTION",(0.E0,-1.E0,0.E0));
#148=VECTOR(",#147,4.6E1);
#149=CARTESIAN_POINT",(-4.E1,0.E0,-5.7E1));
#150=LINE(",#149,#148);
#151=DIRECTION",(0.E0,-1.E0,0.E0));
#152=VECTOR(",#151,4.6E1);
#153=CARTESIAN_POINT",(-6.E1,0.E0,-5.7E1));
#154=LINE(",#153,#152);
#155=DIRECTION",(0.E0,-1.E0,0.E0));
#156=VECTOR(",#155,4.6E1);
#157=CARTESIAN_POINT",(-6.E1,0.E0,-6.5E1));
#158=LINE(",#157,#156);
#159=DIRECTION",(0.E0,-1.E0,0.E0));
#160=VECTOR(",#159,4.6E1);
#161=CARTESIAN_POINT",(-1.3E2,0.E0,-6.5E1));
#162=LINE(",#161,#160);
#163=DIRECTION",(-1.E0,0.E0,0.E0));
#164=VECTOR(",#163,4.E1);
#165=CARTESIAN_POINT",(-7.5E1,-4.1E1,-6.5E1));
#166=LINE(",#165,#164);
#167=DIRECTION",(0.E0,1.E0,0.E0));
#168=VECTOR(",#167,2.E1);
#169=CARTESIAN_POINT",(-1.15E2,-4.1E1,-6.5E1));
#170=LINE(",#169,#168);
#171=DIRECTION",(1.E0,0.E0,0.E0));

```

```

#172=VECTOR(",#171,4.E1);
#173=CARTESIAN_POINT("(-1.15E2,-2.1E1,-6.5E1));
#174=LINE(",#173,#172);
#175=DIRECTION(",(0.E0,-1.E0,0.E0));
#176=VECTOR(",#175,2.E1);
#177=CARTESIAN_POINT("(-7.5E1,-2.1E1,-6.5E1));
#178=LINE(",#177,#176);
#179=DIRECTION(",(-1.E0,0.E0,0.E0));
#180=VECTOR(",#179,4.5E1);
#181=CARTESIAN_POINT("(-7.E1,-1.5E1,-6.5E1));
#182=LINE(",#181,#180);
#183=DIRECTION(",(0.E0,1.E0,0.E0));
#184=VECTOR(",#183,1.E1);
#185=CARTESIAN_POINT("(-1.15E2,-1.5E1,-6.5E1));
#186=LINE(",#185,#184);
#187=DIRECTION(",(1.E0,0.E0,0.E0));
#188=VECTOR(",#187,4.5E1);
#189=CARTESIAN_POINT("(-1.15E2,-5.E0,-6.5E1));
#190=LINE(",#189,#188);
#191=DIRECTION(",(0.E0,-1.E0,0.E0));
#192=VECTOR(",#191,1.E1);
#193=CARTESIAN_POINT("(-7.E1,-5.E0,-6.5E1));
#194=LINE(",#193,#192);
#195=DIRECTION(",(0.E0,-1.E0,0.E0));
#196=VECTOR(",#195,4.6E1);
#197=CARTESIAN_POINT("(-1.3E2,0.E0,-5.9E1));
#198=LINE(",#197,#196);
#199=DIRECTION(",(1.E0,0.E0,0.E0));
#200=VECTOR(",#199,5.E1);
#201=CARTESIAN_POINT("(-2.E2,-3.55E1,-5.9E1));
#202=LINE(",#201,#200);
#203=DIRECTION(",(0.E0,-1.E0,0.E0));
#204=VECTOR(",#203,1.05E1);
#205=CARTESIAN_POINT("(-2.E2,-3.55E1,-5.9E1));
#206=LINE(",#205,#204);
#207=DIRECTION(",(0.E0,-1.E0,0.E0));
#208=VECTOR(",#207,1.05E1);
#209=CARTESIAN_POINT("(-2.E2,0.E0,-5.9E1));
#210=LINE(",#209,#208);
#211=DIRECTION(",(1.E0,0.E0,0.E0));
#212=VECTOR(",#211,5.E1);
#213=CARTESIAN_POINT("(-2.E2,-1.05E1,-5.9E1));
#214=LINE(",#213,#212);
#215=DIRECTION(",(0.E0,-1.E0,0.E0));
#216=VECTOR(",#215,2.5E1);
#217=CARTESIAN_POINT("(-1.5E2,-1.05E1,-5.9E1));
#218=LINE(",#217,#216);
#219=DIRECTION(",(0.E0,0.E0,1.E0));
#220=VECTOR(",#219,1.E1);
#221=CARTESIAN_POINT("(-2.E2,-3.55E1,-5.9E1));
#222=LINE(",#221,#220);
#223=DIRECTION(",(0.E0,1.E0,0.E0));
#224=VECTOR(",#223,2.5E1);
#225=CARTESIAN_POINT("(-2.E2,-3.55E1,-4.9E1));
#226=LINE(",#225,#224);
#227=DIRECTION(",(0.E0,0.E0,1.E0));
#228=VECTOR(",#227,1.E1);
#229=CARTESIAN_POINT("(-2.E2,-1.05E1,-5.9E1));
#230=LINE(",#229,#228);
#231=DIRECTION(",(0.E0,-1.E0,0.E0));

```

```

#232=VECTOR(",#231,4.6E1);
#233=CARTESIAN_POINT(",-2.E2,0.E0,0.E0));
#234=LINE(",#233,#232);
#235=DIRECTION(",(1.E0,0.E0,0.E0));
#236=VECTOR(",#235,5.E1);
#237=CARTESIAN_POINT(",-2.E2,-3.55E1,-4.9E1));
#238=LINE(",#237,#236);
#239=DIRECTION(",(0.E0,0.E0,-1.E0));
#240=VECTOR(",#239,1.E1);
#241=CARTESIAN_POINT(",-1.5E2,-1.05E1,-4.9E1));
#242=LINE(",#241,#240);
#243=DIRECTION(",(0.E0,1.E0,0.E0));
#244=VECTOR(",#243,2.5E1);
#245=CARTESIAN_POINT(",-1.5E2,-3.55E1,-4.9E1));
#246=LINE(",#245,#244);
#247=DIRECTION(",(0.E0,0.E0,1.E0));
#248=VECTOR(",#247,1.E1);
#249=CARTESIAN_POINT(",-1.5E2,-3.55E1,-5.9E1));
#250=LINE(",#249,#248);
#251=DIRECTION(",(1.E0,0.E0,0.E0));
#252=VECTOR(",#251,5.E1);
#253=CARTESIAN_POINT(",-2.E2,-1.05E1,-4.9E1));
#254=LINE(",#253,#252);
#255=DIRECTION(",(0.E0,0.E0,1.E0));
#256=VECTOR(",#255,5.E0);
#257=CARTESIAN_POINT(",-2.5E1,-4.6E1,-6.5E1));
#258=LINE(",#257,#256);
#259=DIRECTION(",(1.E0,0.E0,0.E0));
#260=VECTOR(",#259,1.5E1);
#261=CARTESIAN_POINT(",-4.E1,-4.6E1,-6.5E1));
#262=LINE(",#261,#260);
#263=DIRECTION(",(0.E0,0.E0,-1.E0));
#264=VECTOR(",#263,8.E0);
#265=CARTESIAN_POINT(",-4.E1,-4.6E1,-5.7E1));
#266=LINE(",#265,#264);
#267=DIRECTION(",(1.E0,0.E0,0.E0));
#268=VECTOR(",#267,2.E1);
#269=CARTESIAN_POINT(",-6.E1,-4.6E1,-5.7E1));
#270=LINE(",#269,#268);
#271=DIRECTION(",(0.E0,0.E0,1.E0));
#272=VECTOR(",#271,8.E0);
#273=CARTESIAN_POINT(",-6.E1,-4.6E1,-6.5E1));
#274=LINE(",#273,#272);
#275=DIRECTION(",(1.E0,0.E0,0.E0));
#276=VECTOR(",#275,7.E1);
#277=CARTESIAN_POINT(",-1.3E2,-4.6E1,-6.5E1));
#278=LINE(",#277,#276);
#279=DIRECTION(",(0.E0,0.E0,-1.E0));
#280=VECTOR(",#279,6.E0);
#281=CARTESIAN_POINT(",-1.3E2,-4.6E1,-5.9E1));
#282=LINE(",#281,#280);
#283=DIRECTION(",(1.E0,0.E0,0.E0));
#284=VECTOR(",#283,7.E1);
#285=CARTESIAN_POINT(",-2.E2,-4.6E1,-5.9E1));
#286=LINE(",#285,#284);
#287=DIRECTION(",(0.E0,0.E0,-1.E0));
#288=VECTOR(",#287,5.9E1);
#289=CARTESIAN_POINT(",-2.E2,-4.6E1,0.E0));
#290=LINE(",#289,#288);
#291=DIRECTION(",-1.E0,0.E0,0.E0));

```

```

#292=VECTOR(",#291,2.E2);
#293=CARTESIAN_POINT(",(0.E0,-4.6E1,0.E0));
#294=LINE(",#293,#292);
#295=DIRECTION(",(0.E0,0.E0,1.E0));
#296=VECTOR(",#295,6.E1);
#297=CARTESIAN_POINT(",(0.E0,-4.6E1,-6.E1));
#298=LINE(",#297,#296);
#299=DIRECTION(",(1.E0,0.E0,0.E0));
#300=VECTOR(",#299,2.5E1);
#301=CARTESIAN_POINT(",(0.E0,-4.6E1,-6.E1));
#302=LINE(",#301,#300);
#303=CARTESIAN_POINT(",(7.5E1,-4.6E1,-3.1E1));
#304=DIRECTION(",(0.E0,1.E0,0.E0));
#305=DIRECTION(",(1.E0,0.E0,0.E0));
#306=AXIS2_PLACEMENT_3D(",#303,#304,#305);
#308=CARTESIAN_POINT(",(7.5E1,-4.6E1,-3.1E1));
#309=DIRECTION(",(0.E0,1.E0,0.E0));
#310=DIRECTION(",(1.E0,0.E0,0.E0));
#311=AXIS2_PLACEMENT_3D(",#308,#309,#310);
#313=CARTESIAN_POINT(",(8.775E1,-4.6E1,-3.1E1));
#314=DIRECTION(",(0.E0,1.E0,0.E0));
#315=DIRECTION(",(1.E0,0.E0,0.E0));
#316=AXIS2_PLACEMENT_3D(",#313,#314,#315);
#318=CARTESIAN_POINT(",(8.775E1,-4.6E1,-3.1E1));
#319=DIRECTION(",(0.E0,1.E0,0.E0));
#320=DIRECTION(",(1.E0,0.E0,0.E0));
#321=AXIS2_PLACEMENT_3D(",#318,#319,#320);
#323=CARTESIAN_POINT(",(1.65E2,-4.6E1,-3.1E1));
#324=DIRECTION(",(0.E0,1.E0,0.E0));
#325=DIRECTION(",(1.E0,0.E0,0.E0));
#326=AXIS2_PLACEMENT_3D(",#323,#324,#325);
#328=CARTESIAN_POINT(",(1.65E2,-4.6E1,-3.1E1));
#329=DIRECTION(",(0.E0,1.E0,0.E0));
#330=DIRECTION(",(1.E0,0.E0,0.E0));
#331=AXIS2_PLACEMENT_3D(",#328,#329,#330);
#333=DIRECTION(",(1.E0,0.E0,0.E0));
#334=VECTOR(",#333,2.5E1);
#335=CARTESIAN_POINT(",(2.5E1,-2.6E1,-6.E1));
#336=LINE(",#335,#334);
#337=DIRECTION(",(0.E0,1.E0,0.E0));
#338=VECTOR(",#337,2.E1);
#339=CARTESIAN_POINT(",(2.5E1,-4.6E1,-6.E1));
#340=LINE(",#339,#338);
#341=DIRECTION(",(0.E0,0.E0,1.E0));
#342=VECTOR(",#341,5.E0);
#343=CARTESIAN_POINT(",(2.5E1,-2.6E1,-6.5E1));
#344=LINE(",#343,#342);
#345=DIRECTION(",(0.E0,1.E0,0.E0));
#346=VECTOR(",#345,4.6E1);
#347=CARTESIAN_POINT(",(8.15E1,-4.6E1,-3.1E1));
#348=LINE(",#347,#346);
#349=DIRECTION(",(0.E0,1.E0,0.E0));
#350=VECTOR(",#349,4.6E1);
#351=CARTESIAN_POINT(",(6.85E1,-4.6E1,-3.1E1));
#352=LINE(",#351,#350);
#353=DIRECTION(",(0.E0,1.E0,0.E0));
#354=VECTOR(",#353,4.6E1);
#355=CARTESIAN_POINT(",(9.375E1,-4.6E1,-3.1E1));
#356=LINE(",#355,#354);
#357=DIRECTION(",(0.E0,1.E0,0.E0));

```

```

#358=VECTOR(" ,#357,4.6E1);
#359=CARTESIAN_POINT(" ,(-8.175E1,-4.6E1,-3.1E1));
#360=LINE(" ,#359,#358);
#361=DIRECTION(" ,(0.E0,1.E0,0.E0));
#362=VECTOR(" ,#361,4.6E1);
#363=CARTESIAN_POINT(" ,(-1.7E2,-4.6E1,-3.1E1));
#364=LINE(" ,#363,#362);
#365=DIRECTION(" ,(0.E0,1.E0,0.E0));
#366=VECTOR(" ,#365,4.6E1);
#367=CARTESIAN_POINT(" ,(-1.6E2,-4.6E1,-3.1E1));
#368=LINE(" ,#367,#366);
#369=DIRECTION(" ,(0.E0,0.E0,1.E0));
#370=VECTOR(" ,#369,3.E0);
#371=CARTESIAN_POINT(" ,(-7.5E1,-4.1E1,-6.5E1));
#372=LINE(" ,#371,#370);
#373=DIRECTION(" ,(0.E0,0.E0,1.E0));
#374=VECTOR(" ,#373,3.E0);
#375=CARTESIAN_POINT(" ,(-7.5E1,-2.1E1,-6.5E1));
#376=LINE(" ,#375,#374);
#377=DIRECTION(" ,(0.E0,0.E0,1.E0));
#378=VECTOR(" ,#377,3.E0);
#379=CARTESIAN_POINT(" ,(-1.15E2,-2.1E1,-6.5E1));
#380=LINE(" ,#379,#378);
#381=DIRECTION(" ,(0.E0,0.E0,1.E0));
#382=VECTOR(" ,#381,3.E0);
#383=CARTESIAN_POINT(" ,(-1.15E2,-4.1E1,-6.5E1));
#384=LINE(" ,#383,#382);
#385=DIRECTION(" ,(-1.E0,0.E0,0.E0));
#386=VECTOR(" ,#385,4.E1);
#387=CARTESIAN_POINT(" ,(-7.5E1,-4.1E1,-6.2E1));
#388=LINE(" ,#387,#386);
#389=DIRECTION(" ,(0.E0,-1.E0,0.E0));
#390=VECTOR(" ,#389,2.E1);
#391=CARTESIAN_POINT(" ,(-7.5E1,-2.1E1,-6.2E1));
#392=LINE(" ,#391,#390);
#393=DIRECTION(" ,(1.E0,0.E0,0.E0));
#394=VECTOR(" ,#393,4.E1);
#395=CARTESIAN_POINT(" ,(-1.15E2,-2.1E1,-6.2E1));
#396=LINE(" ,#395,#394);
#397=DIRECTION(" ,(0.E0,1.E0,0.E0));
#398=VECTOR(" ,#397,2.E1);
#399=CARTESIAN_POINT(" ,(-1.15E2,-4.1E1,-6.2E1));
#400=LINE(" ,#399,#398);
#401=CARTESIAN_POINT(" ,(-9.5E1,-3.1E1,-6.2E1));
#402=DIRECTION(" ,(0.E0,0.E0,1.E0));
#403=DIRECTION(" ,(1.E0,0.E0,0.E0));
#404=AXIS2_PLACEMENT_3D(" ,#401,#402,#403);
#406=CARTESIAN_POINT(" ,(-9.5E1,-3.1E1,-6.2E1));
#407=DIRECTION(" ,(0.E0,0.E0,1.E0));
#408=DIRECTION(" ,(-1.E0,0.E0,0.E0));
#409=AXIS2_PLACEMENT_3D(" ,#406,#407,#408);
#411=DIRECTION(" ,(0.E0,0.E0,1.E0));
#412=VECTOR(" ,#411,2.E1);
#413=CARTESIAN_POINT(" ,(-9.05E1,-3.1E1,-6.2E1));
#414=LINE(" ,#413,#412);
#415=DIRECTION(" ,(0.E0,0.E0,1.E0));
#416=VECTOR(" ,#415,2.E1);
#417=CARTESIAN_POINT(" ,(-9.95E1,-3.1E1,-6.2E1));
#418=LINE(" ,#417,#416);
#419=CARTESIAN_POINT(" ,(-9.5E1,-3.1E1,-4.2E1));

```

```

#420=DIRECTION(",(0.E0,0.E0,1.E0));
#421=DIRECTION(",(1.E0,0.E0,0.E0));
#422=AXIS2_PLACEMENT_3D(",#419,#420,#421);
#424=CARTESIAN_POINT(",-9.5E1,-3.1E1,-4.2E1));
#425=DIRECTION(",(0.E0,0.E0,1.E0));
#426=DIRECTION(",-1.E0,0.E0,0.E0));
#427=AXIS2_PLACEMENT_3D(",#424,#425,#426);
#429=DIRECTION(",(0.E0,0.E0,1.E0));
#430=VECTOR(",#429,9.E0);
#431=CARTESIAN_POINT(",-7.E1,-1.5E1,-6.5E1));
#432=LINE(",#431,#430);
#433=DIRECTION(",(0.E0,0.E0,1.E0));
#434=VECTOR(",#433,9.E0);
#435=CARTESIAN_POINT(",-7.E1,-5.E0,-6.5E1));
#436=LINE(",#435,#434);
#437=DIRECTION(",(0.E0,0.E0,1.E0));
#438=VECTOR(",#437,9.E0);
#439=CARTESIAN_POINT(",-1.15E2,-5.E0,-6.5E1));
#440=LINE(",#439,#438);
#441=DIRECTION(",(0.E0,0.E0,1.E0));
#442=VECTOR(",#441,9.E0);
#443=CARTESIAN_POINT(",-1.15E2,-1.5E1,-6.5E1));
#444=LINE(",#443,#442);
#445=DIRECTION(",-1.E0,0.E0,0.E0));
#446=VECTOR(",#445,4.5E1);
#447=CARTESIAN_POINT(",-7.E1,-1.5E1,-5.6E1));
#448=LINE(",#447,#446);
#449=DIRECTION(",(0.E0,-1.E0,0.E0));
#450=VECTOR(",#449,1.E1);
#451=CARTESIAN_POINT(",-7.E1,-5.E0,-5.6E1));
#452=LINE(",#451,#450);
#453=DIRECTION(",(1.E0,0.E0,0.E0));
#454=VECTOR(",#453,4.5E1);
#455=CARTESIAN_POINT(",-1.15E2,-5.E0,-5.6E1));
#456=LINE(",#455,#454);
#457=DIRECTION(",(0.E0,1.E0,0.E0));
#458=VECTOR(",#457,1.E1);
#459=CARTESIAN_POINT(",-1.15E2,-1.5E1,-5.6E1));
#460=LINE(",#459,#458);
#461=CARTESIAN_POINT(",(0.E0,0.E0,0.E0));
#462=CARTESIAN_POINT(",-2.E2,0.E0,0.E0));
#463=VERTEX_POINT(",#461);
#464=VERTEX_POINT(",#462);
#465=CARTESIAN_POINT(",-2.E2,0.E0,-5.9E1));
#466=VERTEX_POINT(",#465);
#467=CARTESIAN_POINT(",-1.3E2,0.E0,-5.9E1));
#468=VERTEX_POINT(",#467);
#469=CARTESIAN_POINT(",-1.3E2,0.E0,-6.5E1));
#470=VERTEX_POINT(",#469);
#471=CARTESIAN_POINT(",-6.E1,0.E0,-6.5E1));
#472=VERTEX_POINT(",#471);
#473=CARTESIAN_POINT(",-6.E1,0.E0,-5.7E1));
#474=VERTEX_POINT(",#473);
#475=CARTESIAN_POINT(",-4.E1,0.E0,-5.7E1));
#476=VERTEX_POINT(",#475);
#477=CARTESIAN_POINT(",-4.E1,0.E0,-6.5E1));
#478=VERTEX_POINT(",#477);
#479=CARTESIAN_POINT(",(0.E0,0.E0,-6.5E1));
#480=VERTEX_POINT(",#479);
#481=CARTESIAN_POINT(",(0.E0,-4.6E1,0.E0));

```

```

#482=CARTESIAN_POINT("(-2.E2,-4.6E1,0.E0));
#483=VERTEX_POINT("#481);
#484=VERTEX_POINT("#482);
#485=CARTESIAN_POINT("(-2.E2,-4.6E1,-5.9E1));
#486=VERTEX_POINT("#485);
#487=CARTESIAN_POINT("(-1.3E2,-4.6E1,-5.9E1));
#488=VERTEX_POINT("#487);
#489=CARTESIAN_POINT("(-1.3E2,-4.6E1,-6.5E1));
#490=VERTEX_POINT("#489);
#491=CARTESIAN_POINT("(-6.E1,-4.6E1,-6.5E1));
#492=VERTEX_POINT("#491);
#493=CARTESIAN_POINT("(-6.E1,-4.6E1,-5.7E1));
#494=VERTEX_POINT("#493);
#495=CARTESIAN_POINT("(-4.E1,-4.6E1,-5.7E1));
#496=VERTEX_POINT("#495);
#497=CARTESIAN_POINT("(-4.E1,-4.6E1,-6.5E1));
#498=VERTEX_POINT("#497);
#499=CARTESIAN_POINT("(-1.5E2,-3.55E1,-4.9E1));
#500=CARTESIAN_POINT("(-1.5E2,-1.05E1,-4.9E1));
#501=VERTEX_POINT("#499);
#502=VERTEX_POINT("#500);
#503=CARTESIAN_POINT("(-2.E2,-3.55E1,-5.9E1));
#504=CARTESIAN_POINT("(-2.E2,-3.55E1,-4.9E1));
#505=VERTEX_POINT("#503);
#506=VERTEX_POINT("#504);
#507=CARTESIAN_POINT("(-2.E2,-1.05E1,-4.9E1));
#508=VERTEX_POINT("#507);
#509=CARTESIAN_POINT("(-2.E2,-1.05E1,-5.9E1));
#510=VERTEX_POINT("#509);
#511=CARTESIAN_POINT("(-1.5E2,-3.55E1,-5.9E1));
#512=VERTEX_POINT("#511);
#513=CARTESIAN_POINT("(-1.5E2,-1.05E1,-5.9E1));
#514=VERTEX_POINT("#513);
#515=CARTESIAN_POINT("(-7.5E1,-4.1E1,-6.2E1));
#516=CARTESIAN_POINT("(-1.15E2,-4.1E1,-6.2E1));
#517=VERTEX_POINT("#515);
#518=VERTEX_POINT("#516);
#519=CARTESIAN_POINT("(-1.15E2,-2.1E1,-6.2E1));
#520=VERTEX_POINT("#519);
#521=CARTESIAN_POINT("(-7.5E1,-2.1E1,-6.2E1));
#522=VERTEX_POINT("#521);
#523=CARTESIAN_POINT("(-7.5E1,-4.1E1,-6.5E1));
#524=CARTESIAN_POINT("(-1.15E2,-4.1E1,-6.5E1));
#525=VERTEX_POINT("#523);
#526=VERTEX_POINT("#524);
#527=CARTESIAN_POINT("(-1.15E2,-2.1E1,-6.5E1));
#528=VERTEX_POINT("#527);
#529=CARTESIAN_POINT("(-7.5E1,-2.1E1,-6.5E1));
#530=VERTEX_POINT("#529);
#531=CARTESIAN_POINT("(-7.E1,-1.5E1,-5.6E1));
#532=CARTESIAN_POINT("(-1.15E2,-1.5E1,-5.6E1));
#533=VERTEX_POINT("#531);
#534=VERTEX_POINT("#532);
#535=CARTESIAN_POINT("(-1.15E2,-5.E0,-5.6E1));
#536=VERTEX_POINT("#535);
#537=CARTESIAN_POINT("(-7.E1,-5.E0,-5.6E1));
#538=VERTEX_POINT("#537);
#539=CARTESIAN_POINT("(-7.E1,-1.5E1,-6.5E1));
#540=CARTESIAN_POINT("(-1.15E2,-1.5E1,-6.5E1));
#541=VERTEX_POINT("#539);

```

```

#542=VERTEX_POINT("#540);
#543=CARTESIAN_POINT("(-1.15E2,-5.E0,-6.5E1));
#544=VERTEX_POINT("#543);
#545=CARTESIAN_POINT("(-7.E1,-5.E0,-6.5E1));
#546=VERTEX_POINT("#545);
#547=CARTESIAN_POINT("(-9.05E1,-3.1E1,-4.2E1));
#548=CARTESIAN_POINT("(-9.95E1,-3.1E1,-4.2E1));
#549=VERTEX_POINT("#547);
#550=VERTEX_POINT("#548);
#551=CARTESIAN_POINT("(-9.05E1,-3.1E1,-6.2E1));
#552=CARTESIAN_POINT("(-9.95E1,-3.1E1,-6.2E1));
#553=VERTEX_POINT("#551);
#554=VERTEX_POINT("#552);
#555=CARTESIAN_POINT("(-2.5E1,-4.6E1,-6.5E1));
#556=CARTESIAN_POINT("(-2.5E1,-4.6E1,-6.E1));
#557=VERTEX_POINT("#555);
#558=VERTEX_POINT("#556);
#559=CARTESIAN_POINT("(-2.5E1,-2.6E1,-6.5E1));
#560=VERTEX_POINT("#559);
#561=CARTESIAN_POINT("(0.E0,-2.6E1,-6.5E1));
#562=VERTEX_POINT("#561);
#563=CARTESIAN_POINT("(0.E0,-2.6E1,-6.E1));
#564=VERTEX_POINT("#563);
#565=CARTESIAN_POINT("(0.E0,-4.6E1,-6.E1));
#566=VERTEX_POINT("#565);
#567=CARTESIAN_POINT("(-2.5E1,-2.6E1,-6.E1));
#568=VERTEX_POINT("#567);
#569=CARTESIAN_POINT("(-3.E1,-2.E1,-5.5E1));
#570=CARTESIAN_POINT("(-3.E1,-6.E0,-5.5E1));
#571=VERTEX_POINT("#569);
#572=VERTEX_POINT("#570);
#573=CARTESIAN_POINT("(0.E0,-2.E1,-6.5E1));
#574=CARTESIAN_POINT("(-3.E1,-2.E1,-6.5E1));
#575=VERTEX_POINT("#573);
#576=VERTEX_POINT("#574);
#577=CARTESIAN_POINT("(-3.E1,-6.E0,-6.5E1));
#578=VERTEX_POINT("#577);
#579=CARTESIAN_POINT("(0.E0,-6.E0,-6.5E1));
#580=VERTEX_POINT("#579);
#581=CARTESIAN_POINT("(0.E0,-2.E1,-5.5E1));
#582=VERTEX_POINT("#581);
#583=CARTESIAN_POINT("(0.E0,-6.E0,-5.5E1));
#584=VERTEX_POINT("#583);
#585=CARTESIAN_POINT("(-8.15E1,0.E0,-3.1E1));
#586=CARTESIAN_POINT("(-6.85E1,0.E0,-3.1E1));
#587=VERTEX_POINT("#585);
#588=VERTEX_POINT("#586);
#589=CARTESIAN_POINT("(-8.15E1,-4.6E1,-3.1E1));
#590=CARTESIAN_POINT("(-6.85E1,-4.6E1,-3.1E1));
#591=VERTEX_POINT("#589);
#592=VERTEX_POINT("#590);
#593=CARTESIAN_POINT("(-9.375E1,0.E0,-3.1E1));
#594=CARTESIAN_POINT("(-8.175E1,0.E0,-3.1E1));
#595=VERTEX_POINT("#593);
#596=VERTEX_POINT("#594);
#597=CARTESIAN_POINT("(-9.375E1,-4.6E1,-3.1E1));
#598=CARTESIAN_POINT("(-8.175E1,-4.6E1,-3.1E1));
#599=VERTEX_POINT("#597);
#600=VERTEX_POINT("#598);
#601=CARTESIAN_POINT("(-1.7E2,0.E0,-3.1E1));

```

```

#602=CARTESIAN_POINT("(-1.6E2,0.E0,-3.1E1));
#603=VERTEX_POINT("#601);
#604=VERTEX_POINT("#602);
#605=CARTESIAN_POINT("(-1.7E2,-4.6E1,-3.1E1));
#606=CARTESIAN_POINT("(-1.6E2,-4.6E1,-3.1E1));
#607=VERTEX_POINT("#605);
#608=VERTEX_POINT("#606);
#609=CARTESIAN_POINT("(0.E0,0.E0,0.E0));
#610=DIRECTION("(0.E0,-1.E0,0.E0));
#611=DIRECTION("(-1.E0,0.E0,0.E0));
#612=AXIS2_PLACEMENT_3D("#609,#610,#611);
#613=PLANE("#612);
#615=ORIENTED_EDGE("*,*,#614,.T.);
#617=ORIENTED_EDGE("*,*,#616,.T.);
#619=ORIENTED_EDGE("*,*,#618,.T.);
#621=ORIENTED_EDGE("*,*,#620,.T.);
#623=ORIENTED_EDGE("*,*,#622,.T.);
#625=ORIENTED_EDGE("*,*,#624,.T.);
#627=ORIENTED_EDGE("*,*,#626,.T.);
#629=ORIENTED_EDGE("*,*,#628,.T.);
#631=ORIENTED_EDGE("*,*,#630,.T.);
#633=ORIENTED_EDGE("*,*,#632,.T.);
#634=EDGE_LOOP("(#615,#617,#619,#621,#623,#625,#627,#629,#631,#633));
#635=FACE_OUTER_BOUND("#634,.F.);
#637=ORIENTED_EDGE("*,*,#636,.T.);
#639=ORIENTED_EDGE("*,*,#638,.T.);
#640=EDGE_LOOP("(#637,#639));
#641=FACE_BOUND("#640,.F.);
#643=ORIENTED_EDGE("*,*,#642,.T.);
#645=ORIENTED_EDGE("*,*,#644,.T.);
#646=EDGE_LOOP("(#643,#645));
#647=FACE_BOUND("#646,.F.);
#649=ORIENTED_EDGE("*,*,#648,.T.);
#651=ORIENTED_EDGE("*,*,#650,.T.);
#652=EDGE_LOOP("(#649,#651));
#653=FACE_BOUND("#652,.F.);
#654=ADVANCED_FACE("(#635,#641,#647,#653),#613,.F.);
#655=CARTESIAN_POINT("(0.E0,0.E0,0.E0));
#656=DIRECTION("(0.E0,0.E0,1.E0));
#657=DIRECTION("(-1.E0,0.E0,0.E0));
#658=AXIS2_PLACEMENT_3D("#655,#656,#657);
#659=PLANE("#658);
#660=ORIENTED_EDGE("*,*,#614,.F.);
#662=ORIENTED_EDGE("*,*,#661,.T.);
#664=ORIENTED_EDGE("*,*,#663,.T.);
#666=ORIENTED_EDGE("*,*,#665,.F.);
#667=EDGE_LOOP("(#660,#662,#664,#666));
#668=FACE_OUTER_BOUND("#667,.F.);
#669=ADVANCED_FACE("(#668),#659,.T.);
#670=CARTESIAN_POINT("(0.E0,0.E0,-6.5E1));
#671=DIRECTION("(1.E0,0.E0,0.E0));
#672=DIRECTION("(0.E0,0.E0,1.E0));
#673=AXIS2_PLACEMENT_3D("#670,#671,#672);
#674=PLANE("#673);
#676=ORIENTED_EDGE("*,*,#675,.F.);
#678=ORIENTED_EDGE("*,*,#677,.T.);
#680=ORIENTED_EDGE("*,*,#679,.T.);
#682=ORIENTED_EDGE("*,*,#681,.T.);
#684=ORIENTED_EDGE("*,*,#683,.T.);
#685=ORIENTED_EDGE("*,*,#661,.F.);

```

```

#686=ORIENTED_EDGE("*,*,#632,.F.);
#688=ORIENTED_EDGE("*,*,#687,.T.);
#690=ORIENTED_EDGE("*,*,#689,.T.);
#692=ORIENTED_EDGE("*,*,#691,.T.);
#693=EDGE_LOOP(",(#676,#678,#680,#682,#684,#685,#686,#688,#690,#692));
#694=FACE_OUTER_BOUND(",#693,.F.);
#695=ADVANCED_FACE(",(#694),#674,.T.);
#696=CARTESIAN_POINT(",(0.E0,-2.E1,-6.5E1));
#697=DIRECTION(",(0.E0,-1.E0,0.E0));
#698=DIRECTION(",(1.E0,0.E0,0.E0));
#699=AXIS2_PLACEMENT_3D(",#696,#697,#698);
#700=PLANE(",#699);
#702=ORIENTED_EDGE("*,*,#701,.F.);
#703=ORIENTED_EDGE("*,*,#675,.T.);
#705=ORIENTED_EDGE("*,*,#704,.T.);
#707=ORIENTED_EDGE("*,*,#706,.F.);
#708=EDGE_LOOP(",(#702,#703,#705,#707));
#709=FACE_OUTER_BOUND(",#708,.F.);
#710=ADVANCED_FACE(",(#709),#700,.F.);
#711=CARTESIAN_POINT(",(4.E1,0.E0,-6.5E1));
#712=DIRECTION(",(0.E0,0.E0,-1.E0));
#713=DIRECTION(",(1.E0,0.E0,0.E0));
#714=AXIS2_PLACEMENT_3D(",#711,#712,#713);
#715=PLANE(",#714);
#716=ORIENTED_EDGE("*,*,#701,.T.);
#718=ORIENTED_EDGE("*,*,#717,.T.);
#720=ORIENTED_EDGE("*,*,#719,.F.);
#721=ORIENTED_EDGE("*,*,#687,.F.);
#722=ORIENTED_EDGE("*,*,#630,.F.);
#724=ORIENTED_EDGE("*,*,#723,.T.);
#726=ORIENTED_EDGE("*,*,#725,.T.);
#728=ORIENTED_EDGE("*,*,#727,.T.);
#730=ORIENTED_EDGE("*,*,#729,.F.);
#731=ORIENTED_EDGE("*,*,#677,.F.);
#732=EDGE_LOOP(",(#716,#718,#720,#721,#722,#724,#726,#728,#730,#731));
#733=FACE_OUTER_BOUND(",#732,.F.);
#734=ADVANCED_FACE(",(#733),#715,.T.);
#735=CARTESIAN_POINT(",(3.E1,-2.E1,-6.5E1));
#736=DIRECTION(",(1.E0,0.E0,0.E0));
#737=DIRECTION(",(0.E0,1.E0,0.E0));
#738=AXIS2_PLACEMENT_3D(",#735,#736,#737);
#739=PLANE(",#738);
#740=ORIENTED_EDGE("*,*,#717,.F.);
#741=ORIENTED_EDGE("*,*,#706,.T.);
#743=ORIENTED_EDGE("*,*,#742,.T.);
#745=ORIENTED_EDGE("*,*,#744,.F.);
#746=EDGE_LOOP(",(#740,#741,#743,#745));
#747=FACE_OUTER_BOUND(",#746,.F.);
#748=ADVANCED_FACE(",(#747),#739,.F.);
#749=CARTESIAN_POINT(",(4.E1,0.E0,-5.5E1));
#750=DIRECTION(",(0.E0,0.E0,-1.E0));
#751=DIRECTION(",(0.E0,1.E0,0.E0));
#752=AXIS2_PLACEMENT_3D(",#749,#750,#751);
#753=PLANE(",#752);
#754=ORIENTED_EDGE("*,*,#691,.F.);
#756=ORIENTED_EDGE("*,*,#755,.F.);
#757=ORIENTED_EDGE("*,*,#742,.F.);
#758=ORIENTED_EDGE("*,*,#704,.F.);
#759=EDGE_LOOP(",(#754,#756,#757,#758));
#760=FACE_OUTER_BOUND(",#759,.F.);

```

```

#761=ADVANCED_FACE(",(#760),#753,.T.);
#762=CARTESIAN_POINT("(-3.E1,-6.E0,-6.5E1));
#763=DIRECTION("(0.E0,1.E0,0.E0));
#764=DIRECTION("(1.E0,0.E0,0.E0));
#765=AXIS2_PLACEMENT_3D(",#762,#763,#764);
#766=PLANE(",#765);
#767=ORIENTED_EDGE("*,*,#719,.T.);
#768=ORIENTED_EDGE("*,*,#744,.T.);
#769=ORIENTED_EDGE("*,*,#755,.T.);
#770=ORIENTED_EDGE("*,*,#689,.F.);
#771=EDGE_LOOP(",(#767,#768,#769,#770));
#772=FACE_OUTER_BOUND(",#771,.F.);
#773=ADVANCED_FACE(",(#772),#766,.F.);
#774=CARTESIAN_POINT("(-4.E1,0.E0,-5.7E1));
#775=DIRECTION("(-1.E0,0.E0,0.E0));
#776=DIRECTION("(0.E0,0.E0,-1.E0));
#777=AXIS2_PLACEMENT_3D(",#774,#775,#776);
#778=PLANE(",#777);
#779=ORIENTED_EDGE("*,*,#628,.F.);
#781=ORIENTED_EDGE("*,*,#780,.T.);
#783=ORIENTED_EDGE("*,*,#782,.T.);
#784=ORIENTED_EDGE("*,*,#723,.F.);
#785=EDGE_LOOP(",(#779,#781,#783,#784));
#786=FACE_OUTER_BOUND(",#785,.F.);
#787=ADVANCED_FACE(",(#786),#778,.T.);
#788=CARTESIAN_POINT("(-6.E1,0.E0,-5.7E1));
#789=DIRECTION("(0.E0,0.E0,-1.E0));
#790=DIRECTION("(1.E0,0.E0,0.E0));
#791=AXIS2_PLACEMENT_3D(",#788,#789,#790);
#792=PLANE(",#791);
#793=ORIENTED_EDGE("*,*,#626,.F.);
#795=ORIENTED_EDGE("*,*,#794,.T.);
#797=ORIENTED_EDGE("*,*,#796,.T.);
#798=ORIENTED_EDGE("*,*,#780,.F.);
#799=EDGE_LOOP(",(#793,#795,#797,#798));
#800=FACE_OUTER_BOUND(",#799,.F.);
#801=ADVANCED_FACE(",(#800),#792,.T.);
#802=CARTESIAN_POINT("(-6.E1,0.E0,-6.5E1));
#803=DIRECTION("(1.E0,0.E0,0.E0));
#804=DIRECTION("(0.E0,0.E0,1.E0));
#805=AXIS2_PLACEMENT_3D(",#802,#803,#804);
#806=PLANE(",#805);
#807=ORIENTED_EDGE("*,*,#624,.F.);
#809=ORIENTED_EDGE("*,*,#808,.T.);
#811=ORIENTED_EDGE("*,*,#810,.T.);
#812=ORIENTED_EDGE("*,*,#794,.F.);
#813=EDGE_LOOP(",(#807,#809,#811,#812));
#814=FACE_OUTER_BOUND(",#813,.F.);
#815=ADVANCED_FACE(",(#814),#806,.T.);
#816=CARTESIAN_POINT("(-1.3E2,0.E0,-6.5E1));
#817=DIRECTION("(0.E0,0.E0,-1.E0));
#818=DIRECTION("(1.E0,0.E0,0.E0));
#819=AXIS2_PLACEMENT_3D(",#816,#817,#818);
#820=PLANE(",#819);
#821=ORIENTED_EDGE("*,*,#622,.F.);
#823=ORIENTED_EDGE("*,*,#822,.T.);
#825=ORIENTED_EDGE("*,*,#824,.T.);
#826=ORIENTED_EDGE("*,*,#808,.F.);
#827=EDGE_LOOP(",(#821,#823,#825,#826));
#828=FACE_OUTER_BOUND(",#827,.F.);

```

```

#830=ORIENTED_EDGE("*,*,#829,.T.);
#832=ORIENTED_EDGE("*,*,#831,.T.);
#834=ORIENTED_EDGE("*,*,#833,.T.);
#836=ORIENTED_EDGE("*,*,#835,.T.);
#837=EDGE_LOOP(",(#830,#832,#834,#836));
#838=FACE_BOUND(",#837,.F.);
#840=ORIENTED_EDGE("*,*,#839,.T.);
#842=ORIENTED_EDGE("*,*,#841,.T.);
#844=ORIENTED_EDGE("*,*,#843,.T.);
#846=ORIENTED_EDGE("*,*,#845,.T.);
#847=EDGE_LOOP(",(#840,#842,#844,#846));
#848=FACE_BOUND(",#847,.F.);
#849=ADVANCED_FACE(",(#828,#838,#848),#820,.T.);
#850=CARTESIAN_POINT(",( -1.3E2,0.E0,-5.9E1));
#851=DIRECTION(",( -1.E0,0.E0,0.E0));
#852=DIRECTION(",(0.E0,0.E0,-1.E0));
#853=AXIS2_PLACEMENT_3D(",#850,#851,#852);
#854=PLANE(",#853);
#855=ORIENTED_EDGE("*,*,#620,.F.);
#857=ORIENTED_EDGE("*,*,#856,.T.);
#859=ORIENTED_EDGE("*,*,#858,.T.);
#860=ORIENTED_EDGE("*,*,#822,.F.);
#861=EDGE_LOOP(",(#855,#857,#859,#860));
#862=FACE_OUTER_BOUND(",#861,.F.);
#863=ADVANCED_FACE(",(#862),#854,.T.);
#864=CARTESIAN_POINT(",( -2.E2,0.E0,-5.9E1));
#865=DIRECTION(",(0.E0,0.E0,-1.E0));
#866=DIRECTION(",(1.E0,0.E0,0.E0));
#867=AXIS2_PLACEMENT_3D(",#864,#865,#866);
#868=PLANE(",#867);
#870=ORIENTED_EDGE("*,*,#869,.F.);
#872=ORIENTED_EDGE("*,*,#871,.T.);
#874=ORIENTED_EDGE("*,*,#873,.T.);
#875=ORIENTED_EDGE("*,*,#856,.F.);
#876=ORIENTED_EDGE("*,*,#618,.F.);
#878=ORIENTED_EDGE("*,*,#877,.T.);
#880=ORIENTED_EDGE("*,*,#879,.T.);
#882=ORIENTED_EDGE("*,*,#881,.T.);
#883=EDGE_LOOP(",(#870,#872,#874,#875,#876,#878,#880,#882));
#884=FACE_OUTER_BOUND(",#883,.F.);
#885=ADVANCED_FACE(",(#884),#868,.T.);
#886=CARTESIAN_POINT(",( -2.E2,-3.55E1,-5.9E1));
#887=DIRECTION(",(0.E0,-1.E0,0.E0));
#888=DIRECTION(",(0.E0,0.E0,1.E0));
#889=AXIS2_PLACEMENT_3D(",#886,#887,#888);
#890=PLANE(",#889);
#892=ORIENTED_EDGE("*,*,#891,.F.);
#893=ORIENTED_EDGE("*,*,#869,.T.);
#895=ORIENTED_EDGE("*,*,#894,.T.);
#897=ORIENTED_EDGE("*,*,#896,.F.);
#898=EDGE_LOOP(",(#892,#893,#895,#897));
#899=FACE_OUTER_BOUND(",#898,.F.);
#900=ADVANCED_FACE(",(#899),#890,.F.);
#901=CARTESIAN_POINT(",( -2.E2,0.E0,0.E0));
#902=DIRECTION(",( -1.E0,0.E0,0.E0));
#903=DIRECTION(",(0.E0,0.E0,-1.E0));
#904=AXIS2_PLACEMENT_3D(",#901,#902,#903);
#905=PLANE(",#904);
#906=ORIENTED_EDGE("*,*,#891,.T.);
#908=ORIENTED_EDGE("*,*,#907,.T.);

```

```

#910=ORIENTED_EDGE("*,*,#909,.F.);
#911=ORIENTED_EDGE("*,*,#877,.F.);
#912=ORIENTED_EDGE("*,*,#616,.F.);
#913=ORIENTED_EDGE("*,*,#665,.T.);
#915=ORIENTED_EDGE("*,*,#914,.T.);
#916=ORIENTED_EDGE("*,*,#871,.F.);
#917=EDGE_LOOP("(#906,#908,#910,#911,#912,#913,#915,#916));
#918=FACE_OUTER_BOUND(",#917,.F.);
#919=ADVANCED_FACE("(#918),#905,.T.);
#920=CARTESIAN_POINT("(-2.E2,-3.55E1,-4.9E1));
#921=DIRECTION("(0.E0,0.E0,1.E0));
#922=DIRECTION("(0.E0,1.E0,0.E0));
#923=AXIS2_PLACEMENT_3D(",#920,#921,#922);
#924=PLANE(",#923);
#925=ORIENTED_EDGE("*,*,#907,.F.);
#926=ORIENTED_EDGE("*,*,#896,.T.);
#928=ORIENTED_EDGE("*,*,#927,.T.);
#930=ORIENTED_EDGE("*,*,#929,.F.);
#931=EDGE_LOOP("(#925,#926,#928,#930));
#932=FACE_OUTER_BOUND(",#931,.F.);
#933=ADVANCED_FACE("(#932),#924,.F.);
#934=CARTESIAN_POINT("(-1.5E2,0.E0,0.E0));
#935=DIRECTION("(-1.E0,0.E0,0.E0));
#936=DIRECTION("(0.E0,1.E0,0.E0));
#937=AXIS2_PLACEMENT_3D(",#934,#935,#936);
#938=PLANE(",#937);
#939=ORIENTED_EDGE("*,*,#881,.F.);
#941=ORIENTED_EDGE("*,*,#940,.F.);
#942=ORIENTED_EDGE("*,*,#927,.F.);
#943=ORIENTED_EDGE("*,*,#894,.F.);
#944=EDGE_LOOP("(#939,#941,#942,#943));
#945=FACE_OUTER_BOUND(",#944,.F.);
#946=ADVANCED_FACE("(#945),#938,.T.);
#947=CARTESIAN_POINT("(-2.E2,-1.05E1,-4.9E1));
#948=DIRECTION("(0.E0,1.E0,0.E0));
#949=DIRECTION("(0.E0,0.E0,-1.E0));
#950=AXIS2_PLACEMENT_3D(",#947,#948,#949);
#951=PLANE(",#950);
#952=ORIENTED_EDGE("*,*,#909,.T.);
#953=ORIENTED_EDGE("*,*,#929,.T.);
#954=ORIENTED_EDGE("*,*,#940,.T.);
#955=ORIENTED_EDGE("*,*,#879,.F.);
#956=EDGE_LOOP("(#952,#953,#954,#955));
#957=FACE_OUTER_BOUND(",#956,.F.);
#958=ADVANCED_FACE("(#957),#951,.F.);
#959=CARTESIAN_POINT("(0.E0,-4.6E1,0.E0));
#960=DIRECTION("(0.E0,-1.E0,0.E0));
#961=DIRECTION("(-1.E0,0.E0,0.E0));
#962=AXIS2_PLACEMENT_3D(",#959,#960,#961);
#963=PLANE(",#962);
#965=ORIENTED_EDGE("*,*,#964,.F.);
#966=ORIENTED_EDGE("*,*,#725,.F.);
#967=ORIENTED_EDGE("*,*,#782,.F.);
#968=ORIENTED_EDGE("*,*,#796,.F.);
#969=ORIENTED_EDGE("*,*,#810,.F.);
#970=ORIENTED_EDGE("*,*,#824,.F.);
#971=ORIENTED_EDGE("*,*,#858,.F.);
#972=ORIENTED_EDGE("*,*,#873,.F.);
#973=ORIENTED_EDGE("*,*,#914,.F.);
#974=ORIENTED_EDGE("*,*,#663,.F.);

```

```

#975=ORIENTED_EDGE("*,*,#683,.F.);
#977=ORIENTED_EDGE("*,*,#976,.T.);
#978=EDGE_LOOP(",(#965,#966,#967,#968,#969,#970,#971,#972,#973,#974,#975,
#977));
#979=FACE_OUTER_BOUND(",#978,.F.);
#981=ORIENTED_EDGE("*,*,#980,.F.);
#983=ORIENTED_EDGE("*,*,#982,.F.);
#984=EDGE_LOOP(",(#981,#983));
#985=FACE_BOUND(",#984,.F.);
#987=ORIENTED_EDGE("*,*,#986,.F.);
#989=ORIENTED_EDGE("*,*,#988,.F.);
#990=EDGE_LOOP(",(#987,#989));
#991=FACE_BOUND(",#990,.F.);
#993=ORIENTED_EDGE("*,*,#992,.F.);
#995=ORIENTED_EDGE("*,*,#994,.F.);
#996=EDGE_LOOP(",(#993,#995));
#997=FACE_BOUND(",#996,.F.);
#998=ADVANCED_FACE(",(#979,#985,#991,#997),#963,.T.);
#999=CARTESIAN_POINT(",-2.5E1,-4.6E1,-6.5E1));
#1000=DIRECTION(",-1.E0,0.E0,0.E0));
#1001=DIRECTION(",(0.E0,1.E0,0.E0));
#1002=AXIS2_PLACEMENT_3D(",#999,#1000,#1001);
#1003=PLANE(",#1002);
#1004=ORIENTED_EDGE("*,*,#964,.T.);
#1006=ORIENTED_EDGE("*,*,#1005,.T.);
#1008=ORIENTED_EDGE("*,*,#1007,.F.);
#1009=ORIENTED_EDGE("*,*,#727,.F.);
#1010=EDGE_LOOP(",(#1004,#1006,#1008,#1009));
#1011=FACE_OUTER_BOUND(",#1010,.F.);
#1012=ADVANCED_FACE(",(#1011),#1003,.F.);
#1013=CARTESIAN_POINT(",-4.E1,0.E0,-6.E1));
#1014=DIRECTION(",(0.E0,0.E0,-1.E0));
#1015=DIRECTION(",(0.E0,1.E0,0.E0));
#1016=AXIS2_PLACEMENT_3D(",#1013,#1014,#1015);
#1017=PLANE(",#1016);
#1018=ORIENTED_EDGE("*,*,#976,.F.);
#1019=ORIENTED_EDGE("*,*,#681,.F.);
#1021=ORIENTED_EDGE("*,*,#1020,.F.);
#1022=ORIENTED_EDGE("*,*,#1005,.F.);
#1023=EDGE_LOOP(",(#1018,#1019,#1021,#1022));
#1024=FACE_OUTER_BOUND(",#1023,.F.);
#1025=ADVANCED_FACE(",(#1024),#1017,.T.);
#1026=CARTESIAN_POINT(",-2.5E1,-2.6E1,-6.5E1));
#1027=DIRECTION(",(0.E0,1.E0,0.E0));
#1028=DIRECTION(",(1.E0,0.E0,0.E0));
#1029=AXIS2_PLACEMENT_3D(",#1026,#1027,#1028);
#1030=PLANE(",#1029);
#1031=ORIENTED_EDGE("*,*,#729,.T.);
#1032=ORIENTED_EDGE("*,*,#1007,.T.);
#1033=ORIENTED_EDGE("*,*,#1020,.T.);
#1034=ORIENTED_EDGE("*,*,#679,.F.);
#1035=EDGE_LOOP(",(#1031,#1032,#1033,#1034));
#1036=FACE_OUTER_BOUND(",#1035,.F.);
#1037=ADVANCED_FACE(",(#1036),#1030,.F.);
#1038=CARTESIAN_POINT(",-7.5E1,-4.6E1,-3.1E1));
#1039=DIRECTION(",(0.E0,1.E0,0.E0));
#1040=DIRECTION(",-1.E0,0.E0,0.E0));
#1041=AXIS2_PLACEMENT_3D(",#1038,#1039,#1040);
#1042=CYLINDRICAL_SURFACE(",#1041,6.5E0);
#1043=ORIENTED_EDGE("*,*,#636,.F.);

```

```

#1045=ORIENTED_EDGE("*,*,#1044,.F.);
#1046=ORIENTED_EDGE("*,*,#980,.T.);
#1048=ORIENTED_EDGE("*,*,#1047,.T.);
#1049=EDGE_LOOP(",(#1043,#1045,#1046,#1048));
#1050=FACE_OUTER_BOUND(",(#1049,.F.);
#1051=ADVANCED_FACE(",(#1050),#1042,.F.);
#1052=CARTESIAN_POINT(",-7.5E1,-4.6E1,-3.1E1));
#1053=DIRECTION(",(0.E0,1.E0,0.E0));
#1054=DIRECTION(",-1.E0,0.E0,0.E0));
#1055=AXIS2_PLACEMENT_3D(",(#1052,#1053,#1054);
#1056=CYLINDRICAL_SURFACE(",(#1055,6.5E0);
#1057=ORIENTED_EDGE("*,*,#638,.F.);
#1058=ORIENTED_EDGE("*,*,#1047,.F.);
#1059=ORIENTED_EDGE("*,*,#982,.T.);
#1060=ORIENTED_EDGE("*,*,#1044,.T.);
#1061=EDGE_LOOP(",(#1057,#1058,#1059,#1060));
#1062=FACE_OUTER_BOUND(",(#1061,.F.);
#1063=ADVANCED_FACE(",(#1062),#1056,.F.);
#1064=CARTESIAN_POINT(",-8.775E1,-4.6E1,-3.1E1));
#1065=DIRECTION(",(0.E0,1.E0,0.E0));
#1066=DIRECTION(",-1.E0,0.E0,0.E0));
#1067=AXIS2_PLACEMENT_3D(",(#1064,#1065,#1066);
#1068=CYLINDRICAL_SURFACE(",(#1067,6.E0);
#1069=ORIENTED_EDGE("*,*,#642,.F.);
#1071=ORIENTED_EDGE("*,*,#1070,.F.);
#1072=ORIENTED_EDGE("*,*,#986,.T.);
#1074=ORIENTED_EDGE("*,*,#1073,.T.);
#1075=EDGE_LOOP(",(#1069,#1071,#1072,#1074));
#1076=FACE_OUTER_BOUND(",(#1075,.F.);
#1077=ADVANCED_FACE(",(#1076),#1068,.F.);
#1078=CARTESIAN_POINT(",-8.775E1,-4.6E1,-3.1E1));
#1079=DIRECTION(",(0.E0,1.E0,0.E0));
#1080=DIRECTION(",-1.E0,0.E0,0.E0));
#1081=AXIS2_PLACEMENT_3D(",(#1078,#1079,#1080);
#1082=CYLINDRICAL_SURFACE(",(#1081,6.E0);
#1083=ORIENTED_EDGE("*,*,#644,.F.);
#1084=ORIENTED_EDGE("*,*,#1073,.F.);
#1085=ORIENTED_EDGE("*,*,#988,.T.);
#1086=ORIENTED_EDGE("*,*,#1070,.T.);
#1087=EDGE_LOOP(",(#1083,#1084,#1085,#1086));
#1088=FACE_OUTER_BOUND(",(#1087,.F.);
#1089=ADVANCED_FACE(",(#1088),#1082,.F.);
#1090=CARTESIAN_POINT(",-1.65E2,-4.6E1,-3.1E1));
#1091=DIRECTION(",(0.E0,1.E0,0.E0));
#1092=DIRECTION(",-1.E0,0.E0,0.E0));
#1093=AXIS2_PLACEMENT_3D(",(#1090,#1091,#1092);
#1094=CYLINDRICAL_SURFACE(",(#1093,5.E0);
#1095=ORIENTED_EDGE("*,*,#648,.F.);
#1097=ORIENTED_EDGE("*,*,#1096,.F.);
#1098=ORIENTED_EDGE("*,*,#992,.T.);
#1100=ORIENTED_EDGE("*,*,#1099,.T.);
#1101=EDGE_LOOP(",(#1095,#1097,#1098,#1100));
#1102=FACE_OUTER_BOUND(",(#1101,.F.);
#1103=ADVANCED_FACE(",(#1102),#1094,.F.);
#1104=CARTESIAN_POINT(",-1.65E2,-4.6E1,-3.1E1));
#1105=DIRECTION(",(0.E0,1.E0,0.E0));
#1106=DIRECTION(",-1.E0,0.E0,0.E0));
#1107=AXIS2_PLACEMENT_3D(",(#1104,#1105,#1106);
#1108=CYLINDRICAL_SURFACE(",(#1107,5.E0);
#1109=ORIENTED_EDGE("*,*,#650,.F.);

```

```

#1110=ORIENTED_EDGE("*,*,#1099,.F.);
#1111=ORIENTED_EDGE("*,*,#994,.T.);
#1112=ORIENTED_EDGE("*,*,#1096,.T.);
#1113=EDGE_LOOP(",(#1109,#1110,#1111,#1112));
#1114=FACE_OUTER_BOUND(",(#1113,.F.);
#1115=ADVANCED_FACE(",(#1114),#1108,.F.);
#1116=CARTESIAN_POINT(",( -7.5E1, -4.1E1, -6.5E1));
#1117=DIRECTION(",(0.E0, -1.E0, 0.E0));
#1118=DIRECTION(",( -1.E0, 0.E0, 0.E0));
#1119=AXIS2_PLACEMENT_3D(",(#1116,#1117,#1118);
#1120=PLANE(",(#1119);
#1121=ORIENTED_EDGE("*,*,#829,.F.);
#1123=ORIENTED_EDGE("*,*,#1122,.T.);
#1125=ORIENTED_EDGE("*,*,#1124,.T.);
#1127=ORIENTED_EDGE("*,*,#1126,.F.);
#1128=EDGE_LOOP(",(#1121,#1123,#1125,#1127));
#1129=FACE_OUTER_BOUND(",(#1128,.F.);
#1130=ADVANCED_FACE(",(#1129),#1120,.F.);
#1131=CARTESIAN_POINT(",( -7.5E1, -2.1E1, -6.5E1));
#1132=DIRECTION(",(1.E0, 0.E0, 0.E0));
#1133=DIRECTION(",(0.E0, -1.E0, 0.E0));
#1134=AXIS2_PLACEMENT_3D(",(#1131,#1132,#1133);
#1135=PLANE(",(#1134);
#1136=ORIENTED_EDGE("*,*,#835,.F.);
#1138=ORIENTED_EDGE("*,*,#1137,.T.);
#1140=ORIENTED_EDGE("*,*,#1139,.T.);
#1141=ORIENTED_EDGE("*,*,#1122,.F.);
#1142=EDGE_LOOP(",(#1136,#1138,#1140,#1141));
#1143=FACE_OUTER_BOUND(",(#1142,.F.);
#1144=ADVANCED_FACE(",(#1143),#1135,.F.);
#1145=CARTESIAN_POINT(",( -1.15E2, -2.1E1, -6.5E1));
#1146=DIRECTION(",(0.E0, 1.E0, 0.E0));
#1147=DIRECTION(",(1.E0, 0.E0, 0.E0));
#1148=AXIS2_PLACEMENT_3D(",(#1145,#1146,#1147);
#1149=PLANE(",(#1148);
#1150=ORIENTED_EDGE("*,*,#833,.F.);
#1152=ORIENTED_EDGE("*,*,#1151,.T.);
#1154=ORIENTED_EDGE("*,*,#1153,.T.);
#1155=ORIENTED_EDGE("*,*,#1137,.F.);
#1156=EDGE_LOOP(",(#1150,#1152,#1154,#1155));
#1157=FACE_OUTER_BOUND(",(#1156,.F.);
#1158=ADVANCED_FACE(",(#1157),#1149,.F.);
#1159=CARTESIAN_POINT(",( -1.15E2, -4.1E1, -6.5E1));
#1160=DIRECTION(",( -1.E0, 0.E0, 0.E0));
#1161=DIRECTION(",(0.E0, 1.E0, 0.E0));
#1162=AXIS2_PLACEMENT_3D(",(#1159,#1160,#1161);
#1163=PLANE(",(#1162);
#1164=ORIENTED_EDGE("*,*,#831,.F.);
#1165=ORIENTED_EDGE("*,*,#1126,.T.);
#1167=ORIENTED_EDGE("*,*,#1166,.T.);
#1168=ORIENTED_EDGE("*,*,#1151,.F.);
#1169=EDGE_LOOP(",(#1164,#1165,#1167,#1168));
#1170=FACE_OUTER_BOUND(",(#1169,.F.);
#1171=ADVANCED_FACE(",(#1170),#1163,.F.);
#1172=CARTESIAN_POINT(",( -1.3E2, 0.E0, -6.2E1));
#1173=DIRECTION(",(0.E0, 0.E0, -1.E0));
#1174=DIRECTION(",(0.E0, 1.E0, 0.E0));
#1175=AXIS2_PLACEMENT_3D(",(#1172,#1173,#1174);
#1176=PLANE(",(#1175);
#1177=ORIENTED_EDGE("*,*,#1124,.F.);

```

T.C. YÜKSEKÖĞRETİM KURULU
 DOKÜMANTASYON BİRİMİ

```

#1178=ORIENTED_EDGE("*,*,#1139,.F.);
#1179=ORIENTED_EDGE("*,*,#1153,.F.);
#1180=ORIENTED_EDGE("*,*,#1166,.F.);
#1181=EDGE_LOOP(",(#1177,#1178,#1179,#1180));
#1182=FACE_OUTER_BOUND(",(#1181,.F.);
#1184=ORIENTED_EDGE("*,*,#1183,.F.);
#1186=ORIENTED_EDGE("*,*,#1185,.F.);
#1187=EDGE_LOOP(",(#1184,#1186));
#1188=FACE_BOUND(",(#1187,.F.);
#1189=ADVANCED_FACE(",(#1182,#1188),#1176,.T.);
#1190=CARTESIAN_POINT(",( -9.5E1, -3.1E1, -6.2E1));
#1191=DIRECTION(",(0.E0,0.E0,1.E0));
#1192=DIRECTION(",(1.E0,0.E0,0.E0));
#1193=AXIS2_PLACEMENT_3D(",(#1190,#1191,#1192);
#1194=CYLINDRICAL_SURFACE(",(#1193,4.5E0);
#1195=ORIENTED_EDGE("*,*,#1183,.T.);
#1197=ORIENTED_EDGE("*,*,#1196,.T.);
#1199=ORIENTED_EDGE("*,*,#1198,.F.);
#1201=ORIENTED_EDGE("*,*,#1200,.F.);
#1202=EDGE_LOOP(",(#1195,#1197,#1199,#1201));
#1203=FACE_OUTER_BOUND(",(#1202,.F.);
#1204=ADVANCED_FACE(",(#1203),#1194,.F.);
#1205=CARTESIAN_POINT(",( -9.5E1, -3.1E1, -6.2E1));
#1206=DIRECTION(",(0.E0,0.E0,1.E0));
#1207=DIRECTION(",(1.E0,0.E0,0.E0));
#1208=AXIS2_PLACEMENT_3D(",(#1205,#1206,#1207);
#1209=CYLINDRICAL_SURFACE(",(#1208,4.5E0);
#1210=ORIENTED_EDGE("*,*,#1185,.T.);
#1211=ORIENTED_EDGE("*,*,#1200,.T.);
#1213=ORIENTED_EDGE("*,*,#1212,.F.);
#1214=ORIENTED_EDGE("*,*,#1196,.F.);
#1215=EDGE_LOOP(",(#1210,#1211,#1213,#1214));
#1216=FACE_OUTER_BOUND(",(#1215,.F.);
#1217=ADVANCED_FACE(",(#1216),#1209,.F.);
#1218=CARTESIAN_POINT(",( -9.5E1, -3.1E1, -4.2E1));
#1219=DIRECTION(",(0.E0,0.E0,1.E0));
#1220=DIRECTION(",(1.E0,0.E0,0.E0));
#1221=AXIS2_PLACEMENT_3D(",(#1218,#1219,#1220);
#1222=PLANE(",(#1221);
#1223=ORIENTED_EDGE("*,*,#1198,.T.);
#1224=ORIENTED_EDGE("*,*,#1212,.T.);
#1225=EDGE_LOOP(",(#1223,#1224));
#1226=FACE_OUTER_BOUND(",(#1225,.F.);
#1227=ADVANCED_FACE(",(#1226),#1222,.F.);
#1228=CARTESIAN_POINT(",( -7.E1, -1.5E1, -6.5E1));
#1229=DIRECTION(",(0.E0,-1.E0,0.E0));
#1230=DIRECTION(",( -1.E0,0.E0,0.E0));
#1231=AXIS2_PLACEMENT_3D(",(#1228,#1229,#1230);
#1232=PLANE(",(#1231);
#1233=ORIENTED_EDGE("*,*,#839,.F.);
#1235=ORIENTED_EDGE("*,*,#1234,.T.);
#1237=ORIENTED_EDGE("*,*,#1236,.T.);
#1239=ORIENTED_EDGE("*,*,#1238,.F.);
#1240=EDGE_LOOP(",(#1233,#1235,#1237,#1239));
#1241=FACE_OUTER_BOUND(",(#1240,.F.);
#1242=ADVANCED_FACE(",(#1241),#1232,.F.);
#1243=CARTESIAN_POINT(",( -7.E1, -5.E0, -6.5E1));
#1244=DIRECTION(",(1.E0,0.E0,0.E0));
#1245=DIRECTION(",(0.E0,-1.E0,0.E0));
#1246=AXIS2_PLACEMENT_3D(",(#1243,#1244,#1245);

```

```

#1247=PLANE(",#1246);
#1248=ORIENTED_EDGE("*,*,#845,.F.);
#1250=ORIENTED_EDGE("*,*,#1249,.T.);
#1252=ORIENTED_EDGE("*,*,#1251,.T.);
#1253=ORIENTED_EDGE("*,*,#1234,.F.);
#1254=EDGE_LOOP(",(#1248,#1250,#1252,#1253));
#1255=FACE_OUTER_BOUND(",#1254,.F.);
#1256=ADVANCED_FACE(",(#1255),#1247,.F.);
#1257=CARTESIAN_POINT(",-1.15E2,-5.E0,-6.5E1));
#1258=DIRECTION(",(0.E0,1.E0,0.E0));
#1259=DIRECTION(",(1.E0,0.E0,0.E0));
#1260=AXIS2_PLACEMENT_3D(",#1257,#1258,#1259);
#1261=PLANE(",#1260);
#1262=ORIENTED_EDGE("*,*,#843,.F.);
#1264=ORIENTED_EDGE("*,*,#1263,.T.);
#1266=ORIENTED_EDGE("*,*,#1265,.T.);
#1267=ORIENTED_EDGE("*,*,#1249,.F.);
#1268=EDGE_LOOP(",(#1262,#1264,#1266,#1267));
#1269=FACE_OUTER_BOUND(",#1268,.F.);
#1270=ADVANCED_FACE(",(#1269),#1261,.F.);
#1271=CARTESIAN_POINT(",-1.15E2,-1.5E1,-6.5E1));
#1272=DIRECTION(",-1.E0,0.E0,0.E0));
#1273=DIRECTION(",(0.E0,1.E0,0.E0));
#1274=AXIS2_PLACEMENT_3D(",#1271,#1272,#1273);
#1275=PLANE(",#1274);
#1276=ORIENTED_EDGE("*,*,#841,.F.);
#1277=ORIENTED_EDGE("*,*,#1238,.T.);
#1279=ORIENTED_EDGE("*,*,#1278,.T.);
#1280=ORIENTED_EDGE("*,*,#1263,.F.);
#1281=EDGE_LOOP(",(#1276,#1277,#1279,#1280));
#1282=FACE_OUTER_BOUND(",#1281,.F.);
#1283=ADVANCED_FACE(",(#1282),#1275,.F.);
#1284=CARTESIAN_POINT(",-1.3E2,0.E0,-5.6E1));
#1285=DIRECTION(",(0.E0,0.E0,-1.E0));
#1286=DIRECTION(",(0.E0,1.E0,0.E0));
#1287=AXIS2_PLACEMENT_3D(",#1284,#1285,#1286);
#1288=PLANE(",#1287);
#1289=ORIENTED_EDGE("*,*,#1236,.F.);
#1290=ORIENTED_EDGE("*,*,#1251,.F.);
#1291=ORIENTED_EDGE("*,*,#1265,.F.);
#1292=ORIENTED_EDGE("*,*,#1278,.F.);
#1293=EDGE_LOOP(",(#1289,#1290,#1291,#1292));
#1294=FACE_OUTER_BOUND(",#1293,.F.);
#1295=ADVANCED_FACE(",(#1294),#1288,.T.);
#1296=CLOSED_SHELL(",(#654,#669,#695,#710,#734,#748,#761,#773,#787,#801,#815,
#849,#863,#885,#900,#919,#933,#946,#958,#998,#1012,#1025,#1037,#1051,#1063,
#1077,#1089,#1103,#1115,#1130,#1144,#1158,#1171,#1189,#1204,#1217,#1227,#1242,
#1256,#1270,#1283,#1295));
#1297=MANIFOLD_SOLID_BREP(",#1296);
#1298=DIMENSIONAL_EXPONENTS(1.E0,0.E0,0.E0,0.E0,0.E0,0.E0,0.E0);
#1299=(LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.MILLI.,METRE.));
#1300=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(2.54E1),#1299);
#1301=(CONVERSION_BASED_UNIT('INCH',#1300)LENGTH_UNIT()NAMED_UNIT(#1298));
#1302=DIMENSIONAL_EXPONENTS(0.E0,0.E0,0.E0,0.E0,0.E0,0.E0,0.E0);
#1303=(NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
#1304=PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(1.74532925199
4E-2),
#1303);
#1305=(CONVERSION_BASED_UNIT('DEGREE',#1304)NAMED_UNIT(#1302)PLANE_ANGLE
_UNIT

```

```

0);
#1306=(NAMED_UNIT(*)SI_UNIT($,.STERADIAN.)SOLID_ANGLE_UNIT());
#1307=UNCERTAINTY_MEASURE_WITH_UNIT(LENGTH_MEASURE(2.152609876270E-
2),#1301,
'closure',
'Maximum model space distance between geometric entities at asserted connectivities');
#1308=(GEOMETRIC_REPRESENTATION_CONTEXT(3)GLOBAL_UNCERTAINTY_ASSIGN
ED_CONTEXT(
(#1307))GLOBAL_UNIT_ASSIGNED_CONTEXT((#1301,#1305,#1306))REPRESENTATION_CO
NTEXT
('ID1','3'));
#1310=APPLICATION_CONTEXT
('CONFIGURATION CONTROLLED 3D DESIGNS OF MECHANICAL PARTS AND
ASSEMBLIES');
#1311=APPLICATION_PROTOCOL_DEFINITION('international standard',
'config_control_design',1994,#1310);
#1312=DESIGN_CONTEXT("#1310,'design');
#1313=MECHANICAL_CONTEXT("#1310,'mechanical');
#1314=PRODUCT('TURKAY-TEZ2','TURKAY-TEZ2','NOT SPECIFIED',(#1313));
#1315=PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE("LAST_VERSIO
N",
#1314,.MADE.);
#1319=PRODUCT_CATEGORY('part',"");
#1320=PRODUCT_RELATED_PRODUCT_CATEGORY('detail',"",(#1314));
#1321=PRODUCT_CATEGORY_RELATIONSHIP("","#1319,#1320);
#1322=SECURITY_CLASSIFICATION_LEVEL('unclassified');
#1323=SECURITY_CLASSIFICATION("","#1322);
#1324=CC_DESIGN_SECURITY_CLASSIFICATION(#1323,(#1315));
#1325=APPROVAL_STATUS('approved');
#1326=APPROVAL(#1325,"");
#1327=CC_DESIGN_APPROVAL(#1326,(#1323,#1315,#1316));
#1328=CALENDAR_DATE(97,25,12);
#1329=COORDINATED_UNIVERSAL_TIME_OFFSET(8,0,BEHIND.);
#1330=LOCAL_TIME(14,26,5.7E1,#1329);
#1331=DATE_AND_TIME(#1328,#1330);
#1332=APPROVAL_DATE_TIME(#1331,#1326);
#1333=DATE_TIME_ROLE('creation_date');
#1334=CC_DESIGN_DATE_AND_TIME_ASSIGNMENT(#1331,#1333,(#1316));
#1335=DATE_TIME_ROLE('classification_date');
#1336=CC_DESIGN_DATE_AND_TIME_ASSIGNMENT(#1331,#1335,(#1323));
#1337=PERSON('UNSPECIFIED','UNSPECIFIED',$,$,$,$);
#1338=ORGANIZATION('UNSPECIFIED','UNSPECIFIED','UNSPECIFIED');
#1339=PERSON_AND_ORGANIZATION(#1337,#1338);
#1340=APPROVAL_ROLE('approver');
#1341=APPROVAL_PERSON_ORGANIZATION(#1339,#1326,#1340);
#1342=PERSON_AND_ORGANIZATION_ROLE('creator');
#1343=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#1339,#1342,(#1315,#13
16));
#1344=PERSON_AND_ORGANIZATION_ROLE('design_supplier');
#1345=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#1339,#1344,(#1315));
#1346=PERSON_AND_ORGANIZATION_ROLE('classification_officer');
#1347=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#1339,#1346,(#1323));
#1348=PERSON_AND_ORGANIZATION_ROLE('design_owner');
#1349=CC_DESIGN_PERSON_AND_ORGANIZATION_ASSIGNMENT(#1339,#1348,(#1314));
#45=CIRCLE("#44,6.5E0);
#50=CIRCLE("#49,6.5E0);
#55=CIRCLE("#54,6.E0);
#60=CIRCLE("#59,6.E0);
#65=CIRCLE("#64,5.E0);
#70=CIRCLE("#69,5.E0);

```

```

#307=CIRCLE(" ,#306,6.5E0);
#312=CIRCLE(" ,#311,6.5E0);
#317=CIRCLE(" ,#316,6.E0);
#322=CIRCLE(" ,#321,6.E0);
#327=CIRCLE(" ,#326,5.E0);
#332=CIRCLE(" ,#331,5.E0);
#405=CIRCLE(" ,#404,4.5E0);
#410=CIRCLE(" ,#409,4.5E0);
#423=CIRCLE(" ,#422,4.5E0);
#428=CIRCLE(" ,#427,4.5E0);
#614=EDGE_CURVE(" ,#463,#464,#4,.T.);
#616=EDGE_CURVE(" ,#464,#466,#8,.T.);
#618=EDGE_CURVE(" ,#466,#468,#12,.T.);
#620=EDGE_CURVE(" ,#468,#470,#16,.T.);
#622=EDGE_CURVE(" ,#470,#472,#20,.T.);
#624=EDGE_CURVE(" ,#472,#474,#24,.T.);
#626=EDGE_CURVE(" ,#474,#476,#28,.T.);
#628=EDGE_CURVE(" ,#476,#478,#32,.T.);
#630=EDGE_CURVE(" ,#478,#480,#36,.T.);
#632=EDGE_CURVE(" ,#480,#463,#40,.T.);
#636=EDGE_CURVE(" ,#587,#588,#45,.T.);
#638=EDGE_CURVE(" ,#588,#587,#50,.T.);
#642=EDGE_CURVE(" ,#595,#596,#55,.T.);
#644=EDGE_CURVE(" ,#596,#595,#60,.T.);
#648=EDGE_CURVE(" ,#603,#604,#65,.T.);
#650=EDGE_CURVE(" ,#604,#603,#70,.T.);
#661=EDGE_CURVE(" ,#463,#483,#74,.T.);
#663=EDGE_CURVE(" ,#483,#484,#294,.T.);
#665=EDGE_CURVE(" ,#464,#484,#234,.T.);
#675=EDGE_CURVE(" ,#575,#582,#78,.T.);
#677=EDGE_CURVE(" ,#575,#562,#82,.T.);
#679=EDGE_CURVE(" ,#562,#564,#86,.T.);
#681=EDGE_CURVE(" ,#564,#566,#90,.T.);
#683=EDGE_CURVE(" ,#566,#483,#298,.T.);
#687=EDGE_CURVE(" ,#480,#580,#94,.T.);
#689=EDGE_CURVE(" ,#580,#584,#98,.T.);
#691=EDGE_CURVE(" ,#584,#582,#102,.T.);
#701=EDGE_CURVE(" ,#575,#576,#106,.T.);
#704=EDGE_CURVE(" ,#582,#571,#142,.T.);
#706=EDGE_CURVE(" ,#576,#571,#130,.T.);
#717=EDGE_CURVE(" ,#576,#578,#110,.T.);
#719=EDGE_CURVE(" ,#580,#578,#114,.T.);
#723=EDGE_CURVE(" ,#478,#498,#118,.T.);
#725=EDGE_CURVE(" ,#498,#557,#262,.T.);
#727=EDGE_CURVE(" ,#557,#560,#122,.T.);
#729=EDGE_CURVE(" ,#562,#560,#126,.T.);
#742=EDGE_CURVE(" ,#571,#572,#138,.T.);
#744=EDGE_CURVE(" ,#578,#572,#146,.T.);
#755=EDGE_CURVE(" ,#572,#584,#134,.T.);
#780=EDGE_CURVE(" ,#476,#496,#150,.T.);
#782=EDGE_CURVE(" ,#496,#498,#266,.T.);
#794=EDGE_CURVE(" ,#474,#494,#154,.T.);
#796=EDGE_CURVE(" ,#494,#496,#270,.T.);
#808=EDGE_CURVE(" ,#472,#492,#158,.T.);
#810=EDGE_CURVE(" ,#492,#494,#274,.T.);
#822=EDGE_CURVE(" ,#470,#490,#162,.T.);
#824=EDGE_CURVE(" ,#490,#492,#278,.T.);
#829=EDGE_CURVE(" ,#525,#526,#166,.T.);
#831=EDGE_CURVE(" ,#526,#528,#170,.T.);
#833=EDGE_CURVE(" ,#528,#530,#174,.T.);

```

#835=EDGE_CURVE(" ,#530,#525,#178,.T.);
 #839=EDGE_CURVE(" ,#541,#542,#182,.T.);
 #841=EDGE_CURVE(" ,#542,#544,#186,.T.);
 #843=EDGE_CURVE(" ,#544,#546,#190,.T.);
 #845=EDGE_CURVE(" ,#546,#541,#194,.T.);
 #856=EDGE_CURVE(" ,#468,#488,#198,.T.);
 #858=EDGE_CURVE(" ,#488,#490,#282,.T.);
 #869=EDGE_CURVE(" ,#505,#512,#202,.T.);
 #871=EDGE_CURVE(" ,#505,#486,#206,.T.);
 #873=EDGE_CURVE(" ,#486,#488,#286,.T.);
 #877=EDGE_CURVE(" ,#466,#510,#210,.T.);
 #879=EDGE_CURVE(" ,#510,#514,#214,.T.);
 #881=EDGE_CURVE(" ,#514,#512,#218,.T.);
 #891=EDGE_CURVE(" ,#505,#506,#222,.T.);
 #894=EDGE_CURVE(" ,#512,#501,#250,.T.);
 #896=EDGE_CURVE(" ,#506,#501,#238,.T.);
 #907=EDGE_CURVE(" ,#506,#508,#226,.T.);
 #909=EDGE_CURVE(" ,#510,#508,#230,.T.);
 #914=EDGE_CURVE(" ,#484,#486,#290,.T.);
 #927=EDGE_CURVE(" ,#501,#502,#246,.T.);
 #929=EDGE_CURVE(" ,#508,#502,#254,.T.);
 #940=EDGE_CURVE(" ,#502,#514,#242,.T.);
 #964=EDGE_CURVE(" ,#557,#558,#258,.T.);
 #976=EDGE_CURVE(" ,#566,#558,#302,.T.);
 #980=EDGE_CURVE(" ,#591,#592,#307,.T.);
 #982=EDGE_CURVE(" ,#592,#591,#312,.T.);
 #986=EDGE_CURVE(" ,#599,#600,#317,.T.);
 #988=EDGE_CURVE(" ,#600,#599,#322,.T.);
 #992=EDGE_CURVE(" ,#607,#608,#327,.T.);
 #994=EDGE_CURVE(" ,#608,#607,#332,.T.);
 #1005=EDGE_CURVE(" ,#558,#568,#340,.T.);
 #1007=EDGE_CURVE(" ,#560,#568,#344,.T.);
 #1020=EDGE_CURVE(" ,#568,#564,#336,.T.);
 #1044=EDGE_CURVE(" ,#591,#587,#348,.T.);
 #1047=EDGE_CURVE(" ,#592,#588,#352,.T.);
 #1070=EDGE_CURVE(" ,#599,#595,#356,.T.);
 #1073=EDGE_CURVE(" ,#600,#596,#360,.T.);
 #1096=EDGE_CURVE(" ,#607,#603,#364,.T.);
 #1099=EDGE_CURVE(" ,#608,#604,#368,.T.);
 #1122=EDGE_CURVE(" ,#525,#517,#372,.T.);
 #1124=EDGE_CURVE(" ,#517,#518,#388,.T.);
 #1126=EDGE_CURVE(" ,#526,#518,#384,.T.);
 #1137=EDGE_CURVE(" ,#530,#522,#376,.T.);
 #1139=EDGE_CURVE(" ,#522,#517,#392,.T.);
 #1151=EDGE_CURVE(" ,#528,#520,#380,.T.);
 #1153=EDGE_CURVE(" ,#520,#522,#396,.T.);
 #1166=EDGE_CURVE(" ,#518,#520,#400,.T.);
 #1183=EDGE_CURVE(" ,#553,#554,#405,.T.);
 #1185=EDGE_CURVE(" ,#554,#553,#410,.T.);
 #1196=EDGE_CURVE(" ,#554,#550,#418,.T.);
 #1198=EDGE_CURVE(" ,#549,#550,#423,.T.);
 #1200=EDGE_CURVE(" ,#553,#549,#414,.T.);
 #1212=EDGE_CURVE(" ,#550,#549,#428,.T.);
 #1234=EDGE_CURVE(" ,#541,#533,#432,.T.);
 #1236=EDGE_CURVE(" ,#533,#534,#448,.T.);
 #1238=EDGE_CURVE(" ,#542,#534,#444,.T.);
 #1249=EDGE_CURVE(" ,#546,#538,#436,.T.);
 #1251=EDGE_CURVE(" ,#538,#533,#452,.T.);
 #1263=EDGE_CURVE(" ,#544,#536,#440,.T.);
 #1265=EDGE_CURVE(" ,#536,#538,#456,.T.);

```
#1278=EDGE_CURVE(",#534,#536,#460,.T.);  
#1309=ADVANCED_BREP_SHAPE_REPRESENTATION(",(#1297),#1308);  
#1316=PRODUCT_DEFINITION('design',"#,#1315,#1312);  
#1317=PRODUCT_DEFINITION_SHAPE(",'SHAPE FOR TURKAY-TEZ2.',#1316);  
#1318=SHAPE_DEFINITION_REPRESENTATION(#1317,#1309);  
ENDSEC;  
END-ISO-10303-21;
```



VITA

Türkay Dereli was born in Denizli, Turkey, in 1968. He received his BSc. and MSc. in Mechanical Engineering from the Middle East Technical University (METU), Gaziantep Extension Campus and University of Gaziantep, in 1992 and 1994, respectively.

From October 1992 to February 1996, he worked as a research assistant at the Mechanical Engineering Department, University of Gaziantep. Since March 1996, he has been an instructor at the same department.

He is the co-author of the book: *“An Introduction to Computer Aided Drafting/Design and AutoCAD”* published by University of Gaziantep, Printing Office, in 1997. His current research interests include CAD/CAM, Process Planning, Feature Recognition, CNC Part Programming and Artificial Intelligence.

He is a member of the Turkish Chamber of Mechanical Engineers (MMO), Turkish Mechanical Design and Production Society (MATİM) and Turkish Information Society (TBD). He is married and has a daughter.