



Founded 1905

**PROCESS PLANNING OPTIMIZATION FOR FIVE-AXIS
SCULPTURED SURFACES FINISHING**

LI HAIYAN

(B.Eng., M.Eng.)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2012

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude and appreciation to my supervisor, A/Prof. ZHANG Yunfeng, from the Department of Mechanical Engineering in National University of Singapore, for his invaluable guidance, advice and discussion throughout the entire duration of this project. It has been a rewarding research experience under his supervision.

I would also like to show my appreciation for the financial support in the form of research scholarship from the National University of Singapore.

Special thanks are given to Dr. LI Lingling for her guidance and suggestions, and GENG Lin for his assistance of this research. I also wish to thank all my other fellow students for their support and a pleasant research environment. Besides, I wish to give my thanks to all my other friends, SHI Min, WANG Xue, etc. for their continuously encouragement.

Finally, I thank my family for their kindness and love. Without their deep love and constant support, I would not have completed the study.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	I
TABLE OF CONTENTS	II
SUMMARY	VII
LIST OF TABLES.....	IX
LIST OF FIGURES.....	X
LIST OF GLOSSARY.....	XIII
Symbols.....	XIII
Abbreviations	XIV
CHAPTER 1 INTRODUCTION.....	1
1.1 Five-axis Sculptured Surface Machining.....	1
1.2 Single Cutter Machining vs. Multi-Cutter Set Machining.....	4
1.3 Process Planning for 5-axis Sculptured Surface Machining.....	5
1.3.1 Cutter selection	6
1.3.2 Tool path generation	7
1.3.3 Integrated process planning	9
1.4 Research Motivation	10
1.5 Objectives and Scope of the Study	12

1.6 Outline of the Thesis.....	13
--------------------------------	----

CHAPTER 2 A-MAP CONSTRUCTION AND ITS

IMPROVEMENT14

2.1 Background	14
2.2 Profile Tolerance in A-map Calculation	17
2.3 Analysis on Part Surfaces in A-map Calculation.....	19
2.3.1 Analysis on part surface for LG checking	19
2.3.2 Analysis on part surface for RG checking	20
2.3.3 Analysis on part surface for GC checking	22
2.3.4 Summary on analysis of part surfaces in interference checking.....	22
2.4 The Improved A-map Construction Algorithm	23
2.4.1 Accessible range for LG avoidance	25
2.4.2 Accessible range for RG avoidance	27
2.4.3 Accessible range for GC avoidance	29
2.4.4 The overall search algorithm	31
2.5 Comparison Study.....	33
2.6 Summary	36

CHAPTER 3 A-MAP APPLICATION FOR 5-AXIS MULTI-

CUTTER SELECTION38

3.1 Background	38
3.2 Identification of Feasible Cutters.....	42
3.3 Cutting Region Allocation for a Feasible Cutter	43
3.3.1 Boundary tracing.....	44

3.3.2 Cutting region identification.....	50
3.4 Effective Cutting Region Identification in Multi-Cutter Set	52
3.5 Construction of Candidate Multi-Cutter Sets	54
3.6 Obtain the Optimal Cutter Set.....	56
3.6.1 Machining strip width estimation	57
3.6.2 Output the optimal cutter set.....	60
3.7 The Overall Algorithm.....	60
3.8 Examples and Discussions.....	61
3.8.1 Case study 1: a benchmark part	62
3.8.2 Case study 2: a general example.....	66
3.9 Summary	67

CHAPTER 4 A-MAP APPLICATION FOR OPTIMAL 5-AXIS CUTTER LOCATION (CL) PATH GENERATION.....69

4.1 Background	70
4.2 Overview of the Proposed Optimal CL-Path Generation Method.....	76
4.3 Optimal Cutter Posture Selection along a Cutting Direction.....	78
4.3.1 Optimal cutter posture selection from the A-map	78
4.3.2 Optimal cutter posture selection through an interpolation approach.....	80
4.4 Optimal Cutting Direction Selection	81
4.5 CL Data Generation with Smooth Posture Change on a Path	85
4.5.1 Calculation of the maximum allowable step-forward length.....	86
4.5.2 Generate the CL data at the next CC point	88
4.6 Step-Over Calculation.....	91
4.7 The Overall Algorithm for CL-Path Generation.....	94

4.8 A Comparison Case Study	96
4.9 Summary	99

CHAPTER 5 MULTI-CUTTER MACHINING: CL PATH

GENERATION, SYSTEM IMPLEMENTATION, AND TESTING

.....	101
5.1 Background	102
5.2 Iso-Planar CL Paths Generation in Multi-Cutter Machining	102
5.2.1 Optimal cutting direction selection	104
5.2.2 CL data generation	106
5.2.3 Case study on multi-cutter CL path generation	110
5.3 An Integrated Process Planning System for Multi-Cutter 5-axis Machining	111
5.3.1 The main interface	112
5.3.2 The input to the system	113
5.3.3 Display of a cutter with a specified posture at a surface point	115
5.3.4 Optimal multi-cutter set selection	117
5.3.5 Multi-cutter CL path generation	118

CHAPTER 6 CONCLUSION AND FUTURE WORK123

6.1 Conclusions	123
6.2 Future Work	126

REFERENCES128

APPENDIX A SURFACE WITH STOCK DATAA-1

**APPENDIX B PART OF CL DATA FOR MULTI-CUTTER
MACHINING IN VERICUT.....B-1**

SUMMARY

This thesis presents the study on the process planning optimization problem for 5-axis finish-cut milling of sculptured surfaces with multi-cutters. The process planning issues addressed include multi-cutter selection and tool-path (cutter location or CL path) generation. In both decision-making processes, maximizing machining efficiency is a common optimization objective. This is also an extension of our previous study on optimal single cutter selection and tool-path generation for 5-axis finish milling of sculptured surfaces. To this end, research work has been carried out in the following aspects.

Firstly, the accessibility range (cutter posture range that is free of interferences) of a cutter to a point on a given surface provides the complete set of information for cutter selection and CL generation. In our previous study, an algorithm was developed to obtain the accessibility map (A-map) of a cutter to a point based on the nominal surface (design surface). In this study, the effects of surface tolerance and stock surface are considered and incorporated into the A-map evaluation algorithm, making the A-map information more accurate.

Secondly, for a partially-accessible cutter to a surface, the cutter is only accessible to some portions of the surface, which is called the *cutting regions* of the cutter. In multi-cutter selection, the identification of cutting regions for every partially-accessible cutter is essential for cutting area assignment to different cutters. In this study, a “boundary tracing” algorithm has been developed for identifying the boundaries of all the cutting regions of a cutter. Measures are also taken to further

refining the boundaries such that (1) the points on the boundaries are interference-free, (2) the boundaries become smoother, and (3) the cutting region is sufficiently large. With this cutter/cutting regions information, for a given multi-cutter set, an algorithm has been developed to assign the whole surface to each cutter so that a cutter in the set has its own *effective cutting regions*. With these two algorithms, all the candidate multi-cutter sets can be established.

Thirdly, for a cutter with one of its cutting regions, an approximation algorithm has been developed to estimate the tool-path length based on the analysis on machining strip width. Therefore, for each candidate multi-cutter set, the overall tool-path length for machining the whole surface can be estimated. The cutting efficiency of different multi-cutter sets can then be compared and the optimal multi-cutter set can be identified.

Fourthly, an optimization algorithm has been developed to identify the optimal cutting direction (iso-planar cutting) for a cutter/cutting-region combination, aiming at maximum cutting efficiency. With each cutter/cutting-region combination, the CL-path is generated by undertaking the following: (1) for a single CL path, the CC points are generated one at a time, followed by posture assignment (towards maximum cutting efficiency) and posture change rate check; and (2) the position of the adjacent or next CL path is found by maximizing the machining strip width such that the scallop-height is just below the given tolerance. The generated CL-paths have the following characteristics: (1) high machining efficiency and (2) satisfying the path smoothness constraint from a CL to the next.

Finally, the overall process planning system has been implemented. Tests have been conducted on many types of sculptured surfaces and its efficacy and effectiveness have been proved.

LIST OF TABLES

Table 2.1 A-map comparison among CA-I, CA-II, and CA-III	34
Table 3.1 Basic information for Run	47
Table 3.2 Connectivity for Run	47
Table 3.3 The <i>inBDs</i> for <i>BDs</i>	50
Table 3.4 Library of fillet-end cutters.....	62
Table 3.5 Case study: cutters' accessible information (<i>ARs/A</i>)	63
Table 3.6 Case study 1: points coordinates (x, z) on each boundary	64
Table 4.1 The comparison between the MMSW-PCR and the PCR-MMSW CL-paths	99
Table 5.1 Tool-path comparison	111
Table 5.2 CL path comparison.....	121

LIST OF FIGURES

Figure 1.1 Comparison of 3-axis and 5-axis milling (Accessibility).....	3
Figure 1.2 Comparison of 3-axis and 5-axis end-milling (effective cutting shape)	4
Figure 2.1 Types of interference	15
Figure 2.2 Types of cylindrical end-mill and its parameters	17
Figure 2.3 The tolerance zone in sculptured surface machining	18
Figure 2.4 Cutter and the offset surfaces at a CC point.....	20
Figure 2.5 Critical position of RG	20
Figure 2.6 Possible positions between cutter bottom and part surface of RG	21
Figure 2.7 Possible positions between cutter body and part surface of GC	22
Figure 2.8 A cylindrical fillet-end cutter at \mathbf{P}_{cc} in the <i>local</i> frame and <i>tool</i> frame.....	24
Figure 2.9 The cutter and surface curve on a normal plane containing \mathbf{x}_ω at \mathbf{P}_{cc}	26
Figure 2.10 Identifying cutter posture range for rear-gouging avoidance	29
Figure 2.11 Identifying cutter posture range for global-collision avoidance.....	30
Figure 2.12 A-map results comparison (CA-I and CA-II) at \mathbf{P}_{cc} ($\theta = 0^\circ$)	35
Figure 2.13 A-map results comparison (CA-II and CA-III) at \mathbf{P}_{cc} ($\theta = 0^\circ$)	36
Figure 3.1 Binary image (red: <i>object</i> points; yellow: <i>background</i> points).....	45
Figure 3.2 The extracted BDs	49
Figure 3.3 Boundaries' relationship.....	50
Figure 3.4 Extracted ARs	52
Figure 3.5 Step-over distance and machining strip width.....	57
Figure 3.6 Machining strip width analysis.....	58
Figure 3.7 Case study 1: surface geometry	62

Figure 3.8 Case study 1: ARs for T_1 and T_6	63
Figure 3.9 Extreme position if a cutter on F1	64
Figure 3.10 Case study 2.....	66
Figure 3.11 The $eARs$ of T_1 , T_5 and T_8	67
Figure 4.1 Types of tool-paths based on path topology.....	71
Figure 4.2 Flowchart of the optimal CL-path generation method	77
Figure 4.3 Feeding direction at \mathbf{P}_{cc} in global frame and local frame	83
Figure 4.4 A single iso-planar path.....	86
Figure 4.5 Calculation of step-forward length for an iso-planar path.....	86
Figure 4.6 Evaluation of machining-strip width at point \mathbf{P}_{cc}	92
Figure 4.7 Calculation of path interval between two adjacent paths at a CC point \mathbf{P}_j	93
Figure 4.8 The part surface	97
Figure 4.9 Average machining strip widths along all the cutting directions	97
Figure 4.10 The CL-paths generation from the two methods.....	98
Figure 4.11 CL-paths comparison along all the cutting directions.....	99
Figure 5.1 Flowchart of 5-axis multi-cutter iso-planar CL path generation.....	103
Figure 5.2 Cutting direction selection approaches.....	105
Figure 5.3 CL path generation for multi-cutter machining.....	108
Figure 5.4 Iso-planar tool-paths with <i>multi-cutter</i> and <i>single-cutter</i> modes	110
Figure 5.5 The part model.....	112
Figure 5.6 Main menu bar.....	113
Figure 5.7 System input: part surface and cutter library.....	114
Figure 5.8 Access to system input: tolerances	114
Figure 5.9 System input: tolerances.....	115
Figure 5.10 Specifying the point of interest.....	116

Figure 5.11 Cutter with posture (0,0) at point (0.4, 0.1).....	116
Figure 5.12 Cutter with posture (0, 68°) at point (0.4, 0.1)	117
Figure 5.13 Dialog box: specifying MR_{min}	118
Figure 5.14 Output of the optimal multi-cutter set	118
Figure 5.15 Generate CL paths for optimal multi-cutter set.....	119
Figure 5.16 The identified optimal cutting direction	119
Figure 5.17 Specify the cutting direction.....	120
Figure 5.18 The generated iso-planar CL paths for multi-cutter set machining.....	120
Figure 5.19 Iso-planar CL paths using a single accessible cutter (T_9).....	121
Figure 5.20 Machining simulation result for multi-cutter and single cutter modes...	122

LIST OF GLOSSARY

Symbols

f :	feeding direction
R :	Cutter major radius
r_f :	Cutter minor radius
L :	Cutter length
τ :	Profile tolerance
τ_{in} :	Inside tolerance
τ_{out} :	Outside tolerance
S_D :	Design surface
S_M :	Machined surface
S_{in} :	Inside bound surface
S_{out} :	Outside bound surface
W :	The machining strip width
(X_L, Y_L, Z_L) :	The local coordinate frame at a surface point
(X_T, Y_T, Z_T) :	The tool coordinate frame at a surface point
λ :	Tilting angle of a cutter at a surface point in the local frame
θ :	Rotational angle of a cutter at a surface point in the local frame
κ_{max} :	The maximum principal curvature on a surface point
κ_{min} :	The minimum principal curvature on a surface point
κ_{tmax} :	The maximum principal curvature on a cutter surface point
κ_{tmin} :	The minimum principal curvature on a cutter surface point

T_a : The largest cutter that can finish the whole surface without interference

L_{index} : The notion of the tool-path length index

h : The scallop height tolerance

α_ω : The feeding angle represented in global frame

τ_{PCR} : Posture change rate tolerance

Abbreviations

A-map: Accessibility map

APs: Accessible points

APR: Accessible posture range

ARs: Accessible regions

BAPs: Boundary accessible points

BDs: Boundaries

CA: Cutter accessibility

CAM: Computer-Aided Machining

CC point: Cutter contact point to the surface

CL: Cutter location

CNC: Computer Numerical Controlled

DOF: Degrees of freedom

eAPs: Effective accessible points

eARs: Effective accessible regions

GC: Global-collision

LG: Local-gouging

ML: Machine axis limits

MMSW: Maximum machining strip width

MR: Machining ratio

MSW: Machining strip width

NC machine: Numerically controlled machine

NURBS: Non-Uniform Rational B-Spline

PCR: Posture change rate

RG: Rear-gouging

CHAPTER 1

INTRODUCTION

Five-axis end milling has been increasingly used for fabricating parts with sculptured surfaces, such as turbine blades, propellers, 3D moulds and dies. With the added two more degrees of freedom (DOF) than 3-axis end-milling, 5-axis end-milling allows simultaneous change of cutter position and orientation to match the part surface, and thus offers many advantages such as better cutter accessibility, set-up process reduction, fast material removal rates, and improved surface finish.

With the availability of high speed automatic tool change mechanisms on modern Computer Numerical Controlled (CNC) machines, multi-cutter machining of sculptured surface has become quite attractive. Compared to machining using a single cutter, the application of multi-cutter machining provides more potential on improving the cutting efficiency.

This chapter briefly introduces the technology of 5-axis end-milling in sculptured surface machining using a single cutter as well as a multi-cutter set. The necessity for automated process planning is highlighted. Furthermore, based on the discussion of the state-of-art in commercial Computer-aided Manufacturing (CAM) systems and published research work, the motivation of this study is presented, followed by the detailed description of the research scope.

1.1 Five-axis Sculptured Surface Machining

With the increased aesthetic appeal and complex functional needs in aerospace, shipbuilding, automotive, and dies/moulds manufacturing industries in recent years, the demand for complicated mechanical components with sculptured surfaces has risen rapidly (Balasubramaniam *et al.*, 2003; Radzevich, 2005). Due to the irregular distributed curvature, machining these surfaces is a challenging task. Traditionally, these surfaces are produced by the skilled hands of artisans. However, due to the manual involvement, this method is time-consuming and error-prone. With the growing industrial demand for sculptured surfaces and the development of CNC machines, CNC milling with super accuracy and efficiency becomes a vital approach in sculptured surfaces manufacturing (Choi and Jerard, 1998).

In general, CNC sculptured surface milling consists of two main stages of metal removal operations: roughing and finishing. Roughing is initially applied to remove bulk of materials from the stock to obtain the intermediate part surface, while finishing is to further machine the intermediate part surface into final part surface, which has surface error within the specified tolerance. A fair amount of time is spent in finishing phase due to the small pick-feed rate and the accuracy requirement. Therefore, the efficiency and accuracy of the whole machining process largely depends on that of the finishing stage.

The most common types of CNC sculptured surface milling are 3-axis and 5-axis end-milling. Three-axis end-milling has played an important role at the beginning of CNC machining age. In 3-axis end-milling, the cutter moves with a fixed axis direction to any point in its workspace with 3 translational DOF. Because of the simple translational tool movement, it is easy to position the tool during the milling. With the growing need for complex components in industry, 5-axis end-milling has gained more popularity in sculptured surface machining. In 5-axis end machining,

with 3 translational joints and 2 rotational joints, the machine not only moves a tool to any point in its workspace, but also positions it in any arbitrary orientation relative to the surface. Therefore, compared to 3-axis machining, 5-axis machining of sculptured surfaces offers many advantages.

Firstly, in 5-axis end-milling, a cutter has better accessibility. As shown in Figure 1.1a, during one setup in 3-axis end-milling, only those regions of a part that are visible from a particular direction can be milled and inaccessible regions need to be milled by reconfiguring the cutter setup along another direction. In 5-axis end-milling, the cutter can reach the local surface by changing the orientation dynamically to access the areas that are inaccessible to a cutter in 3-axis end-milling (see Figure 1.1b). This flexibility of 5-axis end-milling results in fewer setups and therefore higher productivity.

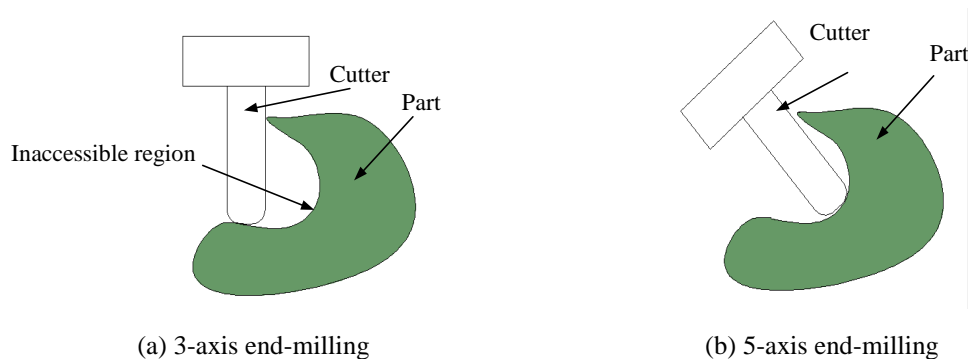


Figure 1.1 Comparison of 3-axis and 5-axis milling (Accessibility)

Secondly, 5-axis end-milling can improve machining productivity and machined surface quality. As shown in Figure 1.2a, during 3-axis finishing, the cutter's cutting geometry is unchangeable in respect to the changed surface features. This results in large scallops left after machining, which may require substantial hand polishing. However, as can be seen in Figure 1.2b, in 5-axis end-milling, the cutter's cutting profile is able to closely match the part surface profile by

dynamically adjusting the orientation to achieve wider effective cutting edge. This leads to much less scallops left on the part surface and larger machining strip width.

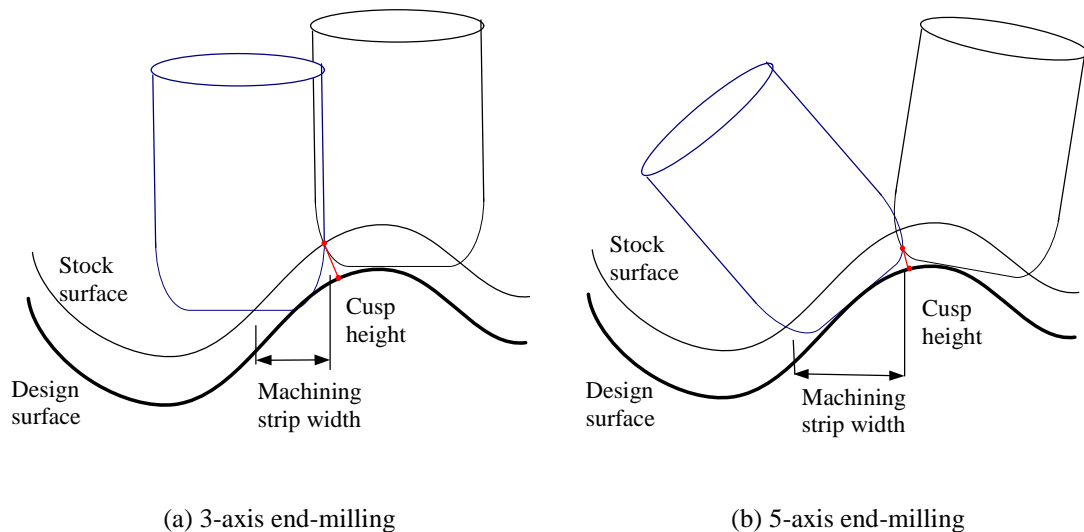


Figure 1.2 Comparison of 3-axis and 5-axis end-milling (effective cutting shape)

However, despite the advantages in 5-axis machining, the increased flexibility by the two additional revolute axes also leads to complication in process planning, e.g., cutter selection and tool path generation.

1.2 Single Cutter Machining vs. Multi-Cutter Set Machining

In sculptured surface machining, a larger cutter generally yields high efficiency but more likely to cause interference (gouging and collision), while a smaller cutter is less likely to cause interference but generally needs longer tool path and machining time. As the last phase of 5-axis machining, finish cut is where products with sculptured surfaces take their final shape. Thus, it is desirable to use multiple cutters for this task to meet both efficiency and accuracy requirements. The larger cutters can be used to machine the large flat or convex areas while smaller cutters are used for the critical concave or saddle surface regions. However, the idea of multiple cutters was once considered impractical and uneconomical as manual tool

change and set up in machining operations would take too much time. A single cutter to machine the whole surface was usually preferred. Nowadays, with the availability of high speed automatic tool change mechanisms on modern CNC machines, tool changing can now be achieved within seconds and the once costly tool change time penalty in multi-cutter machining is greatly reduced. The use of multiple cutters thus has become quite attractive. In multi-cutter machining, typically larger cutters are used wherever possible to quickly remove large amounts of material while smaller tools are then used where the big tool cannot access without gouging and collision. The use of multiple cutters is possible to achieve significant reductions in processing time and machining cost compared to the use of a single cutter to machine the whole surface (Lim *et al.*, 2000, Gau, 1997), especially for surfaces with a large nearly-flat or convex areas but with small critical concave or saddle areas. In addition, multi-cutter machining will result in less tool wear due to the decreased machining time.

Although multi-cutter machining provides more advantages, it also increases the complexity in process planning, like how to choose the optimal multi-cutter set and how to identify the cutting areas for each cutter in the multi-cutter set in tool path generation phase, etc.

1.3 Process Planning for 5-axis Sculptured Surface Machining

In 5-axis sculptured surface machining, either using a single cutter or a multi-cutter set, process planning is an important issue. It includes cutter selection and tool path generation. The former selects a cutter or a multi-cutter set from the given cutter library that must be able to traverse the whole surface without causing interference. The latter selects a tool path pattern, generates the cutter contact (CC) points, and determines the cutter's posture (orientation) at each CC point. In both planning tasks,

the primary concern is to avoid interference between the cutter, the part, and the environment. Due to the complicated tool movement and complex surface shape, it is a challenging task to determine the interference-free posture during process planning.

Currently, most of the commercially available CAM systems do not have a systematic method on automatic process planning for 5-axis sculptured surface machining (Balasubramaniam *et al.*, 2003). They generally require intensive user interference on checking, verification, and reworking of the NC part programming (Jun *et al.*, 2003). On the other hand, there has also been a fair amount of reported work in the area of the automation of process planning for 5-axis machining of sculptured surfaces since the late of 1980's (Lee and Chang, 1996; Lee, 1998; Jensen *et al.*, 2002; Chiou and Lee, 2002; Li and Zhang, 2006). A brief review of some of relevant work to this study is given in the following sections.

1.3.1 Cutter selection

Cutter selection lies at the heart of manufacturing processes, which affects not only the productivity but also the surface finish (Lim *et al.*, 2000). In most of the commercially available CAM systems, it still requires skillful human intervention to input the cutter parameters (Chiou and Lee, 2002). In addition, most of the reported 5-axis tool-path generation methods focus on developing automated methods of generating interference-free tool path by assuming that the cutter is already selected. However, it is nearly impossible for a user to determine what may be an optimal cutter or multi-cutter set for a given sculptured surface. To avoid potential problems associated with gouging and collision, the user often has to make a very conservative choice that result in low machining efficiency and high production cost.

Among the reported work on automated cutter selection, most of them are of 3-axis end-milling. There has also limited reported work on automatic cutter selection for 5-axis sculptured surface machining (Lee *et al.*, 1996; Jensen *et al.*, 2002), which mainly focused on developing algorithms to select the single largest cutter that can traverse the whole surface without interference. The major limitation of the algorithms developed by Lee *et al.* (1996) and Jensen *et al.* (2002) is that the algorithms are trial-and-error in nature, which chooses a cutter and then conducts the procedure of tool-path generation for verification. This leads to either heavy computational load or compromise of machining efficiency. There is no reported effective method that is able to choose an optimal cutter for 5-axis finish cut on a given surface without generating the tool-path.

On the other hand, there are also some reported studies on algorithms of multi-cutter selection for end-milling of sculptured surfaces (Yang and Han, 1999; Arya *et al.*, 2001; D'Souza *et al.*, 2004). Though comprehensive and effective, these algorithms are limited to selecting multi-cutter set in 3-axis machining, which cannot be directly extended to 5-axis sculptured surface machining owing to the two additional rotational DOF in a 5-axis machine. Nevertheless, the considerations on multi-cutter selection in these studies provide useful references to develop an efficiency algorithm for 5-axis optimal multi-cutter set selection in this study.

1.3.2 Tool path generation

For 5-axis sculptured surface machining, several tool-path topologies/patterns have been studied, such as serial-pattern (Lee, 1998; Chiou and Lee, 2002; Li and Feng, 2004), radial-pattern (Kim and Choi, 2002), and contour-pattern (Park, 2003). Both the serial-type and radial-type are for machining one area, while the contour-

type is for cutting a vertical or slant wall (Choi and Jerard, 1998). Besides, one other pattern is called iso-planar (Jensen and Anderson, 1993; Pi *et al.*, 1998), which defines tool-paths on a series of parallel intermediate planes in the Cartesian space. Iso-planar tool path topology is effective on sculptured surface as curves from different surface patches are joined into a single tool path and thus it is widely employed in practice.

For iso-planar tool path topology, the direction of the parallel intermediate planes, which is also known as the cutting direction, needs to be specified before tool path generation. This is effectively an optimization issue, e.g., which cutting direction gives the shortest tool-path or highest cutting efficiency. Over the years, there have been several studies on developing the algorithms in selecting the optimal cutting directions (Held, 1991; Park and Choi, 2000). However, these works are limited to two dimensional area machining with fixed cutter axis, which cannot be directly used in 5-axis sculptured surface machining.

When the cutting direction is specified, the CL paths can be generated in an iterative manner: (1) generation of the CL data on the first tool-path and (2) generation of the CL data on the next tool-path, one at a time. Here, the CL data refers to the location of a CC point and the corresponding cutter posture. To achieve high cutting efficiency, it is desirable to have the machining strip width as large as possible while satisfying the accuracy requirement (Choi and Jerard, 1998; Lee, 1998). Generally, the maximum machining strip width is achieved at the current CC point by following an iterative approach: (1) searching for a suitable cutter posture at the CC point aiming at maximizing the machining strip width and (2) calculating the deviation between the machined surface and the design surface. If the resultant surface error reaches its maximum possible value within the pre-defined profile

tolerance, the CL data is found. Otherwise, the cutter posture will be adjusted followed by surface error evaluation. It can be seen that in this procedure the step for searching cutter posture is essential. However, due to the complexity of optimal cutter posture selection in 5-axis machining, most of the reported work suffers heavy computation load. In addition, cutter dynamics plays an important role in surface finish. So far, the reported work on 5-axis tool-path generation has not paid much effort on this.

1.3.3 Integrated process planning

In process planning for 5-axis sculptured surfaces end-milling, the cutter's accessibility to the part surface is an important issue to be addressed in both cutter selection and CL path generation. The issue to be addressed in cutter selection is to make sure that the cutter has an interference-free posture at every point on its cutting regions, while in CL path generation, it is essential that the final selected CL data do not cause any interferences. In other words, accessibility evaluation for a cutter at a point on the surface is a common issue in these two planning tasks. It is therefore desirable to integrate these two tasks by obtaining the accessibility information and share it between the two tasks. Such integration could help increase the efficiency significantly. However, most of the reported work treats the cutter selection and tool-path generation as two separate tasks.

In our previous work (Li, 2007), the cutter's accessible orientation range at a surface point is named as accessibility map (A-map). A unique algorithm has been developed to evaluate the A-map of a fillet-ended cutter by considering the machine axis limits, avoidance of local-gouging, rear-gouging, and global-collision. Based on the A-map evaluation algorithm, with the focus on single cutter machining, Li (2007)

has developed an integrated algorithm that is able to find the optimal fillet-end cutter and then generate the iso-planar CL path for 5-axis sculptured surface finishing. Firstly, the cutter selection is conducted before tool path generation. This is achieved by, for each cutter, conducting A-map evaluation at every sampled surface point of the surface. Starting from the largest available cutter in the cutter library, if the A-map is empty at a sampled point, a smaller cutter is chosen to repeat the A-map evaluation procedure until a cutter that has non-empty A-map at every sampled point is found. Secondly, since the density of the sampled points is generally much higher than that of the CC points generated at the later stage, the A-map information of the selected cutter can be used for CL path generation/optimization. An algorithm for iso-planar CL path optimization has been developed aiming at smooth cutting dynamics as the top-prioritized objective and maximum machining strip width as the secondary-prioritized objective. In the algorithm, the optimal path direction with minimum average orientation change rate over the whole surface is selected to maintain smooth tool dynamics. In the CL path generation stage, the A-maps of the sampled surface points are used as references for obtaining the accessibility of the generated CC points by applying an interpolation procedure. In this way, the process planning is achieved in an integrated way.

1.4 Research Motivation

Process planning is an important task in 5-axis sculptured surface machining. At the same time, it is also a challenging and difficult task since it involves the simultaneous consideration of multiple constraints as well as optimization issues such as cutting efficiency and smooth dynamics. Based on the brief review of the previously reported research literatures in the last couple of sections, several

important research issues have been identified that need further study in order to achieve automated process planning. First of all, in the reported work, cutter selection and tool-path generation are treated as separate tasks, leading to information redundancy and extra computation load. Secondly, cutter selection is either ignored or addressed during CL path generation in a trial-and-error manner. Thirdly, among the limited reported work on cutter selection for 5-axis machining of sculptured surfaces, only single cutter is considered; while reported work on multi-cutter selection mainly focused on 3-axis machining, which cannot be directly applied to 5-axis machining. Finally, for CL path generation, an optimization issue on how to select the optimal cutting direction for maximum cutting efficiency in iso-planar pattern still needs extensive study.

On the other hand, the developed approach for process planning in our previous work (Li, 2007) follows the integrated mode. The concept of A-map makes the exploitation of the full advantages of 5-axis machining achievable. By using the A-map, the optimal cutter selection (single cutter) is performed before CL path generation. For iso-planar pattern, an algorithm has been developed to select the optimal cutting direction selection aiming at smooth tool dynamics. The research work presented in this thesis follows this basis integrated approach to address the following unaddressed issues:

- (1) In the A-map construction algorithm by Li (2007), only the nominal surface or design surface is considered for interference avoidance. This may cause errors.
- (2) The automated cutter selection algorithm is only applicable for single cutter machining, which does not make full use of larger cutters for better efficiency.
- (3) In the CL path generation algorithms, the machining strategy for the iso-planar pattern is smooth cutter dynamics as well as high cutting efficiency. This is

achieved by setting smooth cutting dynamics as the objective for cutting direction selection while using maximum cutting strip width as the objective in CL determination. These two different strategies in two different stages may cause conflicts.

1.5 Objectives and Scope of the Study

The integrated process planning approach proposed in our early work (Li, 2007), in which the A-map information is used throughout different stages of process planning, has been proved to be effective and efficient. This study follows this approach in general, and at the same time, addresses some unresolved but critical issues to make the integrated process planning system more complete and comprehensive. The main problem covered in this work includes optimal multi-cutter set selection and iso-planar CL path optimization. The detailed objectives are given as follows:

(1) Improvement on the A-map construction algorithm

In the previous work, the profile tolerance is not taken into consideration during the A-map construction. To make the algorithm more practical, in this research, the tolerance is added into the accessible range calculation for avoidance of all types of interference.

(2) A-map application on optimal multi-cutter set selection

To make full use of the efficiency potential of larger cutter, and meanwhile, grant the machined surface quality by using a smaller cutter in the critical part, this research will investigate to select one optimal multi-cutter set from a fillet-end cutter library for finishing a NURBS surface with maximum cutting efficiency.

(3) Cutting regions construction for each cutter in the multi-cutter set

During the multi-cutter set machining, it is essential to identify each cutter's cutting region. Therefore, the method for allocating whole machining surface to different cutters will be developed.

(4) A-map application on iso-planar CL path optimization

As the extension of our previous work on CL path optimization, this research will develop a different optimization method on iso-planar CL path generation. The maximum cutting efficiency is guaranteed by selecting the cutting direction as the one that achieves the maximum machining strip width over the whole surface, while the cutter dynamics is controlled by making sure the cutter posture change between two CC points along each path is minimized.

1.6 Outline of the Thesis

In this thesis, four technical issues are researched with corresponding solutions/algorithms developed. Since the natures of these 4 issues are quite different, literature on each issue will be conducted independently and presented in the corresponding chapter. In Chapter 2, the improvement on A-map construction is presented. In Chapter 3, the method for automatically selecting an optimal multi-cutter set for 5-axis sculptured surface finishing is presented. In addition, the algorithm for cutting area identification for each cutter in the multi-cutter set is detailed. The method for iso-planar CL path optimization on a surface region with single cutter machining is proposed in Chapter 4. The integration of process planning for 5-axis multi-cutter set machining is given in Chapter 5. In Chapter 6, the conclusions and recommendations for future work are given.

CHAPTER 2

A-MAP CONSTRUCTION AND ITS IMPROVEMENT

In 5-axis finish end-milling of a sculptured surface, given a cutter, the interference-free posture range for the cutter at any point on the surface is probably the most important property for process planning tasks such as cutter selection and CL path generation. In our previous work (Li, 2007), this property is defined as accessibility map of a cutter to a point on the surface, called A-map. By considering the interference avoidance including machine axis limits, local gouging, rear gouging, and global collision, a method to construct the A-map was developed, which is discrete in nature. However, the surface profile tolerance and the stock surface were not taken into consideration, i.e., only nominal geometry of the design surface is assumed. In this study, this problem is rectified. The original A-map construction method is briefly introduced in this chapter, followed by the improvements.

2.1 Background

In 5-axis sculptured surfaces end-milling, there are three types of interferences: local-gouging (LG), rear-gouging (RG) and global-collision (GC), as shown in Figure 2.1, where \mathbf{P} is the cutter contact (CC) point and \mathbf{f} the feeding direction. LG (see Figure 2.1a) refers to the removal of excess materials in the vicinity of the CC point due to the mismatch of the curvatures of the cutter's local surface and those of the part surface at the CC point. RG (see Figure 2.1b) refers to the removal of excess material by the rear of the cutter due to the large cutter size or the inaccessible orientation. GC

(see Figure 2.1c) occurs when a cutter contacts with the part surface through the cutter holder or cutter shank. In both tasks of 5-axis process planning, apart from accuracy concerns, the primary consideration is to avoid these three types of interference.

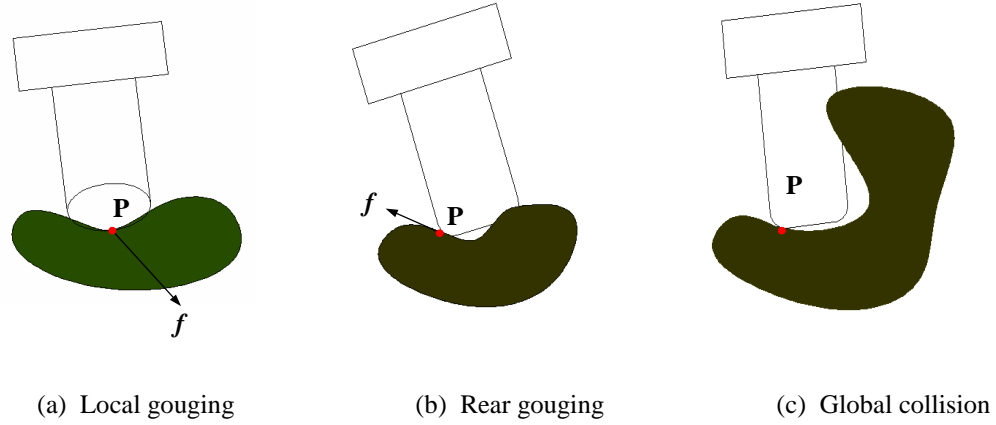


Figure 2.1 Types of interference

So far, there has been much reported work on developing methods/strategies to avoid interferences in 5-axis end-milling of sculptured surfaces. In general, these methods can be categorized into two basic types as follows.

The first type is single interference-free orientation determination. Li and Jerard (1994) proposed a check-and-correct method to find a suitable orientation to the cutter. Chen *et al.* (2005) presented an approach to obtain the interference free posture by avoiding LG and RG. The cutter posture that is LG free and produces the maximum cutting efficiency is obtained firstly by matching the instantaneous cutting profile of the cutter and the surface as much as possible. The inclination angle is then adjusted to avoid RG. This method takes the curvature match into account to obtain the initial LG free posture. However, if there is no suitable inclination angle for RG free, there is no mentioning on how to conduct further search. Similar methods can be found in Lee (1998), Pi *et al.* (1998), Wang and Yu (2003). The advantage of this type

of method is that computation load is generally low. However, it does not make full use of the flexibility potential of 5-axis machining.

The other type is interference-free orientation range construction. The feasible orientations are constructed through different approaches. Lee and Chang (1995) proposed a 2-phase approach by using convex hull property of B-spline surfaces to approximate the local visibility. Balasubramaniam *et al.* (2000) calculated the accessibility through visibility computation and posture definition. The concept of visibility is firstly used to determine the direction from which a point in the delta-volume is accessible to an observer outside the convex envelop using graphics hardware. Since the visibility could not take the cutter's geometry into consideration, a pseudo-gradient search is then performed in the neighborhood of the visibility direction to define the valid posture by representing the cutter in a series of triangulated slabs. The iterations of checking interferences and correction of postures are then applied. This method could achieve efficient computation by making use of the graphics hardware. However, although the visibility method could result a set of accessible range, the pseudo-gradient search is still a trial-and-error method.

In our previous work, regarding to cutter's accessible range, the concept of A-map is in use. When a cutter is positioned at a point on the part surface, its A-map refers to the posture range in terms of the two rotational angles within which the cutter does not have any interference with the part and the surrounding objects. The A-map construction algorithm for cylindrical fillet-end cutters was developed. The general cylindrical fillet-end mill shown in Figure 2.2a also covers two special cases: the flat-end mill (see Figure 2.2b) and ball-end mill (see Figure 2.2c). The key parameters of a cylindrical fillet-end cutter include cutter major radius (R), fillet radius (r_f), and cutter length (L). The fillet-end cutter becomes a flat-end cutter when

$r_f = 0$ and a ball-end cutter when $R = r_f$. In the A-map construction algorithm, the machining surface is firstly discretized into a set of high-density points, called sampled points. At each point, the A-map is constructed by considering the avoidance of machine axis limits (ML), LG, RG and GC.

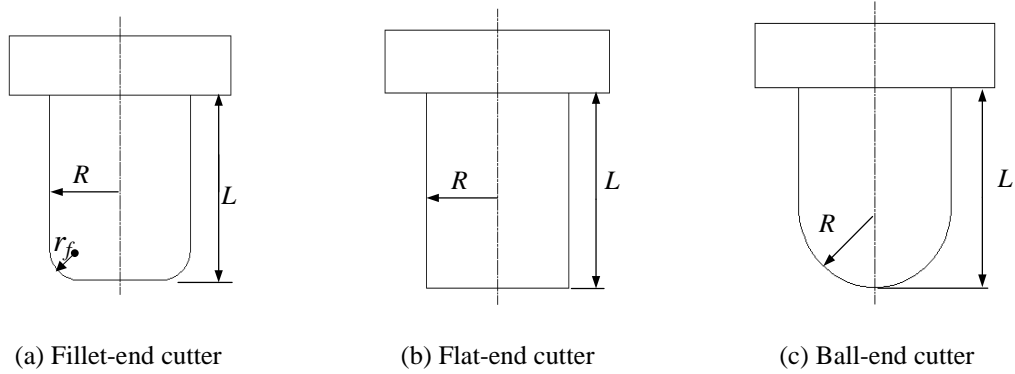


Figure 2.2 Types of cylindrical end-mill and its parameters

The A-map effectively characterizes the accessibility of a cutter to a point on the machining surface, which provides all important geometric information for cutter selection and generation of interference-free tool-paths. However, the A-map construction algorithm is based on the nominal design surface only, whereas in reality, the shape error tolerance and the stock surface are also need to be taken into consideration. This simplification could result in some errors in the actual machining. In addition, the cutter length is assumed to be infinite in our previous work, which contributes to more conservative A-maps in general. Therefore, in this study, the A-map construction algorithm will be modified to address these two problems.

2.2 Profile Tolerance in A-map Calculation

In sculptured surface machining, compared with the design surface, the machined surface must be within a specified machining tolerance, which is called the profile tolerance τ (Choi and Jerard, 1998). The basic concept of a surface profile

tolerance is illustrated in Figure 2.3. A tolerance zone is defined by the envelope surfaces created by sweeping a sphere of diameter τ along the surface. It defines the allowable variations of the surface profile. As long as the machined surface (S_M) lies inside the tolerance zone then the surface is said to be within tolerance. For the sake of convenience, the surface profile tolerance is given as inside tolerance τ_{in} and outside tolerance τ_{out} . For a design surface S_D , the two bounded tolerance surfaces are named as the outside bound surface S_{out} and inside bound surface S_{in} , which are expressed as (Jensen *et al.*, 2002):

$$\begin{aligned} S_{out} &= S_D + n\tau_{out} \\ S_{in} &= S_D - n\tau_{in} \end{aligned} \quad (2.1)$$

where n is the unit surface normal vector.

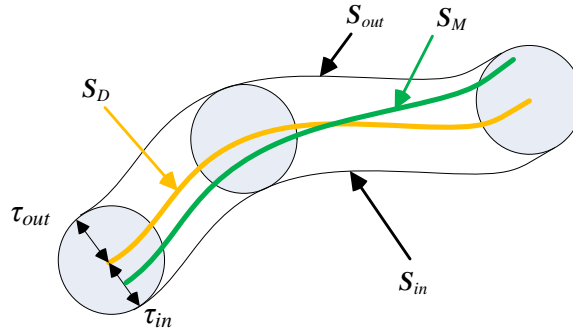


Figure 2.3 The tolerance zone in sculptured surface machining

Another important surface is the stock surface S_S , which is the surface before finish cut. It is set at an offset ε from S_D and expressed as,

$$S_S = S_D + n\varepsilon \quad (2.2)$$

In general, $\varepsilon > \tau_{out}$ and $\varepsilon < r_f$.

Since both the stock surface S_S and the machined surface S_M exist during the finish-cut machining, it is important to choose a proper “part surface” for the calculation of various interface avoidance during A-map construction. In the following sections, the suitable part surfaces used for checking each type of

interferences are discussed. Based on the analysis, an improved A-map construction method is developed and the details of the sub-algorithms for obtaining the various interference-free ranges are described.

2.3 Analysis on Part Surfaces in A-map Calculation

2.3.1 Analysis on part surface for LG checking

Local-gouging (LG) occurs when the curvature of the cutter's local surface is smaller than that of the part surface at the CC point such that the cutter cuts excess material. For LG avoidance, it is important to know which surface should be used to generate the CC points.

Theoretically, to machine the same portion of the surface, a CC point can be located on any of these surfaces: S_D , S_M , S_{in} and S_{out} . Figure 2.4 shows two possible positions of CC point \mathbf{P}_{cc} (on S_D) and \mathbf{P}_{cc}' (on S_{in}), together with the cutter positions at the two CC points and all the part surfaces except S_M . Note that f represents the feeding direction at the CC point. With a fixed posture, the corresponding machining strip widths are shown as W (CC point is \mathbf{P}_{cc}) and W' (CC point is \mathbf{P}_{cc}'). It can be easily seen that $W' > W$. That is, the deeper the part surface (where the CC point is located, e.g., S_{in}) is, the larger the machining strip width could be. Therefore, for better machining efficiency, CC points should be chosen on S_{in} . However, due to the approximation nature of the point represented part surface, a generated CC point may be slightly under S_{in} . Therefore, it would be safer to choose CC point on a surface above S_{in} . As a trade-off between cutting efficiency and safety, CC point is chosen on S_D in this study. As a CC point is chosen on S_D , the local machining region, which is the surface region around the CC point, is also on S_D . Therefore, S_D is used as the part

surface to check for LG avoidance. That is, to avoid LG at \mathbf{P}_{cc} , the curvature of the cutter's local surface should be larger than that of S_D along all possible directions.

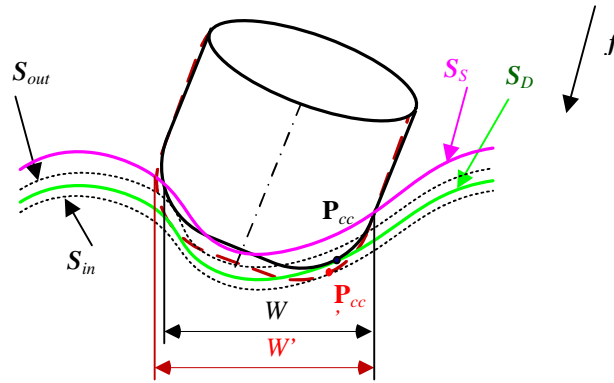


Figure 2.4 Cutter and the offset surfaces at a CC point

2.3.2 Analysis on part surface for RG checking

At a CC point, rear-gouging (RG) occurs if a point on the cutter bottom surface and outside the local machining region (e.g., point \mathbf{Q} on the cutter in Figure 2.5) is underneath the machined surface S_M . Regarding to point \mathbf{Q} , Figure 2.5 shows the critical position for RG avoidance where \mathbf{Q} is on S_M . To avoid RG, \mathbf{Q} must not fall below S_M . Since S_M is unknown at this stage, a proper part surface should be found to replace it.

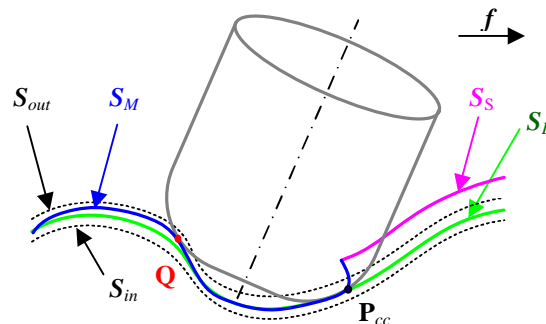


Figure 2.5 Critical position of RG

By ignoring S_M , Figure 2.6 shows 4 possible RG cases between the cutter and the different part surfaces. Points Q_{out} , Q_D and Q_{in} shown in Figure 2.6a, b, and c, respectively, represent the cutter bottom point Q falling on S_{out} , S_D and S_{in} , respectively. If Q falls on any other position above the one in Figure 2.6a, no RG will occur. For finish-cut machining, RG in Figure 2.6b and c are not allowed, although it satisfies the tolerance requirement (Jensen *et al.*, 2002). Figure 2.6d shows that some portions of the cutter bottom falls underneath S_{in} . This position does not satisfy the tolerance requirement and it thus is ruled out. In summary, Figure 2.6a is considered as the critical position for RG check and correction, i.e., S_{out} , instead of S_D used in the previous work, should be used for RG check. Since S_{out} is outside of S_D , the RG check will result in a tighter posture range compared to that resulted from the previous work.

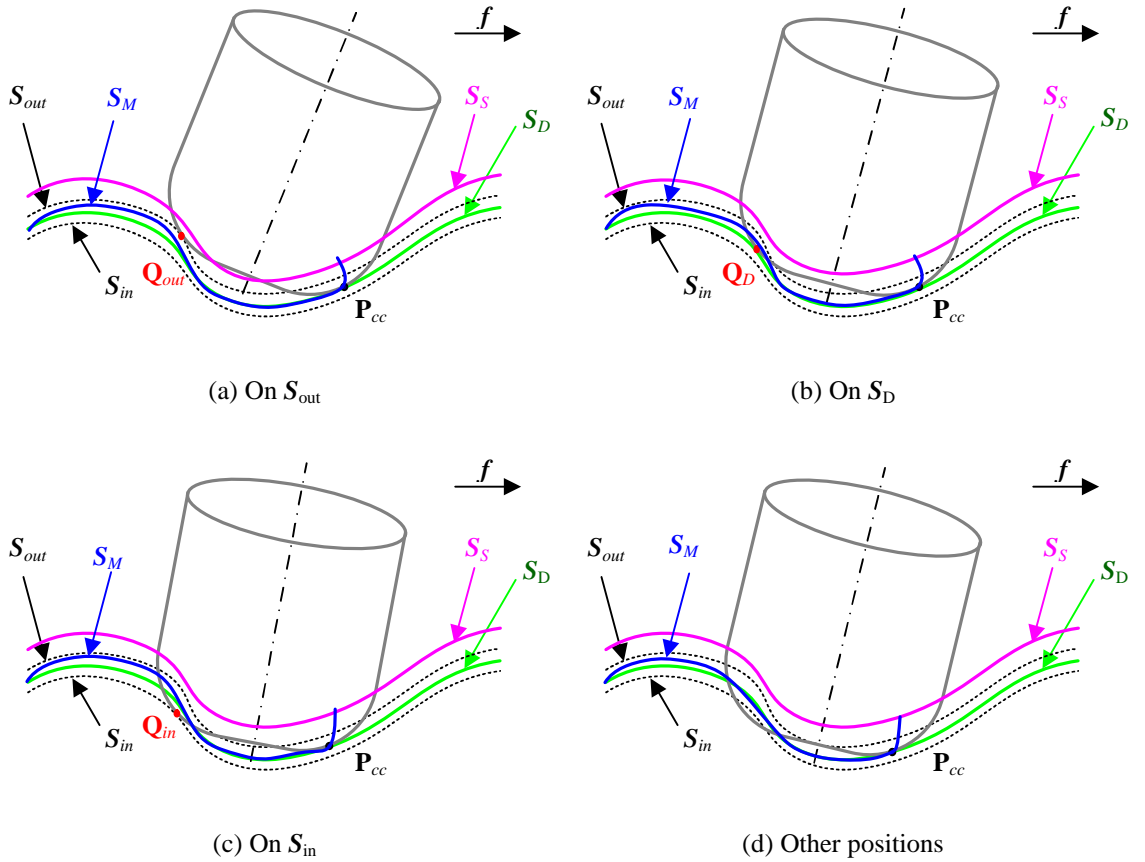


Figure 2.6 Possible positions between cutter bottom and part surface of RG

2.3.3 Analysis on part surface for GC checking

At a CC point, global-collision (GC) is defined as the interference between the cutter body and the machined surface S_M or the stock surface S_S . Based on the definition, to identify GC at a CC point P_{cc} , Figure 2.7a shows the critical position where the cutter body collides with S_S at point Q_1 ; while Figure 2.7b shows the critical position where the cutter body collides with S_M at point Q_2 . As GC results in serious cutter damage, it is strictly not allowed. Since S_M is unknown at this stage, GC is checked on S_S to achieve a conservative A-map. That is, for GC avoidance, S_S is chosen as the part surface to check the interference with the cutter surface.

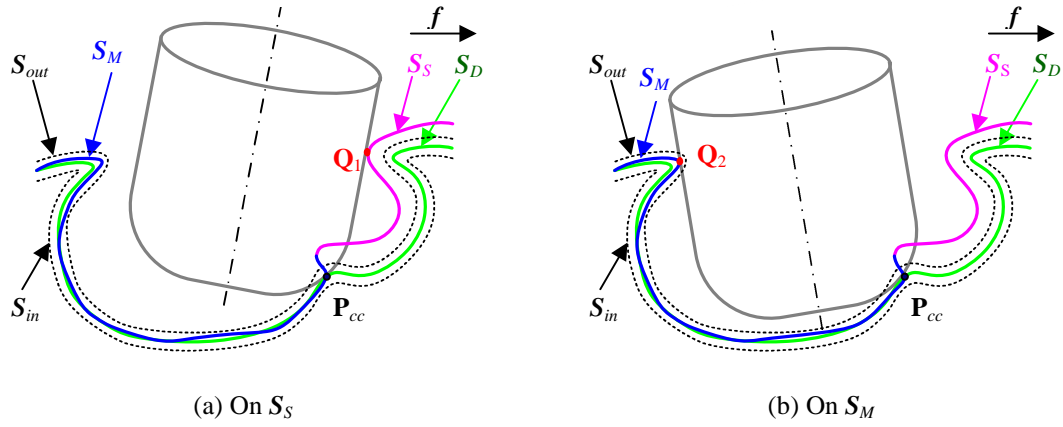


Figure 2.7 Possible positions between cutter body and part surface of GC

2.3.4 Summary on analysis of part surfaces in interference checking

By considering the machining error and profile tolerance, the CC point should be chosen on S_D . In summary, the ideal cutter posture is: (1) no interference in the vicinity of the CC point, (2) cutter rear surface is not underneath S_{out} , and (3) cutter body is not in touch with S_S . That is, for LG checking, S_D should be used as the part surface; for RG checking, S_{out} should be used as the part surface; while for GC checking, S_S should be used as the part surface.

From Eqs. (2.1) and (2.2), we can see that S_{out} and S_S can be represented by a certain offset of S_D . In the process planning for finish cut, normally the known surface is S_D . If S_D is represented by a set of discrete points $\mathbf{P}_k, \{k = 1, 2, \dots, m\}$, other surfaces, e.g., S_{out} can be represented by a corresponding set of points \mathbf{P}_{outk} , which is given as

$$\mathbf{P}_{outk} = \mathbf{P}_k + \mathbf{n}_k \tau_{out} \quad (2.3)$$

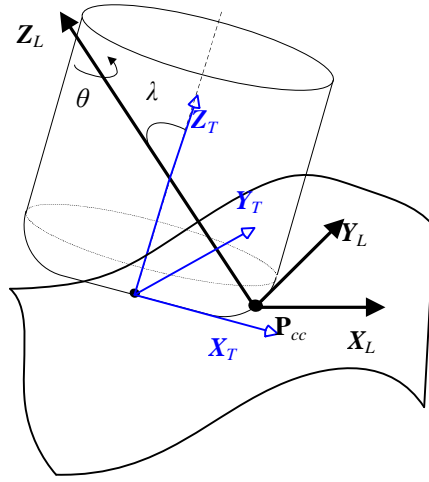
where \mathbf{n}_k is the unit surface normal vector at point \mathbf{P}_k . similarly, S_S can be obtained in this way. Using offset points to approximate the offset surface may result some self-intersections at certain portions, several studies have been conducted to calculating an accurate offset surface (Aomura and Uehara, 1990; Sun *et al.* 2004). On the other hand, it is worth mentioning that although the self-intersection curves may exist, it would not cause significant effect on the A-map calculation due to the fact that the interference is checked between the cutter and each individual point, and not the approximated offset curves.

In the following section, the previously developed A-map construction algorithm for a cutter to a point on a surface is extended by considering the profile tolerance and the cutter length in each type of interference avoidance.

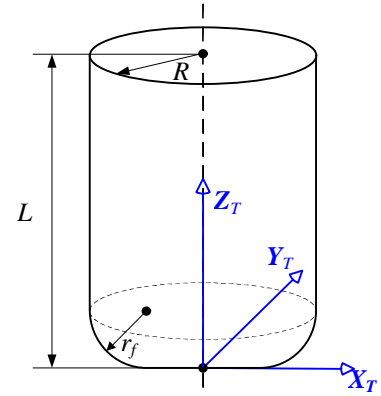
2.4 The Improved A-map Construction Algorithm

A part surface is assumed to be represented by a set of NURBS patches with C^2 continuity. There are 3 coordinate frames used in the A-map construction: *machine*, *local*, and *tool*. The *machine* frame is the coordinate system determined by the machine configuration. The local frame (X_L – Y_L – Z_L), shown in Figure 2.8a, originates at the point of interest \mathbf{P}_{cc} with Z_L -axis along the surface normal vector at \mathbf{P}_{cc} , X_L -axis along the surface's *maximum* principal direction, and Y_L -axis along the surface's *minimum* principal direction. A cutter's posture is defined by an angle pair (λ, θ) ,

meaning the cutter's axis inclines counter-clockwise with λ about \mathbf{Y}_L -axis and rotates a θ about \mathbf{Z}_L -axis. The *tool* frame (\mathbf{X}_T - \mathbf{Y}_T - \mathbf{Z}_T) is defined with its origin at the cutter bottom centre and \mathbf{Z}_T -axis along the cutter axis direction. The intersection line between the bottom plane and the plane defined by \mathbf{Z}_T -axis and \mathbf{P}_c defines the \mathbf{X}_T -axis that points towards \mathbf{P}_c . \mathbf{Y}_T -axis is defined by $\mathbf{Y}_T = \mathbf{Z}_T \times \mathbf{X}_T$. Through the definition, it can be seen that θ is 0 when \mathbf{X}_L -axis and \mathbf{X}_T -axis are co-planar, and $\lambda = 0$ when \mathbf{Z}_L -axis and \mathbf{Z}_T -axis are parallel.



(a) Local frame and tool frame



(b) Cutter geometry and tool frame

Figure 2.8 A cylindrical fillet-end cutter at \mathbf{P}_{cc} in the *local* frame and *tool* frame

A cylindrical fillet-end cutter together with the tool frame is shown in Figure 2.8b. For a specific point on the surface, the normal curvature is the curvature of an intersection curve between the surface and the plane containing the surface normal vector at the point. The maximum (κ_{\max}) and minimum (κ_{\min}) normal curvatures are called the principal curvatures. For a cylindrical fillet-end cutter, the cutting edge is located on the filleted portion of the cutter surface. At \mathbf{P}_{cc} , the cutter surface normal coincides with the part surface normal, and the principal curvatures of the cutter surface. Let $\kappa_{r_{\min}}$ be the normal curvature of cutter surface at \mathbf{P}_{cc} on the \mathbf{Y}_T - \mathbf{Z}_L plane

and $\kappa_{t\max}$ be the principal curvature of the cutter surface at \mathbf{P}_{cc} on the $X_T\text{--}Z_L$ plane.

They can be expressed as (Jensen *et al.*, 2002):

$$\begin{aligned}\kappa_{t\min} &= \frac{1}{\frac{R-r_f}{\sin \lambda} + r_f} \\ \kappa_{t\max} &= \frac{1}{r_f}\end{aligned}\tag{2.4}$$

From this equation, it can be seen that as long as the R is no less than r_f , $\kappa_{t\min}$ will always be no larger than $\kappa_{t\max}$.

For A-map construction, the feasible posture ranges, also called the interference-free posture ranges, are calculated separately first and then merged to find the A-map. Among the 4 feasible posture ranges, the ML range can be obtained by directly transforming the machine axis limits from the machine frame to the local frame at \mathbf{P}_{cc} . The LG, RG, and GC ranges are identified in terms of λ at every discrete θ for the cutter at \mathbf{P}_{cc} , respectively. The details are given in the following sections.

2.4.1 Accessible range for LG avoidance

As discussed in Section 2.3.1, the accessible range for LG-free should be conducted by checking the interference between the cutter's local surface at a CC point and the local surface on S_D . Besides, the cutter length does not affect the accessible range regarding to LG avoidance.

Given a posture (λ, θ) , as shown in Figure 2.9, the normal curvature of the surface curve at the CC point (\mathbf{P}_{cc}) on S_D along any direction x_ω (the angle between x_ω and X_L -axis is ω and $0 \leq \omega \leq 2\pi$) on the tangent plane is given as,

$$\kappa_{s\omega} = \kappa_{\max} \cos^2 \omega + \kappa_{\min} \sin^2 \omega\tag{2.5}$$

where κ_{\max} and κ_{\min} are the maximum and minimum principal curvatures of local part surface at \mathbf{P}_{cc} on S_D , respectively.

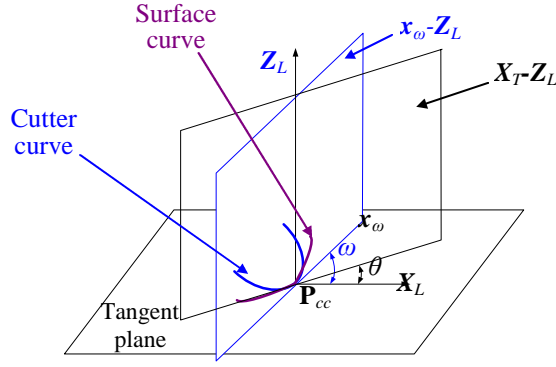


Figure 2.9 The cutter and surface curve on a normal plane containing x_ω at P_{cc}

The normal curvature of the cutter curve on the normal plane containing x_ω is:

$$\kappa_{t\omega} = \kappa_{t\max} \cos^2(\omega - \theta) + \kappa_{t\min} \sin^2(\omega - \theta) \quad (2.6)$$

To make sure that the cutter is free of local-gouging at this point, the normal curvatures of the cutter and the part surface P_{cc} in every possible direction need to be compared to ensure the prevention of local-gouging, i.e.,

$$\kappa_{t\omega} - \kappa_{s\omega} > 0 \quad (2.7)$$

By combining Eqs. (2.5) to (2.7), we have:

$$\sin \lambda > \frac{r_1(r_f \kappa_{\max} - \cos^2 \theta)}{r_f(1 - r_f \kappa_{\max})} \quad (2.8)$$

$$\sin \lambda > \frac{r_1 \kappa_{\max} (1 - r_f \kappa_{\min}) - r_1 (\kappa_{\max} - \kappa_{\min}) \cos^2 \theta}{(1 - r_f \kappa_{\min})(1 - r_f \kappa_{\max})} \quad (2.9)$$

where $r_1 = R - r_f$. Given a θ , two minimum values of λ , λ_1 and λ_2 , if there are any, can be obtained from Eqs. (2.8) and (2.9), respectively. The accessible range is therefore $[\lambda_{\theta-lg}, 90^\circ]$, where $\lambda_{\theta-lg} = \max(\lambda_1, \lambda_2)$. It can be seen that ω is not involved in the calculation of the accessible range. This analytical method can effectively and efficiently find the accessible range for a given θ , i.e., $[\theta, (\lambda_{\theta-lg}, 90^\circ)]$.

To reduce some calculation load, before starting the above calculation, one quick filtering process can be conducted. As we know, if the local surface is convex, there is no LG at this point. The condition of a local surface being convex is:

$$\begin{aligned}\kappa_{\max} &\leq 0 \\ \kappa_{\min} &\leq 0\end{aligned}\tag{2.10}$$

If the above condition satisfies, any λ is accessible, i.e., $\lambda_{\theta-lg} = 0^\circ$.

In addition, to avoid LG, the normal curvature of the cutter's local surface should be larger than the corresponding part surface's normal curvature. That means along any direction x_ω , the normal curvature of the cutter's local surface $\kappa_{t\omega}$ and the normal curvature of the local surface $\kappa_{s\omega}$ should satisfy:

$$\kappa_{t\omega} > \kappa_{s\omega}\tag{2.11}$$

As $\kappa_{t\omega}$ is always no larger than $\kappa_{t\max}$, and $\kappa_{s\omega}$ is always no smaller than κ_{\min} , one condition extracted from Eq. (2.11) is:

$$\kappa_{t\max} > \kappa_{\min}\tag{2.12}$$

By substituting Eq. (2.3) into (2.12), we have

$$\kappa_{\min} < \frac{1}{r_f}\tag{2.13}$$

In other words, to avoid LG, the condition in Eq. (2.13) must be satisfied.

In summary, to obtain the accessible range for LG free, two quick checking procedures can be conducted to reduce the computation load. Firstly, use Eq. (2.10) to check whether the surface is convex. If yes, set $\lambda_{\theta-lg} = 0^\circ$, otherwise, check if Eq. (2.13) is satisfied or not. If not, set $\lambda_{\theta-lg} = 90^\circ$. After the quick check, $\lambda_{\theta-lg}$ can be obtained as $\max(\lambda_1, \lambda_2)$ from Eqs. (2.8) and (2.9).

2.4.2 Accessible range for RG avoidance

For a given θ , let $[\theta, (\lambda_{\theta\text{-rg1}}, \lambda_{\theta\text{-rg2}})]$ represents the accessible range such that no RG occurs. Since RG refers to the case that the cutter bottom surface (including the planar and the filleted portion outside the local machining region) protrudes into the part surface, the cutter length does not affect the accessible range. As discussed in Section 2.3.2, for accessible range calculation based on RG free, the relative position between the cutter bottom surface and S_{out} should be compared.

Theoretically, the points on the cutter's bottom need to be checked with all the sampled points on S_{out} to avoid RG. Since the density of sampled points is high, to reduce the computation time, the points on S_{out} are firstly categorized into RG-free and RG-prone points. As shown in Figure 2.10a, the cutter's pivot point \mathbf{O} is located along the normal vector of the CC point \mathbf{P}_{cc} with a distance r_f from \mathbf{P}_{cc} . Since any point \mathbf{P}_{outk} on the cutter bottom surface satisfies $|\mathbf{OP}_{outk}| \leq 2R - r_f$, the RG-prone points on S_{out} should be within a distance range of $2R - r_f$ from \mathbf{O} . In addition, only those points on S_{out} that are above the tangent plane can possibly cause RG. Therefore, a RG-prone point, $\mathbf{P}_{out}(x_T, y_T, z_T)$, must satisfy $|\mathbf{OP}_{out}| \leq 2R - r_f$ and $\mathbf{P}_{cc}\mathbf{P}_{out} \cdot \mathbf{Z}_L > 0$. On the other hand, the points belonging to the local machining region around \mathbf{P}_{cc} should not be classified as RG-prone points. The points \mathbf{P}_{out} on S_{out} inside the local machining region satisfies:

$$|\mathbf{OP}_{out}| \leq r_f + \tau_{out} \quad (2.14)$$

All the points satisfying Eq. (2.14) are ruled out from RG-prone points. For the resulted RG-prone point $\mathbf{P}_{outi} | i=1, \dots, n$, where n is the total number of RG-prone points, the accessible range $(\lambda_{\theta\text{-rg1-}i}, \lambda_{\theta\text{-rg2-}i})$ is obtained firstly. The common range of all the $(\lambda_{\theta\text{-rg1-}i}, \lambda_{\theta\text{-rg2-}i}) | i=1, \dots, n$, is taken as the $(\lambda_{\theta\text{-rg1}}, \lambda_{\theta\text{-rg2}})$.

The next step is to find out under what condition a RG-prone point \mathbf{P}_{out} causes RG. Referring to Figure 2.10, let \mathbf{Y}'_T be the axis that is parallel to \mathbf{Y}_T -axis and passes

through the pivot point \mathbf{O} . When θ is fixed, the cutter can only rotate about axis \mathbf{Y}'_T . This means given a point $\mathbf{P}_{out}(x_T, y_T, z_T)$ on S_{out} , when θ is fixed, y_T is constant in the tool frame. Therefore, for each point \mathbf{P}_{out} , the accessible range is calculated on the plane $y=y_T$. Figure 2.10b shows the section curves of the cutter and the part surface with the cutter posture at $\lambda = 0$. \mathbf{O}' is the intersection point between axis \mathbf{Y}'_T and plane $y = y_T$. The section curve of the cutter bottom surface is produced with three segments: two arcs $\mathbf{T}_0\mathbf{T}_1$ and $\mathbf{T}_2\mathbf{T}_3$ corresponding to the filleted portion and one horizontal line $\mathbf{T}_1\mathbf{T}_2$ corresponding to the bottom plane of the cutter.

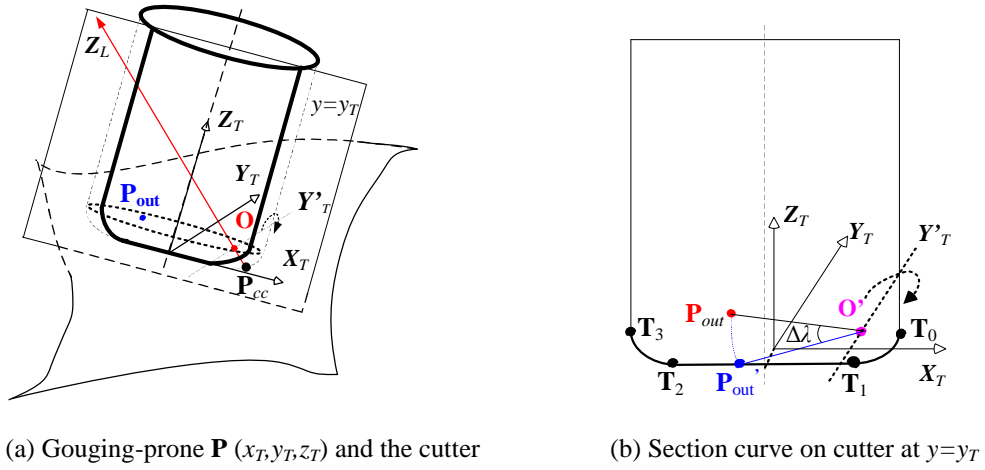


Figure 2.10 Identifying cutter posture range for rear-gouging avoidance

For each RG-prone point, if \mathbf{P}_{out} is below the section curve, the accessible posture range, in terms of λ , is $[0, 90^\circ]$, otherwise, RG occurs and λ needs to be adjusted. If λ is increased by rotating the cutter about axis \mathbf{Y}'_T , \mathbf{P}_{out} will reach the cutter bottom surface at a corresponding point \mathbf{P}_{out}' . Therefore, a minimum λ such that \mathbf{P}_{out} is on the cutter bottom surface at position \mathbf{P}_{out}' should be found. Depending on which segment of the cutter bottom \mathbf{P}_{out}' falls onto, calculation of the increment $\Delta\lambda$ that moves \mathbf{P}_{out} to \mathbf{P}_{out}' is different. For detailed calculation, refer to (Li, 2007).

2.4.3 Accessible range for GC avoidance

Let $[\theta, (\lambda_{\theta-gc1}, \lambda_{\theta-gc2})]$ represent the accessible range such that the cutter shaft does not intersect with the part surface. As discussed in Section 2.3.3, for accessible range construction for GC avoidance, the relative position between the cutter body and the stock surface S_S should be checked. Since a point on S_S collides with the cutter if it falls “inside” the cutter, the cutter length needs to be considered. The constraints of the cutter’s holder is not considered here, which could be easily incorporated into this algorithm if the geometry of the holder is given.

To reduce the computation load, the points on S_S are categorized into GC-free and GC-prone points. Let \mathbf{n}_i represents the normal vector at one specific surface point \mathbf{P}_{Si} . These points facing the cutter satisfies $\mathbf{n}_i \cdot \mathbf{P}_{cc}\mathbf{P}_{Si} < 0$ are identified as GC-prone points. Referring to Figure 2.11a, in the vicinity of the point \mathbf{P}_{cc} , \mathbf{P}_3 and \mathbf{P}_4 are GC-prone points, while \mathbf{P}_1 and \mathbf{P}_2 are GC-free points.

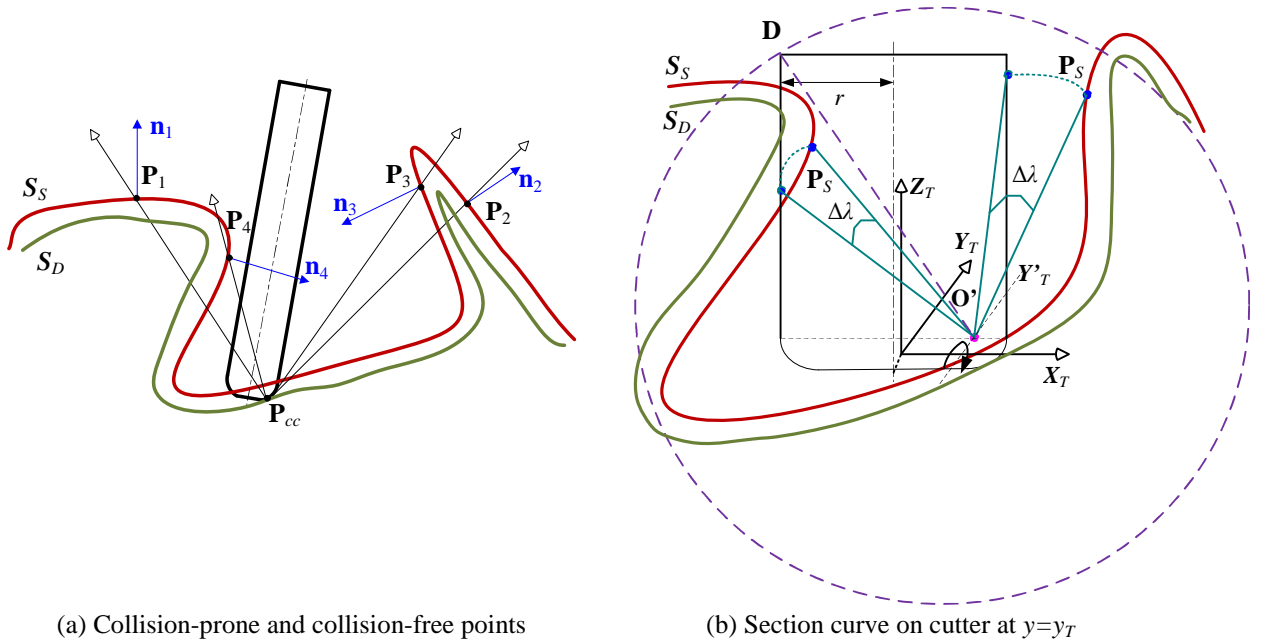


Figure 2.11 Identifying cutter posture range for global-collision avoidance

For a GC-prone point, $\mathbf{P}_S(x_T, y_T, z_T)$, like the algorithm in RG avoidance, a plane of $y = y_T$ is used to section the cutter surface (at $\lambda = 0$). The section curve is produced as shown in Figure 2.11b. The radius of the section curve is $r = \sqrt{R^2 - y_T^2}$.

Let \mathbf{D} be a point on the cutter's top edge and far from \mathbf{O}' , $|\mathbf{O}'\mathbf{D}| = \sqrt{(L-r_f)^2 + (2r-r_f)^2}$.

When rotating the cutter about axis \mathbf{Y}'_T , it can be seen that if \mathbf{P}_S is outside the circle centered at \mathbf{O}' with a radius of $|\mathbf{O}'\mathbf{D}|$, \mathbf{P}_S is GC-free point. That is, all the points in GC-prone set with $|\mathbf{O}'\mathbf{P}_S| \geq |\mathbf{O}'\mathbf{D}|$ are ruled out.

For each resulted GC-prone point, the feasible posture range needs to be found. Firstly, for a point colliding with the cutter, a minimum $\Delta\lambda$ needs to be found to avoid the collision. Referring to Figure 2.11b, point \mathbf{P}_S at the rear is inside the section curve and above the cutter bottom portion. We need to find the minimum $\Delta\lambda$ the cutter must be rotated forward to eliminate the collision. In this case, the accessible range for no GC is $[\Delta\lambda, 90^\circ]$. Besides, even if a point does not collide with the cutter at $\lambda = 0$, a minimum $\Delta\lambda$ that can reach the cutter shaft needs to be identified. Referring to point \mathbf{P}_S in the front, the minimum $\Delta\lambda$ the cutter must be rotated forward such that the point touches the cutter shaft need to be identified. In this case, the accessibility map for no GC is $[0, \Delta\lambda]$. The relative positional relationship between \mathbf{P}_S and the section curve can be categorized into five cases and the methods that handle the different cases are given in our previous study.

Using the above method, the accessible ranges for all GC-prone points can be obtained as $[\lambda_{\theta-gc1-i}, \lambda_{\theta-gc2-i}] \mid i=1, \dots, n$. The overall accessible range for no GC is $[\theta, (\lambda_{\theta-gc1}, \lambda_{\theta-gc2})]$, where $\lambda_{\theta-gc1} = \max\{\lambda_{\theta-gc1-i}, \mid i=1, \dots, n\}$ and $\lambda_{\theta-gc2} = \min\{\lambda_{\theta-gc2-i}, \mid i=1, \dots, n\}$. A complete search at θ has a computation complexity of $O(m)$, where m is the number of sampled points on the surface for interference checking. If $\lambda_{\theta-gc1} > \lambda_{\theta-gc2}$, it means that the cutter is not accessible at this θ .

2.4.4 The overall search algorithm

Based on the above discussion, taking the cutter length and profile tolerance into consideration, the overall algorithms on avoidance of LG, RG and GC are given

here. For a point on the part surface, let $[\theta, (\lambda_{\theta-lg}, 90^\circ)]$, $[\theta, (\lambda_{\theta-rg}, 90^\circ)]$, and $[\theta, (\lambda_{\theta-gc1}, \lambda_{\theta-gc2})]$ represent the accessible ranges for the avoidance of local-gouging, rear-gouging, and global-collision, respectively. The A-map is the intersection of 4 accessible posture ranges based on the respective constraints. If a common region among the three accessible ranges is available, the cutter is accessible to the point. In the overall search algorithm given below, we assume the minimum and maximum values of the tilting angle λ as 0° and 90° , respectively, while these can be generalized by using λ_{\min} and λ_{\max} instead. Similarly, the minimum and maximum values of the rotational angle θ are θ_{\min} and θ_{\max} , respectively. The algorithm is as follows:

Algorithm: A-map construction at a CC point P_{cc}

- Input:** (a) *All the points on design surface S_D $\{\mathbf{P}_k, k = 1, 2, \dots, m\}$*
 (b) *Stock surface offset ε from S_D*
 (c) *The surface out tolerance τ_{out}*
 (d) *A fillet-end cutter (R, r_f, L)*
 (e) *Tilting angle range $[\lambda_{\min}, \lambda_{\max}]$, rotational angle range $[\theta_{\min}, \theta_{\max}]$*

Output: *Cutter's A-map at P_{cc}*

BEGIN

- (1) Calculate the stock surface $S_S \{\mathbf{P}_{Sk}, k = 1, 2, \dots, m\}$
- (2) Calculate the stock surface $S_{out} \{\mathbf{P}_{outk}, k = 1, 2, \dots, m\}$
- (3) Uniformly sample $(\theta_{\min}, \theta_{\max})$ into n angles, set $i = 0$.
- (4) If $i \leq (n-1)$, $\theta_i = \theta_{\min} + (\theta_{\max} - \theta_{\min})(i/(n-1))$; otherwise, go to (9).
- (5) Find the LG free accessible range $[\theta_i, (\lambda_{\theta-lg}, \lambda_{\max})]$, using the method introduced in Section 2.3.1. If such an accessible range does not exist, $i = i + 1$, go to (4).

- (6) Find the RG free accessible ranges, from $(\lambda_{\theta-lg}, \lambda_{\max})$, by comparing the cutter surface with the part surface points on S_{out} using the method in Section 2.3.2. The common accessibility map is taken as $[\theta_i, (\lambda_{\theta-rg}, \lambda_{\max})]$, note that $\lambda_{\theta-rg} \geq \lambda_{\theta-lg}$. If such an accessible range does not exist, $i = i + 1$, go to (4).
- (7) Find the GC free accessible ranges, from $(\lambda_{\theta-rg}, \lambda_{\max})$, by comparing the cutter surface with the part surface points on S_s using the method in Section 2.3.3. The common accessible range is taken as $[\theta_i, (\lambda_{\theta-gc1}, \lambda_{\theta-gc2})]$, note that $\lambda_{\theta-gc1} \geq \lambda_{\theta-rg}$ and $\lambda_{\theta-gc2} \leq \lambda_{\max}$. If such an accessible range does not exist, $i = i + 1$, go to (4).
- (8) Output the A-map at \mathbf{P}_{cc} . Stop.

End

This overall algorithm for constructing the A-map for a cutter at a point on the part surface is called the cutter accessibility (CA) algorithm. The CA algorithm has a computational complexity of $O(km)$, where m is the number of sampled points of the part surface and k is the number of sampled θ in $[\theta_{min}, \theta_{max}]$. It can be seen that the algorithm is numerical in nature, except that the method to find the A-map for the avoidance of local-gouging is analytical. In addition, the proposed A-map calculation considers only geometric concerns. Some technical concerns, such as the preferable tilting and orientation ranges, can be incorporated by specifying $[\lambda_{min}, \lambda_{max}]$ and $[\theta_{min}, \theta_{max}]$ before the search starts.

2.5 Comparison Study

The improved A-map construction algorithm has been implemented under the following two modes: (1) CA-II with consideration of profile tolerance only and (2) CA-III with consideration of both profile tolerance and cutter length. Together with

the original A-map construction algorithm CA-I, these three algorithms are verified by a case study in this section. A comparison is conducted to show the advantages of the algorithm developed in this study.

The part model used in this case study is shown in Figure 2.12a. The design surface is of saddle shape and the stock surface (in yellow) has an offset of $\varepsilon = 0.3$ mm from the design surface, while the other parts are the un-machined stock. The outside and inside tolerance is set as 0.05mm each. The fillet-end cutter has the following dimension: $R = 6\text{mm}$, $r_f = 0.5\text{mm}$, $L = 60\text{mm}$. The ranges of tilting and orientation angles of the machine axis limits (ML) are given as $[0^\circ, 90^\circ]$ and $[0^\circ, 360^\circ]$, respectively.

At point \mathbf{P}_{cc} ($u = 0.2$, $v = 0.8$) on S_D , the feasible tilting angle (λ) ranges for LG-free, RG-free and MG-free at $\theta = 0^\circ$ are calculated using CA-I, CA-II, and CA-III separately. The results are listed in Table 2.1.

Table 2.1 A-map comparison among CA-I, CA-II, and CA-III

	λ range (LG – free)	λ range (RG – free)	λ range (GC – free)	Overall range
CA-I	[0.59, 90]	[2.25, 90]	[70.58, 79.57]	[70.58, 79.57]
CA-II	[0.59, 90]	[2.52, 90]	[70.58, 79.38]	[70.58, 79.38]
CA-III	[0.59, 90]	[2.52, 90]	[61.97, 82.64]	[61.97, 82.64]

From Table 2.1, it can be seen that all 3 λ ranges for LG-free are the same as they all use S_D for LG checking. For RG-free λ range, the two $\lambda_{\theta-rg}^{II}$ from CA-II and $\lambda_{\theta-rg}^{III}$ from CA-III are the same but larger than $\lambda_{\theta-rg}^I$ from CA-I. This is consistent with the fact that S_{out} is used in CA-II and CA-III for the search of $\lambda_{\theta-rg}$, while in CA-I, S_D is used. Furthermore, for the GC-free λ range, the analyses are categorized into two sets: comparison between CA-I and CA-II and that between CA-II and CA-III.

(1) GC results analysis between CA-I and CA-II

The smallest accessible tilting angle $\lambda_{\theta-gc1}^{II}$ from CA-II is the same as $\lambda_{\theta-gc1}^I$ from CA-I, while $\lambda_{\theta-gc2}^{II}$ from CA-II is smaller than $\lambda_{\theta-gc2}^I$ from CA-I. This is due to the fact for the search of $\lambda_{\theta-gc1}$, the non-machined stock surface is the only effective obstacle; while for the search of $\lambda_{\theta-gc2}$, \mathcal{S}_S is the effective obstacle for CA-II and \mathcal{S}_D for CA-I. For further verification on $\lambda_{\theta-gc2}$, a posture with λ close enough to $\lambda_{\theta-gc2}^I$ and also satisfies $\lambda_{\theta-gc2}^{II} < \lambda < \lambda_{\theta-gc2}^I$ (e.g., $\lambda = 79.5^\circ$) is chosen. The graphic verification is conducted with this posture as shown in Figure 2.12a (note the cutter length is displayed as infinite). An enlarged view of the critical portion \mathcal{A} in Figure 2.12a is shown in Figure 2.12b. It can be easily seen that the cutter collides with \mathcal{S}_S , but not \mathcal{S}_D . This is also consistent with the calculated ranges from CA-I and CA-II, respectively.

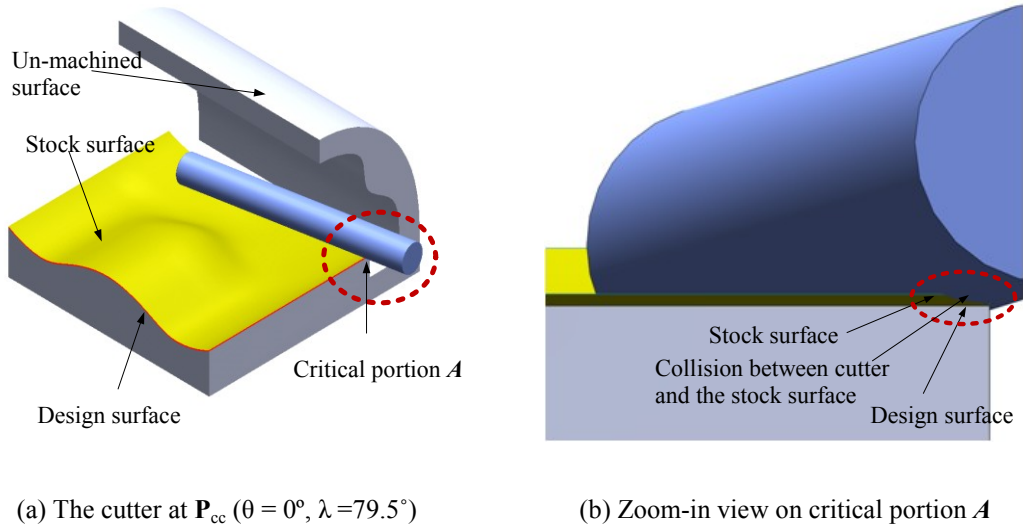
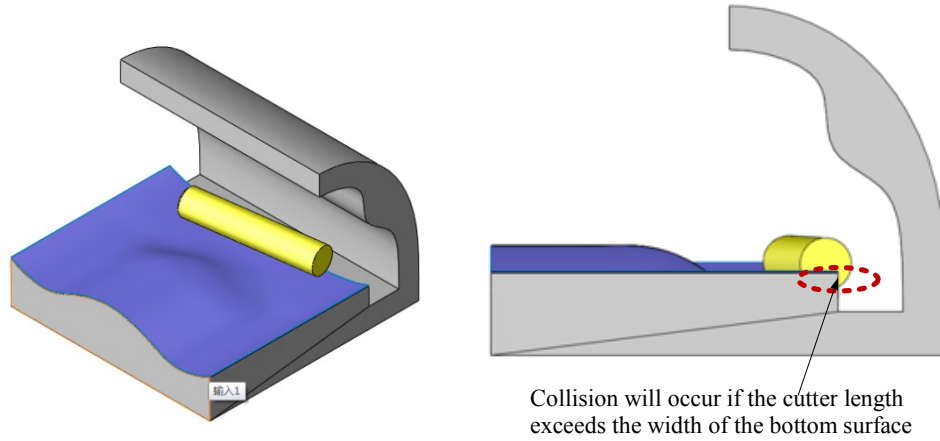


Figure 2.12 A-map results comparison (CA-I and CA-II) at \mathbf{P}_{cc} ($\theta = 0^\circ$)

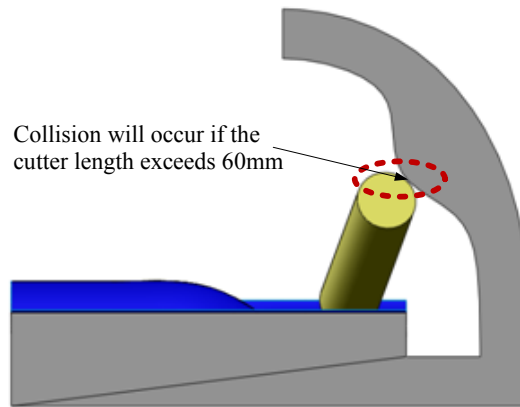
(2) GC results analysis between CA-II and CA-III

The GC-free λ range from CA-III is much wider than that from CA-II. This is due to the fact that the cutter length is considered in CA-III but treated as infinite in CA-II. For verification on $\lambda_{\theta-gc1}$, a posture with λ close enough to $\lambda_{\theta-gc1}^{III}$ and also satisfies $\lambda_{\theta-gc1}^{III} < \lambda < \lambda_{\theta-gc1}^{II}$ (e.g., $\lambda = 62^\circ$) is selected. The graphic view for this posture is given in Figure 2.13a. It can be easily seen that the cutter has no collision

with the part surface, but will collide with un-machined stock on the bottom surface if the length exceeds the width of the bottom stock surface. For verification on $\lambda_{\theta-gc2}$, a posture with λ close enough to $\lambda_{\theta-gc2}^{III}$ and also satisfies $\lambda_{\theta-gc2}^{II} < \lambda < \lambda_{\theta-gc2}^{III}$ (e.g., $\lambda = 82^\circ$) is selected. The graphic view for this posture is given in Figure 2.13b. It can be easily seen that there is no collision, but the cutter will collide with top stock if the length exceeds 60mm.



(a) The cutter with $L = 60\text{mm}$ at \mathbf{P}_{cc} ($\theta = 0^\circ$, $\lambda = 62^\circ$)



(b) The cutter with $L = 60\text{mm}$ at \mathbf{P}_{cc} ($\theta = 0^\circ$, $\lambda = 82^\circ$)

Figure 2.13 A-map results comparison (CA-II and CA-III) at \mathbf{P}_{cc} ($\theta = 0^\circ$)

2.6 Summary

In this section, the A-map concept and the previously developed A-map construction method are briefly introduced. The main problems with the original algorithm have been identified as the selection of the proper “part surface” for the calculation of feasible posture ranges for the avoidance of different interferences. As a result, the surface profile tolerance and cutter length are taken into consideration in the newly improved A-map construction algorithm. This consideration is designed to rectify the errors in A-map construction by achieving more accurate results. A single case study has been conducted and the results show the effectiveness of the improved algorithm.

CHAPTER 3

A-MAP APPLICATION FOR 5-AXIS MULTI-CUTTER SELECTION

Compared to single-cutter machining, the application of multiple cutters can produce much shorter tool-paths and hence greatly reduce the machining time. In our early work (Li, 2007), an algorithm has been developed to select the optimal single-cutter to finish the whole of a given sculptured surface by considering gouging and collision problems. In this study, this method is extended to select an optimal multi-cutter set which are utilized to finish different regions of the surface for 5-axis end-milling (finish cut) of sculptured surface. For a given surface, the feasible cutters, that form all the *possible multi-cutter sets*, are firstly identified by analyzing all the cutters' accessibility to the surface. The *candidate multi-cutter sets* are then extracted from all the possible multi-cutter sets by keeping every cutter's actual cutting region sufficiently large. Based on a proposed method for estimating the cutting efficiency without generating the actual tool path, the optimal multi-cutter set with the maximum cutting efficiency is selected.

3.1 Background

In the final stage of the sculptured surface machining (finish cut), according to the number of cutters used, there exist two types of machining: single-cutter machining and multi-cutter set machining. In the former, normally the largest accessible cutter, which can traverse the whole surface without any interference, is

used. In reality, this largest accessible cutter's size is always comparably small, which results in long tool paths and low cutting efficiency. The latter, which utilizes large cutters to machine their accessible surface regions and a small cutter to finish the remaining region, is possible to overcome this drawback. In addition, nowadays, with the availability of high speed automatic tool change mechanisms on modern CNC machines, tool changing can now be achieved within seconds and the once costly tool change time penalty in multi-cutter machining is greatly reduced. The use of multiple tools thus has become more practical.

Cutter selection lies in the heart of the process planning. Most of the available CAM systems require the user to select appropriate cutters, but it is nearly impossible for a user to determine what may constitute an optimal cutter combination for a given surface and the verification process is generally expensive and time-consuming. To avoid potential problems associated with gouging and collision, the user is often forced to make a rather conservative choice that results in low machining efficiency and high production cost.

Over the years, there has been much reported work on single and multi-cutter selection for machining sculptured surfaces. For multi-cutter selection, the reported work mainly focused on 3-axis machining. In 1999, Yang and Han proposed an approach to select optimal multi-cutter set for 3-axis freeform surface machining. The optimal cutter set is identified as the one that minimizes the overall machining time. D'Souza *et al.* (2004) developed an algorithm to select optimal cutter sequence for 3-axis free-form pockets roughing. The optimal cutter sequence is defined as the one that can achieve minimum cost of machine usage and tool usage. The machine usage cost is related to total processing time, while the tool wear cost is indicated by the ratio of total actual cutting time to tool life. It can be seen that the cutting time is a

major reference in the selection of the optimal multi-cutter set. This is quite reasonable as the cutting efficiency is one of the major concerns in most of the machining. However, both of the aforementioned methods evaluate the machining time by generating the actual tool-paths for each cutter combination, which results in heavy computation load. An alternative way is found in Sun *et al.* (2001) and Yao *et al.* (2003), which approximate the actual cutting time by the ratio of the area or the perimeter of the accessible region over the cutting width between pathes. This method can reduce the computation load significantly. In summary, research on multi-cutter selection for sculptured surface has been carried out extensively and the proposed algorithms are quite comprehensive and effective. However, these algorithms are limited to 3-axis machining, which, owing to the 2 additional rotational degrees in a 5-axis machine, cannot be directly extended to 5-axis sculptured surface machining.

Elber (1995) presented an algorithm of surface decomposition for multi-cutter machining in both 3-axis and 5-axis modes. The surface is decomposed into convex, concave and saddle regions based on curvature. For the convex area, a flat-end tool is suggested, while on the other regions, a ball-end cutter is suggested. The work focuses on the decomposition of part surface and there is a lack of algorithm details for automatic selection of cutter sizes.

On the other hand, previously reported work on automatic cutter selection for 5-axis sculptured surface machining focuses on single-cutter selection. The general idea is to find the “largest” cutter that can travel the whole surface without interference. Here, the “largest” cutter refers to the cutter with maximum cutting efficiency. Lee and Chang (1996) proposed a cutter selection algorithm of flat-end cutter by calculating the maximum effective cutting radius at every sampled point. At each sampled point, the feasibility cone is firstly constructed to obtain the feasible

range of the incline angle and tilt angle. The feasible angle range is then sampled and evaluated to find the effective cutting radius range. A feasible cutter is identified if at every sampled point, there exists an effective cutting curvature larger than local surface curvature. Jensen *et al.* (2002) developed a cutter selection algorithm for fillet-end cutters based on curvature matching machining, in which gouging and collision are considered. The algorithm is effectively trial-and-error in nature. It starts with the largest cutter in a tool database. Beginning with the first point in the sampled data set and the feeding direction, a tool interference detection and correction algorithm is applied to find an interference-free orientation within the machine limits. If at one specific point no such orientation is available, another cutter with larger minor radius or smaller major radius is selected to repeat the checking algorithm. To a certain extent, this method still follows the tool-path generation process.

A single-cutter selection algorithm was developed in our previous work (Li, 2007) that is able to find the optimal fillet-end cutter for 5-axis machining of sculptured surfaces. The part surface is firstly sampled into a set of discrete points. The A-map for a given cutter is then constructed at every sampled point without considering the feeding direction. The largest cutter that has a non-empty A-map at all the sampled points is selected from the cutter library as the optimal one. This method is effective but only one cutter is selected to finish the whole surface, which does not make full use of larger cutters for better efficiency. Furthermore, the ranking (from “large” to “small”) of the feasible cutters are based on a simple heuristic, i.e., a cutter with larger *major radius* is considered “larger” and in case of equal *major radius*, the cutter with smaller *minor radius* is considered “larger”. As an extension of this single-cutter selection algorithm, this chapter presents a heuristic-based approach to

automatically select the optimal multi-cutter set that results in minimum machining time and an acceptable level of accuracy for 5-axis sculptured surface finish cut.

The task of multi-cutter selection can be defined as “given a design surface, a 5-axis machine, and a list of cutters, find the optimal multi-cutter set that can jointly traverse the entire surface without interference”. Based on the definition, three steps are involved in the proposed approach. Firstly, the feasible cutters are extracted from the given cutter library based on their accessibilities to the surface. Secondly, from the feasible cutters, the candidate multi-cutter sets are constructed by setting a lower-limit for each cutter’s accessible area. Thirdly, a heuristic-based method for 5-axis machining efficiency estimation is proposed. With this method, the multi-cutter set with the highest cutting efficiency is selected to finish the entire surface. In addition, the cutting regions for each cutter in the optimal cutter set generated during this selection process are recorded for further tool path generation.

3.2 Identification of Feasible Cutters

In multi-cutter machining, for each cutter, the surface might be divided into accessible and inaccessible regions according to its accessibility to the surface. Therefore, based on their A-maps at all the sampled points, the cutters in the library are grouped into 3 categories: (1) *accessible* cutters that have non-empty A-maps at every sampled point, (2) *partially accessible* cutters that have non-empty A-maps at some sampled points, and (3) *non-accessible* cutters that have no non-empty A-maps at any of the sampled points. To finish the whole surface, a valid multi-cutter set must contain at least one accessible cutter. On the other hand, as the partially accessible cutters are generally larger than the accessible cutters, it is preferable to have some partially accessible larger cutters in the cutter set besides the accessible cutters. Therefore, in any feasible cutter list, we only retain one feasible cutter, which is the

largest cutter that can finish the whole surface without interference, named T_a . All the partially accessible cutters are also included in the feasible cutter list. The feasible cutters can be represented as $L = \{T_1, T_2, \dots, T_i, \dots, T_n, T_a\}$ in which the cutters are ranked from large to small, and n is the total number of partially accessible cutters.

For a partially accessible cutter, its accessibility property is represented by the *accessible regions (ARs)* on the surface. However, for sculptured surface machining, it is a highly challenging work to directly identify the *ARs* from the parametric represented surface due to two facts: (1) expensive numerical search for feature curve extraction from parametric surface and (2) complication in finding boundary curves by merging the feature curves (Park and Choi, 2001). To avoid these difficulties in *AR* identification, one commonly used approach is to represent the sculptured surface with a discrete non-parametric format, i.e., sampled points representation. Actually, during A-map construction, the surface is already sampled into a set of point, the cutter's *ARs* can thus be approximated by the *accessible points (APs)*. It is worth mentioning that for a single cutter, its *ARs* may contain more than one region, each of which is called an *AR*. The following section presents the algorithm on identifying *ARs* for a cutter on a sampled point represented surface.

3.3 Cutting Region Allocation for a Feasible Cutter

The problem of identifying *ARs* of a cutter is defined as “*given a cutter, a set of sampled points (representing the surface), and the cutter's A-map at each sampled point, obtain the ARs for this cutter*”. The proposed method contains three steps. Firstly, the *APs* are identified from the A-maps of the cutter at the sample points. The *APs* next to the inaccessible points are labeled as *boundary accessible points (BAPs)*. Secondly, from the *BAPs*, the *boundaries (BDs)* of each *AR* are constructed. Finally,

each *AR* is identified by analyzing the inclusion of the *BDs* between the accessible and inaccessible regions.

3.3.1 Boundary tracing

Over the years, a variety of algorithms have been developed to identify the boundaries for different surface representations. For parametric surfaces, Lee and Chang (1992) proposed a method to obtain the boundary by linking the intersections between horizontal cutting planes and sculptured surfaces directly. This method is straightforward in concept. However, the computation for surface-surface intersection is quite time-consuming, especially when several surfaces are described with different geometric representations. In addition, extra efforts are needed to merge the intersection curves in a correct order when they are discontinuous. For a Z-map model, several studies have been conducted for boundary extraction. Choi and Jerard (1998) developed a technique for finding boundary contours by intersecting the CL-surface with horizontal cutting planes. In their approach, the grid points belonging to the cutting area in a cutting layer are obtained first, and the boundary contour is obtained from a boundary tracing operation according to the relationship of grid points on the boundary. This approach is simple and stable. Nevertheless, to make sure the boundary contours be tracked correctly, much effort needs to be put on the identification of the boundary grid points. This results in heavy computation load. In the method proposed by Park and Choi (2001), the Z-map model is firstly converted to a binary image. A run-length code is then constructed by coding the consecutive pixels with the same property along a row. The boundaries are finally extracted from the run-length coded binary image through a recursive function and each boundary is labeled with a unique boundary index. The algorithm has advantages over the prior algorithms in terms of efficiency and simplicity: the time complexity is $O(n)$, where n

is the number of runs. However, in the algorithm, the start node of the recursive function must be the left-most node among unlinked left-nodes with the lowest row index. This results in low flexibility. Yang *et al.* (2007) extended this algorithm by removing the restriction on the start node.

In this study, the aforementioned boundary tracing algorithm is adopted in developing an algorithm to trace *BDs* from the *APs*. For a given cutter, the input is a point represented sculptured surface and the A-map information for the cutter at each sampled point. The output is the boundaries' information for all interference-free regions. Two steps are involved in the whole algorithm: (A) Construct run-length code based on the accessibility information, (B) Extract *BDs* represented by linked *BAPs* using the boundary tracing algorithm.

(A) Construct the run-length code

Based on the A-map information, the sampled points are classified into two kinds: *object* point and *background* point. A point with non-empty A-map is classified as object point; while a point with empty A-map is classified as background point. The binary image of the surface is constructed by setting object point as “1” and background point as “0”. Figure 3.1 shows an example of a binary image, with the iso-parametric view in Figure 3.1a and 2D view in Figure 3.1b, where the yellow and red pixels represent background points and object points, respectively.

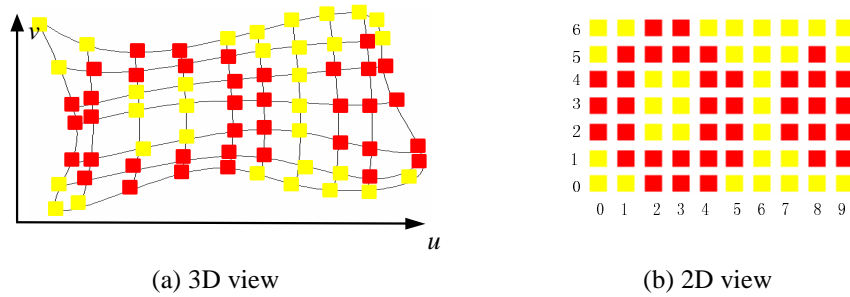


Figure 3.1 Binary image (red: *object* points; yellow: *background* points)

In the field of image processing, a *run* refers to the consecutive pixels with the same property along a row, e.g., the consecutive object points along a row. Each run can be represented by a unique number called *Run-Index* (e.g., R1), which is sequentially assigned along u and v directions. Besides, each run comprises a *Left-Node* and a *Right-Node*, where *Left-Node* represents the left end of a run and *Right-Node* the right end. From this definition, it can be seen that each *BAP* can be either a *Left-Node* or a *Right-Node*. Intuitively, each node has three basic elements: *Node-Index*, *Row-Index* and *Column-Index*. *Node-Index* is similar to *Run-Index*, which is sequentially assigned along u , v direction and used to uniquely represent a node. For example, N6(0, 2) represents a node with *Node-Index* 6, *Row-Index* 0, and *Column-Index* 2. In addition, to conduct the boundary tracing, some connectivity information is also added in the data fields of each node and run. The data field of a node is as follows:

Node = {*Node-Index*, *Row-Index*, *Column-Index*, *Prev-Node*, *Next-Node*, *Link-Node*, *Boundary-Index*, *Area-Index*}, where *Prev-Node* and *Next-Node* are the previous node and next node with the same *Row-Index* of the current node. *Link-Node* is used to represent the node that connects to the current node in the boundary tracing. *Boundary-Index* and *Area-Index* are used to record the boundary and region that contains the current node. Note, the initial value of *Link-Node*, *Boundary-Index*, and *Area-Index* are null as these values are obtained during the boundary tracing and area construction algorithms.

The run-length code refers to coding the consecutive pixels with the same property along a row. Let *Prev-Run* and *Next-Run* be the previous run and next run within the same row of the current run. To further describe the boundary tracing algorithm, some definitions are given first as follows.

Definition 1 Connectivity: If two runs belong to two neighboring rows and share at least one column, then they are called “*Connected*”, otherwise are “*Un-Connected*”.

Definition 2 *Up-Run* and *Down-Run*: Suppose Run1 and Run2 are two connected runs with Run1’s *Row-Index* smaller than Run2, if Run1 and the *Next-Run* of Run2 are not connected, then Run1 is called as the *Down-Run* of Run2, if Run1 and the *Prev-Run* of Run2 are not connected, then Run2 is called as the *Up-Run* of Run1.

The data field of a run is thus given as:

Run = {*Run-Index*, *Left-Node*, *Right-Node*, *Prev-Run*, *Next-Run*, *Up-Run*, *Down-Run*}.

For the example given in Figure 3.1, Table 3.1 shows the basic information for each run, including *Run-Index*, *Left-Node (LN)* and *Right-Node (RN)*. The detailed information for each node is also presented, including *Node-Index*, *Row-Index* and *Column-Index*. Table 3.2 gives the connectivity information for each run, including: *Prev-Run (PR)*, *Next-Run (NR)*, *Up-Run (UR)*, *Down-Run (DR)*.

Table 3.1 Basic information for Run

Row	Run	LN	RN	Run	LN	RN	Run	LN	RN
6	R14	N28(6,2)	N29(6,3)						
5	R12	N24(5,1)	N25(5,4)				R13	N26(5,8)	N27(5,8)
4	R9	N18(4,1)	N19(4,5)	R10	N20(4,8)	N21(4,9)	R11	N22(4,1)	N23(4,5)
3	R6	N12(3,0)	N13(3,1)	R7	N14(3,4)	N15(3,5)	R8	N16(3,7)	N17(3,9)
2	R3	N6(2,0)	N7(2,1)	R4	N8(2,4)	N9(2,5)	R5	N10(2,7)	N11(2,9)
1	R1	N2(1,1)	N3(1,5)				R2	N4(1,8)	N5(1,9)
0	R0	N0(0,2)	N1(0,4)						

Table 3.2 Connectivity for Run

Row	Run	PR	NR	UR	DR	Run	PR	NR	UR	DR	Run	PR	NR	UR	DR
6	R14	Null	Null	Null	R12										
5	R12	Null	R13	R14	R9						R13	R12	Null	Null	R11
4	R9	Null	R10	R12	R6	R10	R9	R11	Null	R7	R11	R10	Null	R13	R8
3	R6	Null	R7	R9	R2	R7	R6	R8	R10	R4	R8	R7	Null	R11	R5
2	R3	Null	R4	R6	R2	R4	R3	R5	R7	R1	R5	R4	Null	R8	R2
1	R1	Null	R2	R3	R0						R2	R1	Null	R5	Null
0	R0	Null	Null	R1	Null										

(B) Boundary tracing algorithm

Based on the above definitions, the process of linking the *BAPs* to represent the *BD* is converted to finding the connectivity of the nodes and runs. A recursive method to trace the boundary is presented here.

Algorithm: Extracting the boundaries

Input: (a) *A set of sampled points representing a NURBS surface region*

(b) *Cutter's A-map at each point*

Output: *All the BDs with each BD represented by the connected BAPs*

BEGIN

- (1) Identify the *APs* from the *A-maps*.
- (2) Construct the run-length code from the *APs*.
- (3) Set $i = 1$.
- (4) Select unlinked node as an initial node (*Init-Node*). Set *Init-Node* as *current node* (*C-Node*) and the run that contains *C-Node* as *current run* (*C-Run*).
- (5) Assign i to the *Boundary-Index* of *C-Node*. Add *C-Node* into the list of BD_i 's *BAPs* sequentially.
- (6) Obtain the *Link-Node* of *C-Node* by checking *C-Node*:

IF *C-Node* is a *Left-Node* of *C-Run*, check on the *Up-Run* of *C-Run*,

IF *Up-Run* is *Null*, set its *Next-Node* as *Link-Node*;

ELSE IF *Up-Run* and *Prev-Run* exist and they are connected,

set *Prev-Node* as its *Link-Node*;

ELSE set *Left-Node* of *C-Run* as *Link-Node*.

END IF

ELSE IF *C-Node* is a *Right-Node*, check on the *Down-Run* of *C-Run*,

IF *Down-Run* is *Null*, set its *Prev-Node* as its *Link-Node*;

ELSE IF its *Down-Run* and *Next-Run* exist and they are connected,
 set its *Next-Node* as its *Link-Node*;

ELSE set *Right-Node* of *C-Run* as its *Link-Node*.

END IF

ELSE IF

(7) Set the *Link-Node* as *C-Node*. Repeat steps (5-7) till *C-Node* is the *Init-Node*.
 This means BD_i is completely extracted.

(8) Set $i = i+1$.

(9) Follow step (4) to find a new *Init-Node*, till all of the nodes' *Link-Nodes* are
 not null, which means all the nodes are linked.

(10) Output each *BD* represented by the connected *BAPs*.

END

Figure 3.2 gives all the boundaries for input surface shown in Figure 3.1. Totally 3 *BDs* exist, which are displayed in dark red, light blue and blue colors. Take the dark red boundary for example, based on Table 3.1, the boundary is represented by the connected *BAPs*: N0 (0, 2) -> N2 (1, 1) -> N6 (2, 0) -> N12 (3, 0) -> N18 (4, 1) -> N24 (5, 1) -> N28 (6, 2) -> N29 (6, 3) -> N25 (5, 4) -> N19 (4, 5) -> N13 (3, 1) -> N7 (2, 1) -> N3 (1, 5) -> N1 (0, 4).

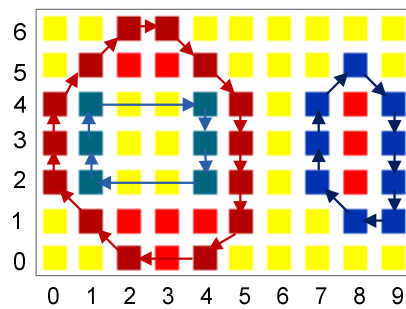


Figure 3.2 The extracted *BDs*

3.3.2 Cutting region identification

From the *BDs* information, Yang *et al.* (2007) constructed the cutting area by arranging the boundaries in a tree structure. The extracted cutting area is useful for generating contour-based tool-paths. However, there is no detail on the indication of the relations between each sampled point and the cutting area. It is necessary to know the point set for each cutting area in some cases, i.e., when identifying the cutting areas, generating the iso-planar tool paths, etc. A new algorithm is presented here to identify each *AR* together with its unique point set.

As we know, the *APs* for a cutter may form one or several *ARs*. To record the region that an *AP* belongs to, an *accessible region index* is labeled on each *AP*. On the other hand, the *inner boundary* is defined as follows: let BD_1 and BD_2 be two different boundaries in a row, if BD_1 is the only boundary such that the two of BD_2 's *BAPs* are inside BD_1 's two *BAPs*, BD_2 is an *inner boundary* (*inBD*) of BD_1 . As an example, Figure 3.3 shows a surface's boundaries and Table 3.3 gives its corresponding *inBDs* on rows 1 and 2.

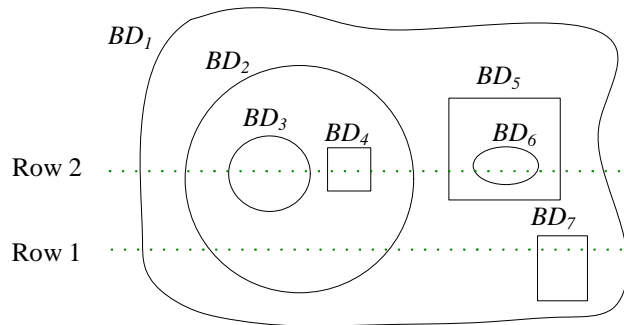


Figure 3.3 Boundaries' relationship

Table 3.3 The *inBDs* for *BDs*

	BD_1	BD_2	BD_3	BD_4	BD_5	BD_6	BD_7
<i>inBDs</i> on Row 1	$\{BD_2, BD_7\}$	<i>Null</i>	<i>Null</i>	<i>Null</i>	<i>Null</i>	<i>Null</i>	<i>Null</i>
<i>inBDs</i> on Row 2	$\{BD_2, BD_5\}$	$\{BD_3, BD_4\}$	<i>Null</i>	<i>Null</i>	$\{BD_6\}$	<i>Null</i>	<i>Null</i>

The recursive algorithm that is developed to construct the *ARs* based on the extracted *BDs* is shown as follows.

Algorithm: Extracting the ARs

Input: (a) *A set of sampled points representing a NURBS surface region*

(b) *Cutter's A-map at each point*

(c) *BDs with each BD represented by the connected BAPs*

Output: *ARs with each AR represented by APs set*

BEGIN

(1) Starting from the first row.

(2) Set the unchecked left-most boundary as *current boundary (C-BD)*.

(3) Get *C-BD*'s left *BAP* and right *BAP*. Between the two *BAPs*, construct *inBDs* for *C-BD*. Order the *inBDs* from left to right along the row as $\{inBD_1, inBD_2, \dots, inBD_n\}$, where n is the num of the *inBDs*.

If $n=0$

Assign *C-BD*'s boundary index as the area index to all the *APs* between the two *BAPs*.

ELSE

Assign *C-BD*'s boundary index as the area index to all *APs* between (*C-BD*'s left *BAP*, $inBD_1$'s left *BAP*), ($inBD_1$'s right *BAP*, $inBD_2$'s left *BAP*) ... ($inBD_n$'s left *BAP*, *C-BD*'s right *BAP*).

END IF

(4) Repeat from step (2) till all the boundaries on this row are checked.

(5) Go to the next row and repeat from step (2) till all the rows are checked.

(6) Output the *ARs* with their respective *APs* (with the same region index).

END

Figure 3.4 shows all the cutting regions for the input surface in Figure 3.1. Two regions are extracted shown in dark red and blue, respectively. It is noticed that there may exist some isolated accessible points. In addition, since approximation errors exist from surface sampling, some potential interference problems may exist on the boundary points. To avoid such problems, a heuristic is applied to refine the original accessible points: for an accessible point, if any of its neighboring points within a certain radius, called the threshold radius, is inaccessible, this point is re-classified as inaccessible. In this way, the accessible and inaccessible points are further checked and re-classified. Based on some experiments, the threshold radius is set as half of a cutter's major radius.

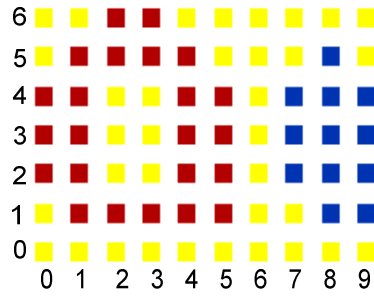


Figure 3.4 Extracted ARs

3.4 Effective Cutting Region Identification in Multi-Cutter Set

From the feasible cutters, a *possible multi-cutter* set can be formed by any number of cutters, in which T_a must be present. In a possible multi-cutter set, some of the ARs of different cutters may overlap. To maximize efficiency, it is preferable to allow the largest cutter to machine the whole of its ARs. As a result, the actual regions to be machined by any of the smaller cutters are less than its original accessible regions. The actual regions machined by a cutter in the multi-cutter set are named as *effective cutting regions (eARs)*. Similarly, the points inside each *eAR* are called the

cutter's *effective accessible points* (*eAPs*). That is, in a multi-cutter set $CSet = \{T_1, \dots, T_i, \dots, T_{m-1}, T_a\}$, in which the cutters are ordered from large to small, the *eARs* for T_i is the subtraction of the sum of *eARs* of all the larger cutters from its original *ARs*, that is,

$$T_i.eARs = T_i.ARs - T_i.ARs \cap \left(\sum_{k=1}^{i-1} T_k.eARs \right) \quad (3.1)$$

In multi-cutter set selection, cutting time is a major factor in ranking each multi-cutter set. As cutting time is largely affected by the cutter's geometry and its corresponding *eARs*, it is essential to extract the *eARs* for each cutter in a *CSet*. The problem of identification of *eAR* with its assigned cutter (termed *eAR/cutter*) is defined as “Given a multi-cutter set, a set of points representing surface, and A-maps for each cutter at each sampled surface point, obtain all *eAR/cutter* represented by a set of *eAPs*”. The overall procedure is described as follows. First of all, for each cutter in the multi-cutter set, the *eAPs* are identified from the largest cutter to the smallest one from the A-map information. The *eAPs* for the largest cutter are its original *APs*, i.e., $T_1.eAPs = T_1.APs$. The *eAPs* of the remaining cutters in the multi-cutter set can then be worked out recursively, one at a time by using Eq. (3.1). Secondly, for each cutter, the sampled surface is converted to a binary image by assigning all *eAPs* as object points and others as background points. A run-length code is then constructed by coding the consecutive pixels with the same property along a row. The *BDs* are extracted from the run-length coded information using the boundary tracing method presented in Section 3.3. Thirdly, for each cutter, all *eAR/cutter* together with a unique point set is identified by analyzing the inclusion relationships of these *BDs*. The detailed algorithm on constructing *eARs* for a cutter set is given below:

Algorithm: Extracting all the *eAR/cutter*

- Input:** (a) *A set of points representing a NURBS surface*
- (b) *A multi-cutter set $\{T_1, \dots, T_i, \dots, T_a\}$ ordered from large to small*
- (c) *A-maps at all sampled points for each cutter*

Output: *All eAR/cutter with each represented by a unique eAPs set*

BEGIN

- (1) Construct a point set named *Unassigned-Ps*. Add all the surface points into it.
- (2) Set the first unchecked cutter from the multi-cutter set as the current cutter (*C-Cutter*).
- (3) From the A-maps of *C-Cutter*, build up the set of *APs*.
- (4) Identify the *eAPs*, which is the intersection points set between *Unassigned-Ps* and *APs*.
- (5) From the *eAPs*, extract the *BDs* using the boundary tracing algorithm in Section 3.3.
- (6) From the *BDs* and the *eAPs*, identify each *eAR* represented by its unique point set using the algorithm presented in Section 3.3.
- (7) Remove all the points in *eAPs* from *Unassigned-Ps*. Repeat from step (2) until all the cutters are checked.
- (8) Output the *eAR/cutter* with the point set of *eAPs*.

END

3.5 Construction of Candidate Multi-Cutter Sets

A valid multi-cutter set can be formed by any number of cutters from the feasible cutter list, in which T_a must be present. The total number of the valid multi-cutter set can be very large and evaluating all the valid sets becomes impractical. On the other hand, although using a multi-cutter set will achieve better efficiency than a single cutter, having too many cutters in a set may cause other problems. Firstly,

inconsistent surface finishes may exist between two adjacent regions machined by two different cutters. Secondly, a cutter may have to travel a rather long distance to machine a very small eAR . The resultant time save from using multi-cutters may not be worth the cost of unproductive air travel and extra cutter change. Based on these observations, it is necessary to restrict the number of cutters in each multi-cutter set and make sure that each eAR is sufficiently large.

Here, we have proposed a heuristic to further refine the valid multi-cutter sets. For each eAR of a valid multi-cutter set, the ratio of its area to the area of the whole surface is defined as its machining ratio (MR). By setting a minimum threshold value MR_{min} to all cutters in a set except T_a , a smaller eAR is then removed from its corresponding cutter. If a cutter ends up without any eAR , it will be removed. Subsequently, the eAR s of the remaining smaller cutters will be updated. In doing so, the maximum number of cutters in a valid multi-cutter set can thus be restricted. After this refining process, a valid multi-cutter set becomes a *candidate multi-cutter set*. The heuristic-based algorithm for obtaining the candidate multi-cutter sets from all the valid multi-cutter sets is described as follows:

- (1) Set the value of MR_{min} (e.g., 20%) and calculate the area of the surface (A).
- (2) Pick a valid cutter set as the current set ($C-set$) and set the first cutter, i.e., the largest one, in $C-set$ as the current cutter ($C-cutter$). Set an empty candidate multi-cutter set $CMC-set$.
- (3) If $C-cutter$ is T_a , go to (4); otherwise, check the area of its eAR s and do the following:
 - (a) Remove the eAR s that are smaller than $(MR_{min} \times A)$.
 - (b) If the remaining eAR s are $NULL$, remove this cutter from the $C-set$. Update the eAR s of the remaining cutters in the $C-set$. Go to (d).

- (c) If the remaining $eARs$ are not $NULL$, remove this cutter from the $C-set$ and place it into $CMC-set$, together with its $eARs$.
- (d) Set the next cutter in $C-set$ as $C-cutter$, repeat (3).
- (4) Place the T_a , together with its $eARs$, into the $CMC-set$ and close the set. Remove the $C-set$ from the valid multi-cutter set pool.
- (5) If the remaining valid set pool is not $NULL$, go to (2); otherwise, output all the candidate multi-cutter sets, stop.

Actually, even before the valid cutter sets are formed, the MR_{min} can be used as a threshold criterion to disqualify any partially accessible cutter when the ratio between the sum of its ARs and the total area of the surface is less than MR_{min} .

3.6 Obtain the Optimal Cutter Set

With all the candidate multi-cutter sets, it is preferable to select the set that gives the shortest tool-path length. However, at this stage, no tool-paths are available and it is time consuming to generate the tool-paths for all the sets. Therefore, it is necessary to develop a method that is able to measure the shortest tool-path length for a cutter to machine a given surface region, without generating any tool-paths.

Given a cutter and a surface region, under the iso-planar tool-path pattern, the general depiction of the tool-path is shown in Figure 3.5a where d is step-over size, the distance between two neighbouring paths. Clearly, the total tool-path length is closely related to step-over and it is desirable to have it as large as possible within the accuracy requirement. At each CC point, the machining strip width W is defined as the distance between two neighbouring cusp height curves when a cutter is moving along a tool path. From the definition we can see that step-over is determined by the machining strip width at all the CC points. In principle, it is reasonable to assume that to maximize d , W at every CC point needs to be close to its maximum. For better

illustration, Figure 3.5b gives a section view of the cutting paths and the surface around a CC point \mathbf{P}_{cc} , in which \mathbf{f} is the cutting direction. Let W_{ave} represents the average machining strip width over all the surface points inside one eAR and $A(eAR)$ as the area. The notion of the tool-path length index (L_{index}) is introduced here as,

$$L_{index} = \frac{A(eAR)}{W_{ave}} \quad (3.2)$$

It can be seen that L_{index} serves as a close measure of the total tool-path length for the given cutter and surface region. Since $A(eAR)$ can be easily calculated from the point set of the eAR , to obtain L_{index} , the key is to estimate the W_{ave} over the eAR . A detailed description is given in the following sections.

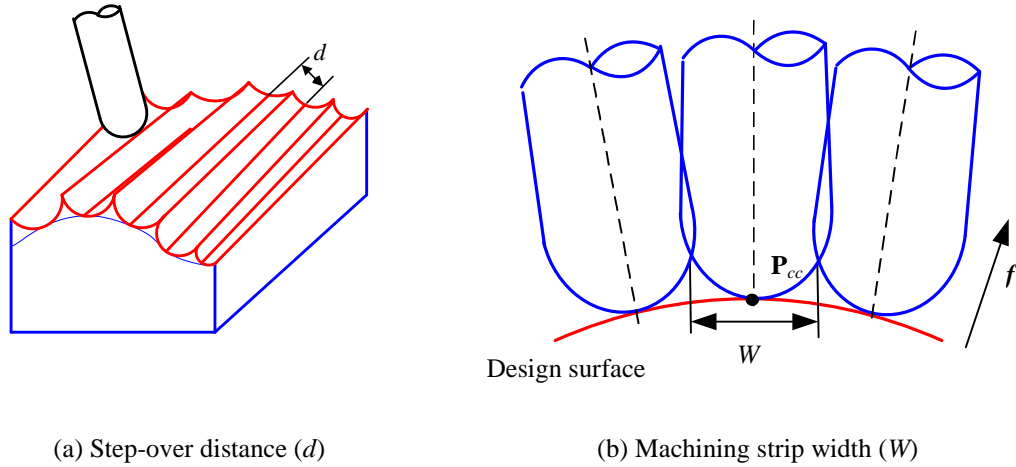


Figure 3.5 Step-over distance and machining strip width

3.6.1 Machining strip width estimation

The machining strip width is a combination function of cutter orientation, cutter geometry, surface geometry, feeding direction, and allowable surface profile tolerance (Lee, 1998). As shown in Figure 3.6, both the feeding direction \mathbf{f} and pick feed direction x_ω (orthogonal to \mathbf{f}) are given at a point \mathbf{P}_{cc} . Let ω represent the angle from X_L to x_ω and α represents the feeding angle from X_L to \mathbf{f} on the plane $\mathbf{X}_L\text{-}\mathbf{Y}_L$. From the definition it can be seen that $\alpha = \omega - 90^\circ$.

According to Euler's formula, the normal curvature of the surface curve at \mathbf{P}_{cc} on the instantaneous plane x_ω - Z_L is given as,

$$\kappa_{s\omega} = \kappa_{s\max} \sin^2 \alpha + \kappa_{s\min} \cos^2 \alpha \quad (3.3)$$

where $k_{s\max}$ and $k_{s\min}$ are the principal curvatures of the part local surface at this point.

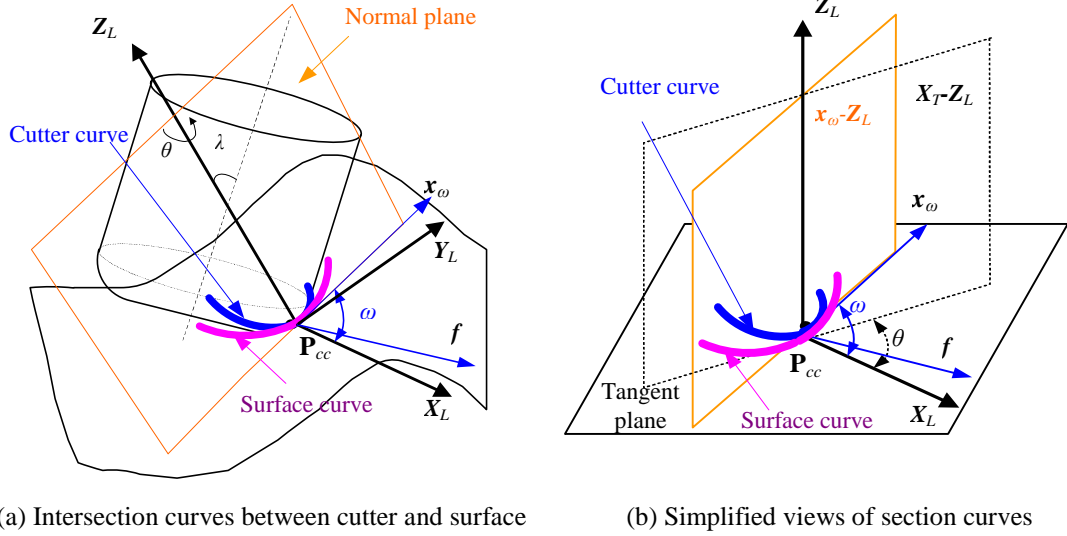


Figure 3.6 Machining strip width analysis

For a fillet-end cutter, the cutting edge is located on the filleted portion. According to Jensen and Red (2002), the fillet-end cutter bottom at a CC point can be approximated locally as a paraboloid. For a cutter having R no smaller than r_f , the approximation of the swept silhouette curve for a fillet-end cutter is obtained by projecting the paraboloid onto Y_T - Z_L plane. Since the angle between \mathbf{f} and X_T on the plane X_L - Y_L is $(\theta - \alpha)$, the normal curvature of the swept silhouette curve k_t' is obtained as:

$$\kappa_t' = \frac{\kappa_{t\min}}{\cos^2(\theta - \alpha)} \quad (3.4)$$

where $k_{t\min}$ is the principal curvature on the cutter local surface at this point and is given in Eq. (2.4). As W is defined as the width of the machined surface curve on the

instantaneous plane, by using the swept silhouette curve to approximate the cutter's effective cutting profile, W is given as (Jensen and Red, 2002):

$$W = \sqrt{\frac{8h}{\kappa_t' - \kappa_{s\omega}}} \quad (3.5)$$

where h is the surface scallop height tolerance. By substituting Eqs. (3.3), (3.4) and (2.4) to Eq. (3.5), it gives

$$W = \sqrt{\frac{8h}{\frac{1}{\left(\frac{R-r_f}{\sin \lambda} + r_f\right) \cos^2(\theta - \alpha)} - (\kappa_{s\max} \sin^2 \alpha + \kappa_{s\min} \cos^2 \alpha)}} \quad (3.6)$$

From Eq. (3.6), it can be seen that for known surface and cutter geometries, W at a CC point is the function of α , θ and λ . In the cutter selection phase, these three parameters are still unknown and have to be estimated. Therefore, some assumptions are made here based on the principle to achieve maximum machining strip width. Firstly, good cutting efficiency is possible to be achieved when the cutting direction is along the minimum curvature direction of the surface. That is when $\alpha = 0^\circ$. Secondly, for a fixed α , while keeping θ unchanged, W is decreasing with respect to the increment of angle λ . Thus, it is preferable to specify angle λ as the lower-bound at θ within accessible cutter orientation range. As θ is also unknown at this stage, the average of the minimum tilting angles over the range of θ ($Ave(\lambda_{min})$) obtained from the A-map is used to approximate the tilting angle at every CC point. Thirdly, while keeping λ unchanged, from Eq. (3.6) it can be seen that W increases with the decreasing of the difference between θ and α . Thus, it is preferable specify θ as the one closest to the feeding angle α , i.e., the cutter is oriented to face the feeding direction. This result is consistent with the findings in Pi *et al.* (1998). In summary, the maximum machining strip width at a CC point can be achieved with the following

posture assigning strategy: (1) $\alpha=0^\circ$, (2) $\lambda=(Ave(\lambda_{min}))$, and (3) $\theta=\alpha$. Accordingly, Eq. (3.6) can be written as:

$$W = \sqrt{\frac{8h}{\frac{1}{(R-r_f)/\sin Ave(\lambda_{min}) + r_f} - \kappa_{smin}}} \quad (3.7)$$

Suppose there are a total number of p sampled points in the eAR , its W_{ave} can be obtained by,

$$W_{ave} = \frac{1}{p} \sum_{k=1}^p W_k \quad (3.8)$$

3.6.2 Output the optimal cutter set

With this strip width estimation method, the L_{index} for a cutter on each of its eAR s can be obtained. For a candidate multi-cutter set, the summation of the L_{index} for each cutter with each of its eAR s is called $\sum(L_{index})$. Finally, the candidate multi-cutter set with the minimum $\sum(L_{index})$ is selected as the optimal cutter set to finish the given surface.

It is worth mentioning that $\sum(L_{index})$ is not indicative of the actual tool-path length. However, it should be proportional (somewhat) to the actual tool-path length under the maximum-machining-strip strategy in iso-planar cutting pattern.

3.7 The Overall Algorithm

The overall algorithm to obtain the optimal multi-cutter set with the shortest tool-path length is described as below:

Algorithm: Optimal cutter set selection

- Input:** (a) *A set of points representing a NURBS surface*
- (b) *A cutter lib. $L = \{T_1, T_2, \dots, T_i, \dots, T_n\}$ ordered from large to small*
- (c) *Minimum threshold value of machining region MR_{min}*

- Output:** (a) *Optimal multi-cutter set*
- (b) *All eAR/cutter with each represented by a set of points*
- (c) *A-maps at each point all the cutters in the set*

BEGIN

- (1) From large to small in the cutter library, calculate the A-maps for each cutter until the cutter T_a is found.
- (2) Form the feasible cutter set $L = \{T_1, \dots, T_i, \dots, T_a\}$
- (3) From the feasible cutter set, construct the candidate cutter set which includes the forming of eARs for each cutter in the multi-cutter set.
- (4) Calculate $\sum(L_{index})$ for each candidate multi-cutter set.
- (5) Output the optimal multi-cutter set which achieves the minimum $\sum(L_{index})$ together with all the eAR/cutter and the A-maps.

END

3.8 Examples and Discussions

The aforementioned algorithm for optimal multi-cutter set selection, including the algorithms on *BDs* tracing and extraction of *eARs*, have been implemented in Visual C++ environment and with OpenGL for display. In this section, two application examples are given. In the first case study, a benchmark part is used to verify the accuracy of the algorithms, including the verifications on the algorithm of *BDs* tracing and *ARs* identification, and the method on multi-cutter selection. The second example is more complex and is used to justify the needs of adding threshold radius when extracting *APs* and demonstrate the capability of the optimal multi-cutter selection algorithm. Table 3.4 shows the cutter library used for all these three examples, which consists of 9 fillet-end cutters listed from large to small.

Table 3.4 Library of fillet-end cutters

Cutter index	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
Major radius R (mm)	12	10	8	6	4	3	2	1.5	1
Minor radius r_f (mm)	0.5	0.5	0.5	0.5	0.5	0.2	0.2	0.2	0.2
Length L (mm)	85	80	70	60	50	45	45	45	45

3.8.1 Case study 1: a benchmark part

In this case study, a simple composite surface shown in Figure 3.7a is employed as a design surface to be machined. It consists of planar patches, cylindrical (convex) patches, and two critical concave features F1 and F2. Figure 3.7b is the y-section view of the surface. F1 and F2 are specially designed cylindrical patches to induce less accessibility as shown in Figure 3.7c and d, respectively. The geometric simplicity of the surface makes it possible to identify all the ARs analytically.

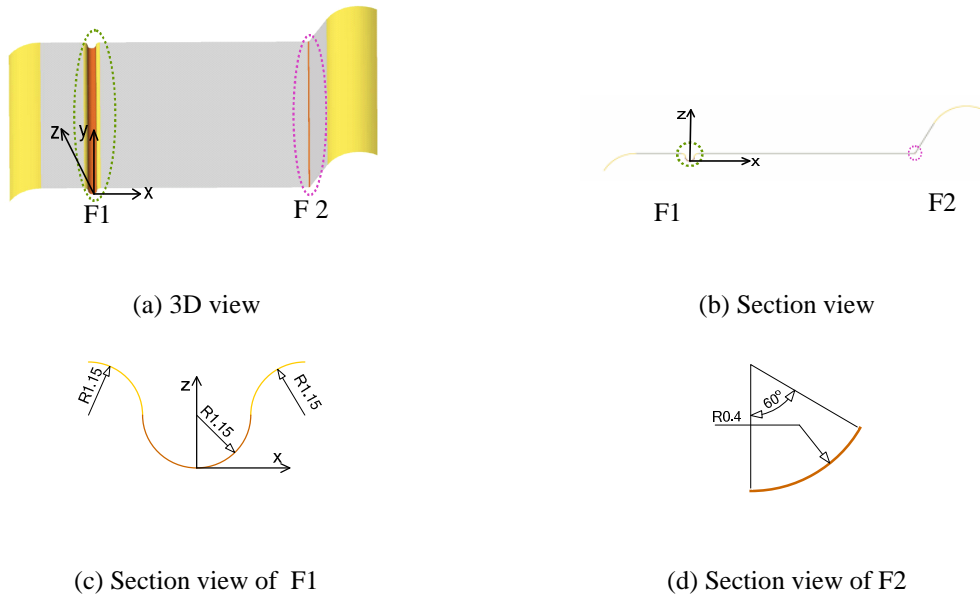


Figure 3.7 Case study 1: surface geometry

(A) Verification on identification of ARs

The part surface was first sampled uniformly along u and v into 201×201 points. At each sampled point, A-map evaluation was conducted for every cutter by avoiding LG, RG and GC. Based on the A-map information, the AP s for each cutter were extracted. AR s were then identified from the AP s. The ratio between AR s and the overall area (A) for each cutter is given in Table 3.5. It can be seen that $T_1 \sim T_8$ are partially accessible cutters, while T_9 is the only accessible cutter. On the other hand, the AR s for $T_1 \sim T_5$ are the same and AR s for $T_6 \sim T_8$ are also the same.

Table 3.5 Case study: cutters' accessible information (ARs/A)

Cutter index	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
ARs/A (%)	64.68	64.68	64.68	64.68	64.68	75.62	75.62	75.62	100

For better illustration, the AR s of T_1 and T_6 are shown in Figure 3.8a and b, respectively. The boundaries for the AR s are also marked. T_1 had 3 AR s and 4 BD s (parallel to y -axis), while T_6 had 2 AR s and 2 BD s (also parallel to y -axis). For all BD s, Table 3.6 shows their positional information with their x and z coordinates in this case, which will be compared with the boundaries to be obtained through analytical means.

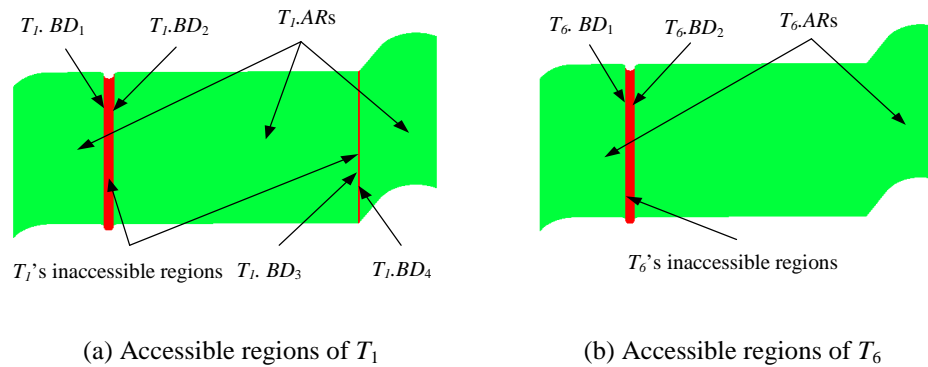


Figure 3.8 Case study 1: AR s for T_1 and T_6

Table 3.6 Case study 1: points coordinates (x, z) on each boundary

	$T_1.BD_1$	$T_1.BD_2$	$T_1.BD_3$	$T_1.BD_4$	$T_6.BD_1$	$T_6.BD_2$
Numerical (our algorithm)	(-1.231, 1.573)	(1.231, 1.573)	(59.769, 2.3)	(60.116, 2.5)	(-1.201, 1.488)	(1.201, 1.488)
Analytical	(-1.227, 1.563)	(1.227, 1.563)	(59.769, 2.3)	(60.116, 2.5)	(-1.201, 1.488)	(1.201, 1.488)

To check the validity of the above results on accessibility, the exact results are calculated by taking advantage of the known geometry. It is noticed that BD_1 and BD_2 are on F1 for both T_1 and T_6 ; and BD_3 and BD_4 are on F2 for T_1 only. These boundaries correspond to the postures of the cutter such that the cutter would interfere with the part surface if it goes beyond these boundaries.

F1 is made of three cylindrical patches of radius 1.15mm. The arc angles of the three patches are 90° 180° and 90° from left to right. From the bottom half-circle it can be seen that no cutter with $R > 1.15$ mm could access the bottom without gouging. This makes T_9 ($R = 1$ mm) the only accessible cutter. For the 90° patch on the left, the $BAPs$ can be found such that the cutter touches the point with tilting angle $\lambda = 0^\circ$ as shown in Figure 3.9. The position (x and z coordinates) of BAP is calculated by,

$$\begin{cases} x = -1.15 \times \cos(\arcsin \frac{1.15 + r_f}{4 \times 1.15}) + 2 \times 1.15 \\ z = 1.15 + 1.15 \times \frac{1.15 + r_f}{4 \times 1.15} \end{cases} \quad (3.9)$$

It can be seen that the position of $BAPs$ is the function of r_f only. Note that due to the symmetrical geometry of F1, the $BAPs$ on the right patch can be calculated similarly.

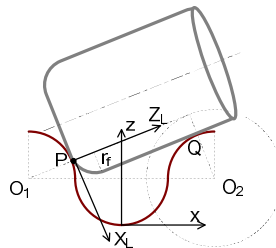


Figure 3.9 Extreme position if a cutter on F1

F2 is a cylindrical patch with a small radius (0.4mm). This means that only tools with minor radius (r_f) smaller than that, i.e., $T_6 \sim T_9$, are accessible. For the remaining cutters, the positions of the *BAPs* can also be easily obtained.

From the analysis above, we know that (1) T_9 is the only accessible cutter and (2) tools (excluding T_9) with the same r_f share the same accessibility. These are exactly in accordance with the results obtained earlier. As there are two types of r_f (0.5mm and 0.2mm) in the cutter library, the largest cutters of both types are T_1 and T_6 , respectively. Based on these geometric relationships, the theoretical values of *BAPs* for T_1 and T_6 are calculated and tabulated in Table 3.6. It can be seen they are quite close to the actual value we obtained. The slight difference is caused by resolution errors in the sampling process.

(B) Verification on selection of optimal multi-cutter set

From the geometry information, the analytical result on the optimal multi-cutter set is also studied. Firstly, as T_9 is the only accessible cutter, T_9 must be present in any valid multi-cutter set. Thus, the feasible cutter list is $L = \{T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9\}$. Secondly, as $T_1 \sim T_5$ share the same *ARs*, it is only meaningful to include one of them in the multi-cutter set. Based on their dimensions, T_1 is chosen. Similarly, T_6 is selected from $T_6 \sim T_8$ to be included in the multi-cutter. Therefore, the resulted candidate multi-cutter sets with *eARs* for each individual cutter are: (1) $\{T_1(64.68\%A), T_9(35.32\%A)\}$, (2) $\{T_1(64.68\%A), T_6(10.94\%A), T_9(24.38\%A)\}$, (3) $\{T_6(75.62\%A), T_9(24.38\%A)\}$ and (4) $\{T_9(100\%A)\}$. It is obvious that set-(1) is more efficient than set-(3) and set-(4). In addition, the *eARs* of T_6 in set-(2) are of 10.94%A. If MR_{min} is set as 20%, T_6 will be removed. In summary, the optimal multi-cutter set from the analytical view is set-(1), i.e., $\{T_1, T_9\}$.

By using our algorithm with MR_{min} set as 20%, the optimal multi-cutter set was also found to be $\{T_1, T_9\}$. Based on the tool-path length estimation model, compared to machining with a single cutter T_9 , the use of cutter set $\{T_1, T_9\}$ can reduce the machining time by 78.4%.

3.8.2 Case study 2: a general example

The design surface of the second example is given in Figure 3.10a, where the whole surface needs to be machined. The surface was sampled uniformly along u and v into 201×201 points. A-map evaluation was conducted to all the cutters at all the sampled points. It was found that the largest accessible cutter is T_8 and the feasible cutter list was thus as $L = \{T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8\}$.

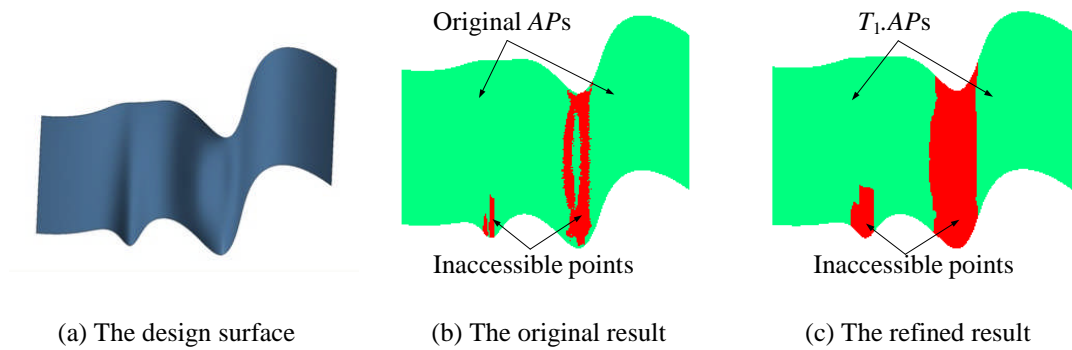


Figure 3.10 Case study 2

Firstly, we use the case of T_1 to show the necessity to refine the original APs . Figure 3.10b shows the original accessibility (APs and inaccessible points) of T_1 directly obtained from its A-maps at all the sampled points. There are totally 11 ARs and some of them are hardly visible due to their small size. In addition, some of the boundaries of the ARs are not smooth. Therefore, it is necessary to refine the original APs such that the boundaries are smoother and individual ARs are sufficiently large. The threshold radius for the refinement process is set as $0.5R(T_1)$ and the refined

result on the sampled points is shown in Figure 3.10c. There are only two *ARs* left and the boundaries are much smoother.

By setting $MR_{min} = 20\%$, the optimal cutter set was found as $\{T_1, T_5, T_8\}$. The *eARs* of the three cutters are shown in Figure 3.11, where $T_1.eARs$, $T_5.eARs$, and $T_8.eARs$ form 70.17%, 25.36% and 4.45% of the whole surface, respectively. Based on the tool-path length estimation model, compared to machining with a single cutter T_8 , the use of cutter set $\{T_1, T_5, T_8\}$ can reduce the machining time by 77.96%.

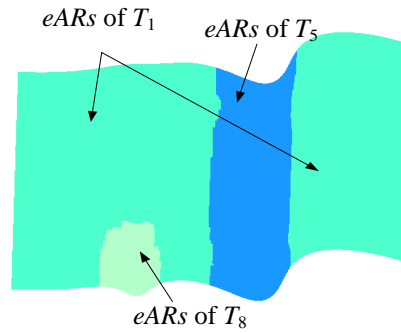


Figure 3.11 The *eARs* of T_1 , T_5 and T_8

3.9 Summary

In this chapter, a heuristic method based on geometric analysis has been developed for the selection of the optimal multi-cutter set that leads to shortest tool-path length in 5-axis finish cut of sculptured surfaces. The optimal multi-cutter set satisfies: (1) the largest accessible cutter is the smallest cutter in a valid multi-cutter set and (2) in a multi-cutter set, each cutter's *eAR* is large enough. Based on these constraints, the optimal multi-cutter set selection is identified in the following steps:

- (1) A feasible cutter list is constructed by dropping the cutters that are smaller than the largest accessible cutter.
- (2) All valid multi-cutter combinations are constructed.

- (3) The candidate multi-cutter sets are extracted by keeping each individual *eAR* sufficiently large.
- (4) The shortest overall tool-path length for each candidate multi-cutter set is estimated and the one with the minimum $\sum(L_{index})$ is selected as the optimal set.

During the multi-cutter set machining, the allocation of the cutting areas for each cutter is critical in the process planning. A comprehensive algorithm is given to identify each *eAR*/cutter together with its unique point set with the input of A-maps. Examples showed the cutting area identification algorithm is produced with good accuracy. Besides, the resulted *eAR*/cutter alongside the multi-cutter set selection process can be directly used in tool-path generation phase.

Furthermore, it is worth to mention that in the proposed heuristic, which uses MR_{min} to make sure each *eAR* is sufficiently large, reduces the surface inconsistency between two small adjacent regions machined by different cutters. In this way, the maximum number of cutters in each multi-cutter set is also reasonably restricted. Besides, the proposed 5-axis tool-path length estimation model gives a general reference to evaluate the machining efficiency while eliminating the heavy computation in actual tool-path generation.

CHAPTER 4

A-MAP APPLICATION FOR OPTIMAL 5-AXIS CUTTER LOCATION (CL) PATH GENERATION

In the process planning for 5-axis milling of sculptured surfaces, tool-path generation is the subsequent task right after cutter selection. In general, the tool-path generation task includes the selection of the cutting direction (for iso-planar path), generation of cutter contact (CC) points together with the cutter's posture at each CC point within the accuracy requirement. A CC point and the cutter posture form a cutter location (CL). To be more specific, a tool-path is called a CL path from now on. Due to the complex shape of sculptured surfaces, well-planned CL paths can significantly increase the machining efficiency with smooth dynamics. Traditionally, the CL path is generated without path optimization due to heavy computation load requirement. In this study, optimization in CL path generation is made possible due to the fact that the A-map at a CC point contains the whole solution space of cutter posture. Moreover, since the A-maps of the sample points are part of the output from the cutter-selection stage, the CL path optimization is thus possible to be achieved with reasonable computation load. This chapter presents an efficient algorithm for generating optimal iso-planar CL paths on a surface region for a single cutter by using the A-maps. The proposed optimization process includes selection of optimal cutting direction and generation of optimal CL data. In both steps, the A-maps resulted from the cutter-selection phase are reused and served as a key information source in

reducing the computation load. In this way, the process planning is achieved in an integrated and efficient manner.

4.1 Background

The process of CL path generation plays a key role in sculptured surface machining. The quality of CL paths is the main factor that determines the efficiency of machining and the quality of the final surface. In general, CL paths of high quality should be able to achieve surface accuracy, good machining efficiency, and smooth cutter dynamics. Firstly, surface accuracy means that after the machining, the deviations between the machined surface and the design surface (called shape error) are within the specified tolerance. Secondly, to achieve good cutting efficiency, the overall CL-path length should be kept as short as possible. Since machining strip width is crucial in determining the CL-path length, the maximum cutter efficiency can thus be achieved by maintaining maximum machining strip width across the CL paths. Thirdly, tool dynamics has significant impact on tool life and surface finish quality, especially as the cutter moves with high feed rates and speeds in 5-axis machining (Mahadevan *et al.*, 2003). Sharp change of cutter posture is one of the main causes of cutter dynamics, which may result in high acceleration/deceleration on the motor speed, leaving “tool marks” on the machined surface and producing out-of-tolerance areas. Since the finish cut is a delicate operation with tight accuracy requirement, it is necessary to keep the change rate of cutter orientation as small as possible.

In the currently commercially available CAD/CAM systems, due to the complexity of the cutter’s orientation in 5-axis machining, most of them still need a significant amount of time and manual effort in CL path generation, including path verification and correction (Ye and Xiong, 2008). On the other hand, a significant

amount of work has been done on developing algorithms of interference-free CL-path generation for 5-axis sculptured surface machining. In general, according to the path topology, the reported CL-path generation methods can be classified into the following types: *iso-parametric*, *iso-planar*, and others (including *iso-scallop*, *iso-curvature*, etc.), as shown in Figure 4.1.

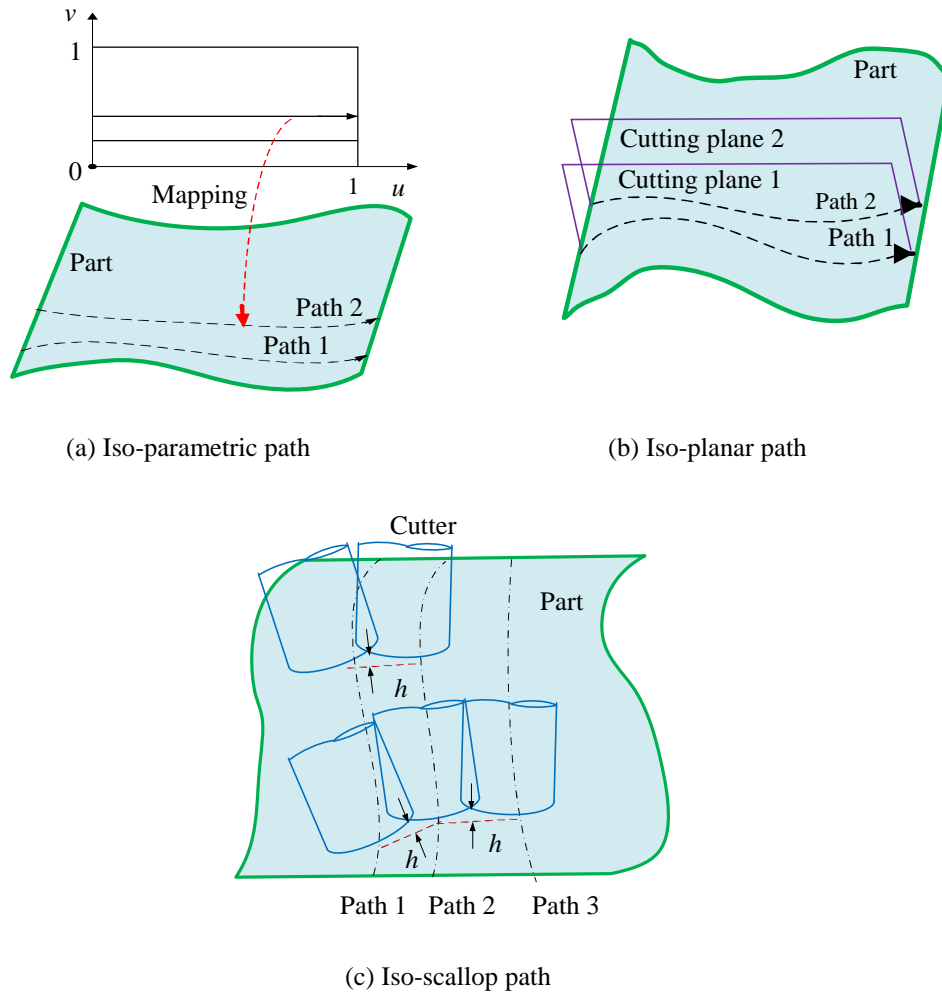


Figure 4.1 Types of tool-paths based on path topology

The iso-parametric path (see Figure 4.1a) defines the paths along the directions with constant increment values of u or v on a parametric surface. Related work on this type of CL-path generation can be found in (Li and Jerard, 1994). Since the coordinates of CC points can be directly derived from the parametric variables' values, it is the simplest method when the part surface is represented in parametric

form. However, due to the nature of iso-parametric pattern, the algorithm is valid only for parametric surfaces. In addition, the generated tool paths are often much denser in one region than others due to the non-uniform transformation between the parametric Euclidean space (Choi *et al.*, 1997).

The iso-planar path (see Figure 4.1b) defines the paths on a series of intermediate parallel planes in the Euclidean space. Iso-planar tool path is effective on sculptured surface as curves from different surface patches are joined together into a single tool path. In practice, owing to its simplicity and robustness, this path pattern is widely used. There are two types of iso-planar pattern: zigzag path and one-way path. Although zigzag-milling provides high productivity in surface machining, one-way milling is commonly used in high speed machining due to high surface quality and longer tool life (Li, 2007).

Apart from iso-parametric and iso-planar path patterns, other used patterns include iso-scallop, iso-curvature, etc. Take iso-scallop path for example (shown in Figure 4.1c, where h represents the resulting scallop height), the CL data are generated such that the maximum machining strip width can be achieved while keeping the scallop height close enough to the surface tolerance. Chiou and Lee (2002) presented a method of iso-scallop tool-path generation for multi-axis sculptured surface machining. Similar work can also be found in Lee (1998) and Li and Feng (2004). For iso-curvature tool path generation, Lee and Chang (1994) proposed an algorithm to generate the CL data such that when positioning the cutter at the CC point with the selected posture, the local curvature between the cutter surface and part surface can be matched as much as possible. Pi *et al.* (1998) developed an algorithm called grind-free tool path generation, which indeed is a combination of constant scallop height and curvature match. Theoretically, although both iso-scallop and iso-

curvature patterns can achieve shorter tool-path length, they may result in inconsistent segments for some cases.

Due to the popularity of iso-planar pattern, it is chosen for CL path generation in this study. In general, iso-planar CL path generation can be considered as a two-phase process: (1) cutting direction selection, (2) CL data generation that includes generating CC points and the cutter postures. In both phases, various optimization strategies can be applied.

Firstly, the cutting direction for iso-planar path has great impact on cutting efficiency for machining of sculptured surfaces. However, most of the reported work on iso-planar tool-path generation assumes that the feeding direction is given by the user (Li and Jerard, 1994; Pi *et al.*, 1998). This is equivalent to random assignment of cutting direction that may result in low quality of the tool path. On the other hand, research on finding optimal feeding direction for 5-axis machining is very limited. Marciniak (1987) pointed out that the maximum strip width can be obtained when the tool moves along the minimum surface curvature. Chiou and Lee (2002) determined the feeding direction aiming at maximum machining strip width by evaluating the potential machining strip width at the CC points.

Secondly, at a local surface point, a large number of orientations can be assigned to the cutter within the accessible range. Since the characteristic of the cutter orientation affects the cutting property, a rating of all possible cutter orientation is necessary to achieve the given machining strategy. There are several studies on selecting the optimal cutter posture to achieve maximum cutting efficiency. One way is to optimize the cutter orientation based on the curvature information in the CC point for maximal material removal. Jensen and Anderson (1993) determined the optimal orientation by choosing the one that results the maximum match of the cutting

tool profile to the local surface. Rao *et al.* (1996) developed a so-called “principle-axis method” for tool orientation determination at each CC point. The optimal orientation is chosen by matching the cutter profile to the local surface on the normal plane containing the surface principle curvature at each CC point. The shortcoming of both of the two algorithms is that the orientation is obtained with a fixed rotational angle, which does not make full use of the whole solution space. In addition, the avoidance of rear gouging and global collision is not incorporated in the algorithms. Recently, a new way for choosing the optimal orientation is presented by evaluating the cutting character of all the feasible orientations at a point. Chiou and Lee (2002) determined the orientation by evaluating the machining strip width along every cutting direction at each point. Similar work can be found in Lauwers *et al.* (2003). This method can achieve good cutting efficiency. At the same time, the selection of cutter posture also affects the cutter dynamics, which affects the surface quality of the machined surface. So far, there has been limited study on selecting the cutter posture to achieve smooth cutter dynamics. Jun *et al.* (2003) proposed a cutter orientation optimization method by using the machining *configuration space* (C-space) during tool-path generation. From the C-space, the feasible orientation region is constructed firstly. The optimal cutter orientation at each CC point is selected by searching the postures on the edge of the feasible region to achieve smooth posture change between neighboring points. This method is possible to achieve smooth cutter dynamics locally but not globally.

In our work on cutter selection, the A-maps at all the sampled points are part of the outputs. By using the A-maps, the algorithm for generating optimal iso-planar CL path has been developed in our previous work (Li, 2007). Firstly, the cutter dynamics is controlled by selecting the optimal cutting direction that results in the

minimum average *posture change rate* (PCR) of all the sample points. Secondly, the CL paths are generated one at a time, in which each CC point is generated by calculating the maximum step-forward length within the profile tolerance and the optimal cutter posture that can achieve a *maximum machining strip width* (MMSW) at each CC point. This two-step optimization arrangement aimed at serving two optimization objectives in a balanced manner: smooth cutting dynamics and high cutting efficiency. Based on these considerations, this method is named PCR-MMSW. With further implementation and testing, some drawbacks and limitations have been observed:

- (1) These two objectives may not align well. This “cutting direction for cutting dynamics” and “cutter posture for cutting efficiency” arrangement may cause the final result serve no objectives well.
- (2) The evaluation of cutting dynamics (CL-path smoothness) is done over the whole surface. In reality, it is the CL-path smoothness on a single path that matters.

Therefore, in this study, a different optimization method on iso-planar CL path generation has been proposed to address these problems. In the proposed method, on one hand, the maximum cutting efficiency is targeted with two measures: (1) selecting the cutting direction as the one that achieves the maximum machining strip width over the whole surface, and (2) at each CC point, choosing the cutter posture that results in maximum cutting efficiency. On the other hand, along each path, the cutter dynamics is controlled by making sure the cutter posture change between two consecutive CC points is within the specified tolerance. For easier reference, this method is called MMSW-PCR. The proposed method inherits the advantages of the previous method, such as the computational efficiency by using an interpolation method to obtain the CC points' cutter postures from the A-maps of its neighboring sampled points, and

machining efficiency by achieving a maximum machining strip width at the CC points. Furthermore, it is possible to achieve a smoother cutter dynamics along each CL-path.

4.2 Overview of the Proposed Optimal CL-Path Generation Method

The proposed iso-planar CL-path optimization method can be defined as “Given a surface region, a fillet-end cutter, the A-maps at all the sampled points (with high density), machining profile tolerance, and the posture change rate tolerance, choose the optimal cutting direction that achieves maximum machining strip width, and then generate the CC points together with the optimal cutter postures on each cutting path that results in smooth cutter dynamics”. Figure 4.2 shows the schematic view of the proposed approach, which includes three main steps.

Firstly, for a given cutter and a sampled point on the surface region, the maximum machining strip width along each sampled cutting direction is estimated. The optimal cutting direction is chosen as the sampled direction that achieves the maximum machining strip width among all the sampled points on the surface region.

Secondly, along the selected optimal cutting direction, the first cutting plane is set at slightly off the surface edge with a small distance Δy_0 (e.g., $\Delta y_0 = 0.1R$, where R is the cutter major radius) in the normal direction of cutting planes. Along this path, the first CC point is generated at the boundary point on the cutting plane and considered the *current CC point* on the *current path*. The optimal cutter posture at this point is obtained from its A-map based on a heuristic for maximum machining strip width. From the first CC point, the *initial position* of the next CC point is generated based on the maximum allowable step-forward length. The cutter posture for this CC point is then obtained using the heuristic-based algorithm described in Section 4.3. If the posture change from the current CC point to the next CC point is within a pre-defined tolerance, the position of the next CC point is said finalized. Otherwise, the

next CC point is re-positioned to be closer to the current CC point and the posture change is checked again. This process continues until the CC point reaches the boundary point of the current path and the current CL-path is completed.

Thirdly, with the current CL-path, the step-over widths for all the CLs on the path are quickly estimated based on the scallop-height tolerance. The smallest step-over width is set as the step-over width to locate the next cutting plane. The CLs for the next path are then generated one by one based on the aforementioned procedure. This process is repeated until the whole surface region is covered. The corresponding CL data in global frame can be easily generated according machine configuration.

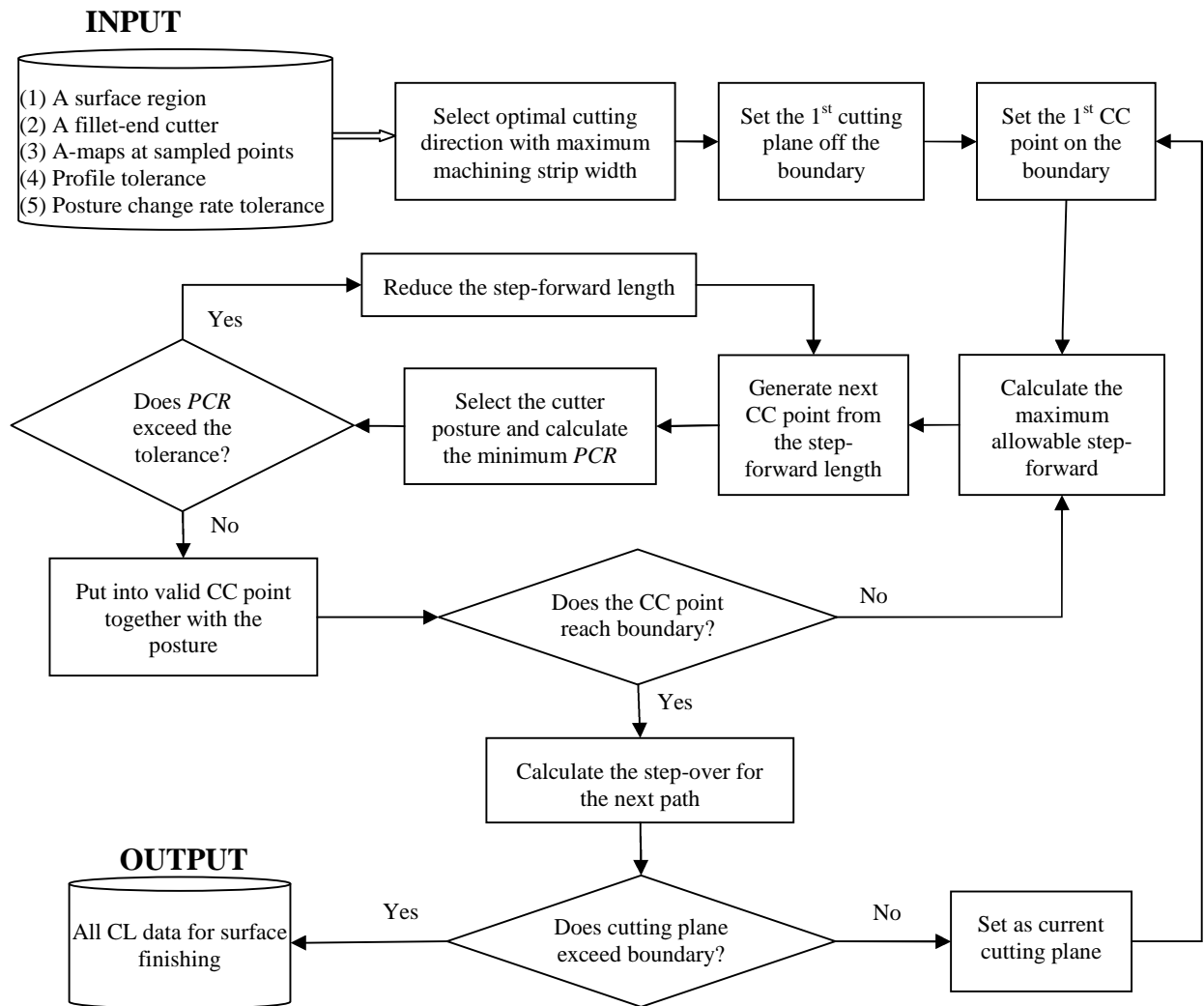


Figure 4.2 Flowchart of the optimal CL-path generation method

It is worth noting that the following 3 sub-algorithms developed in our previous work (Li, 2007) are re-used here: (1) maximum allowable step-forward length estimation, (2) scallop-height calculation based on the current and the next CL-paths, and (3) step-over estimation based on the current CL-path only. In the following sections, the following important algorithms developed in this study are to be described in details: (1) optimal cutter posture selection along a cutting direction for maximum cutting efficiency, (2) optimal cutting direction selection for maximum cutting efficiency, and (3) generation of CL path with smooth posture change in a single path.

4.3 Optimal Cutter Posture Selection along a Cutting Direction

In this section, the heuristic for the selection of optimal cutter posture aiming at maximum machining efficiency is described. The heuristic-based algorithm is developed for the following two different scenarios: (1) a CC point with available A-map, and (2) a CC point with unknown A-map, but the A-maps of its neighboring points are known.

4.3.1 Optimal cutter posture selection from the A-map

According to the machining strip width analysis in Section 3.6.1, the maximum machining strip width W at a point \mathbf{P}_{cc} can be achieved when angle θ has the same value as angle α , i.e., the cutter is oriented to face the feeding direction. Besides, W decreases with respect to the increment of angle λ . Thus, the preferable value of λ is the smallest available one at θ . Therefore, to achieve maximum cutting efficiency, the cutter's posture at \mathbf{P}_{cc} along α should be determined based on the following heuristic:

- (1) θ is chosen as the same to the cutting direction α

(2) λ is chosen from the lower-bound of the feasible range at θ in the A-map.

This heuristic is, however, not always applicable as the A-map at the current CC point may not cover the θ that is the same as the cutting direction α . If this is the case, θ should be chosen to be closed to α as much as possible. However, the choice of θ may not be unique as θ can approach α from both clockwise and counter-clockwise directions. In such cases, the secondary objective in cutter posture selection, i.e., smooth cutter dynamics, is taken into consideration. Here, the cutting dynamics is measured by the posture change rate (PCR) between two neighboring CLs. For example, between CC points \mathbf{P}_{cc} and \mathbf{P}_i , the posture change rate $\text{PCR}_{cc,i}$ is defined as:

$$\text{PCR}_{cc,i} = \frac{|\mathbf{T}_{cc} \mathbf{T}_i|}{|\mathbf{P}_{cc} \mathbf{P}_i|} \quad (4.1)$$

where $|\mathbf{P}_{cc} \mathbf{P}_i|$ is the Euclidean distance between \mathbf{P}_{cc} and \mathbf{P}_i . \mathbf{T}_{cc} and \mathbf{T}_i are the unit vectors in the global frame along the cutter axis directions at \mathbf{P}_{cc} and \mathbf{P}_i , respectively. At \mathbf{P}_{cc} , \mathbf{T}_{cc} with the posture (θ, λ) is given as:

$$\mathbf{T}_{cc} = (\sin \lambda \cos \theta) \mathbf{X}_L + (\sin \lambda \sin \theta) \mathbf{Y}_L + (\cos \lambda) \mathbf{Z}_L \quad (4.2)$$

where \mathbf{X}_L , \mathbf{Y}_L and \mathbf{Z}_L are the unit local frame axis vectors represented in the global frame.

Obviously, a small PCR corresponds to a smooth local CL change. To achieve a small PCR, the cutter posture at the previous CC point \mathbf{P}_{prev} , $(\theta, \lambda)_{\text{prev}}$, should be considered when specifying the current CC point's posture. A selection heuristic is proposed here, i.e., the one with smaller PCR with $(\theta, \lambda)_{\text{prev}}$ is chosen. Therefore, the complete heuristic for selecting the optimal posture at a CC point is given as follows:

- (1) If there exists an accessible λ range when $\theta=\alpha$, choose angle λ from the lower-bound of the feasible range at θ in the A-map. Otherwise, go to (2).

- (2) Select θ_1 and θ_2 from the A-map of the current CC point which are the closest ones to α from clockwise and counter-clockwise direction, respectively. If $|\alpha - \theta_1| \neq |\alpha - \theta_2|$, select $\theta = \{ \theta_k \mid \theta_k = \min\{\theta_1, \theta_2\} \}$. Choose angle λ from the lower-bound of the feasible range at θ in the A-map. Otherwise, go to (3).
- (3) Choose the optimal posture that achieves minimum posture change rate by:
 - (a) Choose angle λ_1 and λ_2 from the lower-bound of the θ_1 and θ_2 in the A-map, respectively.
 - (b) Calculate the \mathbf{T}_1 with (θ_1, λ_1) and \mathbf{T}_2 with (θ_2, λ_2) .
 - (c) Calculate the \mathbf{T}_{prev} with $(\theta, \lambda)_{\text{prev}}$ at \mathbf{P}_{prev} .
 - (d) Calculate $\text{PCR}_{1, \text{prev}}$ (between \mathbf{T}_1 and \mathbf{T}_{prev}) and $\text{PCR}_{2, \text{prev}}$ (between \mathbf{T}_2 and \mathbf{T}_{prev}). The posture with a smaller PCR is chosen as the optimal posture.

4.3.2 Optimal cutter posture selection through an interpolation approach

The heuristic-based algorithm introduced in the last section assumes the availability of A-maps at all the CC points. At this stage, the A-maps of the most of the CC points are not available, unless the CC point coincides with a sampled point. Although the A-map evaluation algorithm can be run but it is often time consuming. At the same time, it is also noticed that the density of the sampled points is generally much higher than that of the CC points. Therefore, to reduce the computation load, an interpolation approach is proposed to obtain the cutter's posture at a CC point without using its A-maps.

Firstly, the neighboring points $\mathbf{P}_i, i = 1, 2, \dots, p$, of CC point \mathbf{P}_{cc} are extracted from the sampled points that are the nearest points around \mathbf{P}_{cc} in (u, v) parameter domain. Based on experience, $p = 15$ tends to produce a reasonable result most of the time. A further filtering process of these neighboring points is done by ruling out

those ones with great change of the surface normal direction from \mathbf{P}_{cc} . The remaining sampled points $\mathbf{P}_i, i = 1, 2, \dots, m$, are called the candidate points.

Along a given cutting direction, the optimal cutter posture at $\mathbf{P}_i, i = 1, \dots, m$, \mathbf{T}_i , can be obtained based on the heuristic in the last section. The cutter axis direction at the current CC point, \mathbf{T}_{cc} , is thus obtained by:

$$\mathbf{T}_{cc} = \sum_{i=1}^m \frac{1}{|\mathbf{P}_{cc} \mathbf{P}_i|^2} \mathbf{T}_i \bigg/ \left| \sum_{i=1}^m \frac{1}{|\mathbf{P}_{cc} \mathbf{P}_i|^2} \mathbf{T}_i \right| \quad (4.3)$$

4.4 Optimal Cutting Direction Selection

In this section, the algorithm for optimal cutting direction selection for isoplanar tool-path is introduced with the objective set as the maximum cutting efficiency. This problem can be defined as “*given a part surface region, a fillet-end cutter, its A-maps at the sampled points, and the surface error tolerance, find a cutting direction along which the machining can achieve the maximum efficiency*”.

To achieve maximum cutting efficiency, the machining strip width, W , should be kept as large as possible when the surface error tolerance is satisfied. According to the geometric characteristics of the cutter and part surface, Yoon *et al.* (2003) proposed that at a point, the machining strip width (W) can be evaluated as:

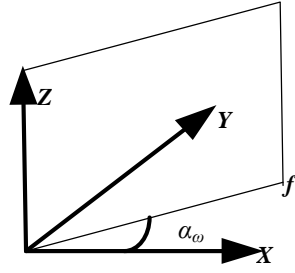
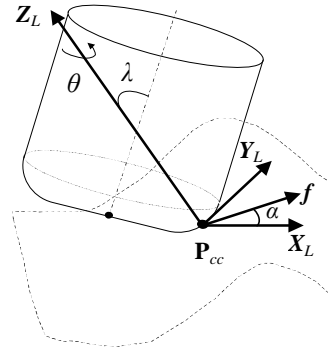
$$W = \sqrt{\frac{8h(a \cos^2 \alpha + 2b \sin \alpha \cos \alpha + c \sin^2 \alpha)}{-b^2 + ac}} \quad (4.4)$$

where a , b and c are expressions related to cutter orientation and surface geometry, which are given as:

$$\begin{aligned} a &= k_{t \max} \cos^2 \theta + k_{t \min} \sin^2 \theta - k_{s \max} \\ b &= (k_{t \max} - k_{t \min}) \sin \theta \cos \theta \\ c &= k_{t \max} \sin^2 \theta + k_{t \min} \cos^2 \theta - k_{s \min} \end{aligned} \quad (4.5)$$

Eqs. (4.4) and (4.5) indicate that at \mathbf{P}_{cc} , for known surface and cutter geometries, W is a function of α , θ , and λ . That is, given a cutting direction α_i at \mathbf{P}_{cc} , for any feasible cutter posture (θ_j, λ_j) , the corresponding $W_{i,j}$ can be evaluated. Thus, at \mathbf{P}_{cc} , we can obtain the $max-W_i$ (which represents the maximum machining strip width along α_i) by calculating the machining strip widths for each posture (θ_j, λ_j) based on the A-map. The maximum machining strip width $MAX-W$ at \mathbf{P}_{cc} can then be obtained by comparing $max-W_i$ along all the sampled cutting directions and the corresponding posture is the optimal one. This method is quite straightforward. However, as a large range of accessible postures may exist at each sampled point, it is very time consuming to evaluate all the sampled postures along each cutting direction. Therefore, instead of using all the postures along each cutting direction, some approximations are used to select a suitable cutter posture at a point that tends to produce maximum machining stripe width. The conclusion in the discussions on the relationship between machining strip width and the three key variables in Section 4.3.1 can be used here. That is, to calculate the $max-W_i$ at \mathbf{P}_{cc} for a given α_i , the cutter posture at \mathbf{P}_{cc} can be set by choosing θ to be closed to α_i as much as possible and λ from the lower-bound of the feasible range at θ in the A-map.

Let α_w denotes an angle of the plane from the $X-Z$ plane in the counter clockwise direction, as shown in Figure 4.3a. From the definition we know that α_w is the cutting direction angle in the global frame. The value of α_w cannot be directly utilized for specifying machining strip width at a point, since the algorithm is conducted in the local frame. Therefore, before starting, the conversion is needed between the global frame and the local frame. In the local frame, cutting direction is denoted as angle α , as shown in Figure 4.3b.

(a) Feeding angle (α_w) in global frame(b) Feeding angle (α) in local frame at \mathbf{P}_{cc} Figure 4.3 Feeding direction at \mathbf{P}_{cc} in global frame and local frame

The feeding direction α_w can be converted to α in the local frame through:

$$\begin{aligned}\cos \alpha &= \frac{([- \sin \alpha_w, \cos \alpha_w, 0]^T \times Z_L)}{|[- \sin \alpha_w, \cos \alpha_w, 0]^T \times Z_L|} \bullet X_L \\ \sin \alpha &= \frac{([- \sin \alpha_w, \cos \alpha_w, 0]^T \times Z_L)}{|[- \sin \alpha_w, \cos \alpha_w, 0]^T \times Z_L|} \bullet Y_L\end{aligned}\quad (4.6)$$

where X_L and Z_L are the unit axis vectors of the local frame represented in the global frame, respectively.

The general procedure of optimal cutting direction selection is thus developed. Firstly, at each sampled point \mathbf{P}_k , the global feeding direction range is uniformly sampled into a number of discrete directions $\{\alpha_{wi}\}$. Secondly, α_{wi} is converted to α_i in the local frame. Given the A-map information, the suitable cutter posture $(\theta_{ik}, \lambda_{ik})$ at \mathbf{P}_k can be specified along α_i , where θ_{ik} is the closest angle to α_i in the A-map and λ_{ik} is at the lower bound of A-map at θ_{ik} . The maximum machining strip width W_{ik} is evaluated by using Eq. (4.4). This process is repeated at all the other sampled points in the surface region. After that, along each sampled cutting direction α_{wi} , the average machining strip width $Ave(max-W_i)$ of all sampled points is obtained. A rating of the cutting directions is then conducted based on $Ave(max-W_i)$. The algorithm of finding the optimal cutting direction with maximum machining strip width is given as follows.

Algorithm: Searching for the optimal cutting direction

Input: (a) A set of sampled points $\{P_j\}$ ($j = 0, 1, \dots, n$) representing a NURBS surface region

(b) A fillet-end cutter (R, r_f, L) and its A-map at each sampled point

(c) Path direction angle range $[\alpha_{\omega min}, \alpha_{\omega max}]$

Output: Optimal cutting direction α_{ω}

BEGIN

- (1) Set $j = 0$.
- (2) Pick a point P_j from $\{P_j\}$.
- (3) Uniformly sample $[\alpha_{\omega min}, \alpha_{\omega max}]$ into $(m+1)$ angles, set $i = 0$.
- (4) Set $\alpha_{\omega i} = \alpha_{\omega min} + (\alpha_{\omega max} - \alpha_{\omega min})(i/m)$. Transform $\alpha_{\omega i}$ into α_i in the local frame.
- (5) Obtain (θ_i, λ_i) by choosing θ_i from the A-map that is closest to the α_i , and λ_i on the lower bound of A-map at θ_i .
- (6) For the given α_i and (θ_i, λ_i) , calculate the machining strip width W_{ij} . Save the data pair: $(\alpha_{\omega i}, W_{ij})$. Set $i = i + 1$. If $i \leq m$, go to (4).
- (7) Set $j = j + 1$. If $j \leq n$, go to (2).
- (8) Uniformly sample $[\alpha_{\omega min}, \alpha_{\omega max}]$ into $(m+1)$ angles, set $i = 0$.
- (9) Set $\alpha_{\omega i} = \alpha_{\omega min} + (\alpha_{\omega max} - \alpha_{\omega min})(i/m)$.
- (10) Calculate the average of W_{ij} over all the points in $\{P_j\}$ and record as $Ave(max-W_i)$. Save the data pair $(\alpha_{\omega i}, Ave(max-W_i))$. Set $i = i + 1$. If $i \leq m$, repeat from (9).
- (11) Set $\alpha_{\omega i}$ which achieves the maximum value of $Ave(max-W_i)$ the optimal feeding direction α_{ω} . Stop.

END

4.5 CL Data Generation with Smooth Posture Change on a Path

After obtaining the optimal cutting direction, the next step is to generate the CL data, which includes the generation of the paths, the CC points on each path, and selection of the cutter posture for each CC point. For iso-planar cutting, a path is produced by intersecting the surface region with the current cutting plane. On each path, the first CC point is obtained by intersecting the cutting plane and the surface region boundary. The subsequent CC points on the current path are generated mainly based on the specified profile tolerance. At the same time, it is also important to keep the posture change rate (PCR) along the path under a pre-defined bond. Therefore, starting from the first CC point, this process can be described as “finding the next CC point on the same path that satisfies: (1) the deviation of the tool motion trajectory is within the tolerance τ from the part surface, (2) the posture is able to achieve large machining strip width, and (3) the PCR from the previous CC point is within the predefined value τ_{PCR} ”.

The flowchart of generating CL data along a single path is given in Figure 4.2. From the first CC point, the *initial position* of the next CC point on the current path is generated by calculating the maximum allowable step-forward length within the profile tolerance. The optimal posture at this point is obtained from its A-map based on the same heuristic for maximum machining strip width. The posture change from the current CC point to the next CC point is then calculated. If this change is within a pre-defined tolerance, the position of the next CC point is said finalized, and this CC point becomes the current CC point. Otherwise, the next CC point is re-positioned to be closer to the current CC point and the posture change is checked again. This “CC point generation” and “posture change checking” process continues until the boundary point of the current path is reached and the current CL-path is said finalized.

4.5.1 Calculation of the maximum allowable step-forward length

As shown in Figure 4.4, at a CC point $\mathbf{P}_i(u_i, v_i)$, the next CC point on the same path $\mathbf{P}_{i+1}(u_{i+1}, v_{i+1})$ can be determined such that the largest deviation d from the line segment $\mathbf{P}_i\mathbf{P}_{i+1}$ to the part surface is very close to but smaller than the profile tolerance τ . Note here the cutting plane here is assumed to be normal to \mathbf{Y} axis as $y = y_i$.

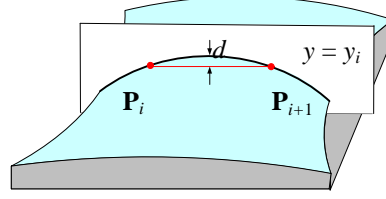
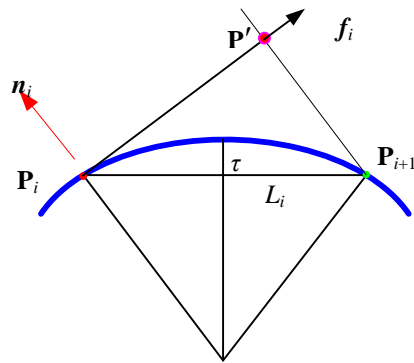


Figure 4.4 A single iso-planar path

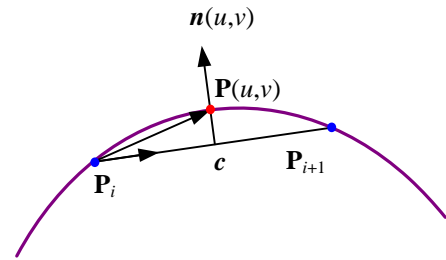
By approximating the surface curve at \mathbf{P}_i to a circular curve, as shown in Figure 4.5a, the step-forward length between $\mathbf{P}_i\mathbf{P}_{i+1}$, L_i , can be estimated as:

$$L_i = \sqrt{8R\tau - 4\tau^2} = \sqrt{8\tau / \kappa - 4\tau^2} = 2\sqrt{\frac{\tau(2 - \kappa\tau)}{\kappa}} \quad (4.7)$$

Where κ is the curvature of the path curve at \mathbf{P}_i , which can be obtained using Meusnier theory (Guggenheimer, 1977).



(a) Step-forward calculation



(b) Chord length deviation from the path curve

Figure 4.5 Calculation of step-forward length for an iso-planar path

With L_i , the method by Hwang (1992) is adopted here to search for \mathbf{P}_{i+1} on the path with the distance L_i from \mathbf{P}_i along the cutting direction. As shown in Figure 4.5a,

let \mathbf{P}' be the intersection point between the line starting with \mathbf{P}_i along the tangent direction \mathbf{f}_i and the line starting with \mathbf{P}_{i+1} along \mathbf{n}_i direction. \mathbf{P}' and \mathbf{P}_{i+1} can be expressed by:

$$\mathbf{P}' = \mathbf{P}_i + ((R - \tau)L_i\kappa)\mathbf{f}_i \quad (4.8)$$

$$\mathbf{P}_{i+1} = \mathbf{P}' + l\mathbf{n}_i \quad (4.9)$$

Where l is the distance of \mathbf{P}_{i+1} from \mathbf{P}' . To obtain the step-forward length, the maximum deviation between line segment $\mathbf{P}_i \mathbf{P}_{i+1}$ and the path curve need to be evaluated. Let point $\mathbf{P}(u, v)$ (see Figure 4.5b) represent the point on the path curve at which the deviation reaches its maximum. From geometry analysis, the normal $\mathbf{n}(u, v)$ at \mathbf{P} is perpendicular to $\mathbf{P}_i \mathbf{P}_{i+1}$. The deviation distance d between $\mathbf{P}_i \mathbf{P}_{i+1}$ and path curve is represented as:

$$d = \left| (\mathbf{P} - \mathbf{P}_i) - \left[(\mathbf{P} - \mathbf{P}_i) \bullet \frac{\mathbf{P}_{i+1} - \mathbf{P}_i}{\|\mathbf{P}_{i+1} - \mathbf{P}_i\|} \right] \frac{\mathbf{P}_{i+1} - \mathbf{P}_i}{\|\mathbf{P}_{i+1} - \mathbf{P}_i\|} \right| \quad (4.10)$$

Based on the above discussion, the numerical method to search for the initial position of the next CC points \mathbf{P}_{i+1} on a single path includes three steps:

- (1) Set the initial value of step-forward length between $\mathbf{P}_i \mathbf{P}_{i+1}$ regarding to local surface geometry at \mathbf{P}_i ;
- (2) Search for an estimated point \mathbf{P}_{i+1} on the path with the distance as the step-forward length from \mathbf{P}_i ;
- (3) Check whether the deviation between the tool trajectory and the path is within the neighboring range, i.e., $(1-\delta)\tau \leq d \leq \tau$, where δ is the pre-defined small value (e.g., 0.05). If the condition is not satisfied, the step-forward length is accordingly changed (increased or reduced), and steps (2) and (3) are repeated till the suitable \mathbf{P}_{i+1} is obtained.

The resulted L_i is called the maximum allowable step-forward length L_{imax} at CC point \mathbf{P}_i . \mathbf{P}_{i+1} is the initial position of the next CC point. The combination $(\mathbf{P}_i, L_{imax}, \mathbf{P}_{i+1})$ is recorded to be used in the next step.

4.5.2 Generate the CL data at the next CC point

From the combination $(\mathbf{P}_i, L_{imax}, \mathbf{P}_{i+1})$, the next step is to finalize the position of the next CC point starting from its initial position \mathbf{P}_{i+1} . This is achieved by checking to see whether the PCR between \mathbf{P}_i and \mathbf{P}_{i+1} within a predefined value τ_{PCR} . If this constraint is not satisfied, the distance between \mathbf{P}_i and \mathbf{P}_{i+1} is shortened and a new position of \mathbf{P}_{i+1} obtained. This process continues until the constraint is satisfied and \mathbf{P}_{i+1} is the next CC point. The details are given in the following sections.

Let PCR_i represent the PCR between the two CLs at \mathbf{P}_i and \mathbf{P}_{i+1} , and \mathbf{T}_i and \mathbf{T}_{i+1} be the unit vectors in global frame along the cutter axis directions at \mathbf{P}_i and \mathbf{P}_{i+1} , respectively. To calculate PCR_i using Eq. (4.1), \mathbf{T}_i and \mathbf{T}_{i+1} need to be obtained first. As discussed in Section 4.3.2, along a cutting direction, the cutter posture at a point of interest can be determined from the interpolation of its neighboring points' optimal cutter orientations. In this study, it is assumed that cutting planes are perpendicular to XY -plane of global frame. Thus, a path direction angle α_ω is an angle of the cutting plane from the XZ -plane in counter clockwise direction in the global frame. To use the method introduced in Section 4.3.2 for determining cutter postures at \mathbf{P}_i and \mathbf{P}_{i+1} , α_ω has to be converted to angle α_i and α_j in the respective local frame as:

$$\begin{aligned}\cos \alpha_k &= \frac{([- \sin \alpha_\omega, \cos \alpha_\omega, 0]^T \times \mathbf{Z}_{Lk}) \cdot \mathbf{X}_{Lk}}{|[- \sin \alpha_\omega, \cos \alpha_\omega, 0]^T \times \mathbf{Z}_{Lk}|} \\ \sin \alpha_k &= \frac{([- \sin \alpha_\omega, \cos \alpha_\omega, 0]^T \times \mathbf{Z}_{Lj}) \cdot \mathbf{Y}_{Lk}}{|[- \sin \alpha_\omega, \cos \alpha_\omega, 0]^T \times \mathbf{Z}_{Lk}|}\end{aligned} \quad (k=i, i+1) \quad (4.11)$$

where $[-\sin \alpha_o, \cos \alpha_o, 0]^T$ is the normal vector of the cutting plane in the global frame. \mathbf{X}_{Lk} , \mathbf{Y}_{Lk} and \mathbf{Z}_{Lk} denote the unit axis vectors of the local frame at point \mathbf{P}_k represented in the global frame, respectively.

Using Eq. (4.11), path direction α_o is converted to α_i in the local frame at \mathbf{P}_i and angle α_{i+1} at \mathbf{P}_{i+1} . Regarding to angles α_i and α_{i+1} , both the cutter orientation (λ_i, θ_i) at \mathbf{P}_i and $(\lambda_{i+1}, \theta_{i+1})$ at \mathbf{P}_{i+1} can be determined from their A-maps, using the algorithm introduced in Section 4.3.2. By using the Eq. (4.2), \mathbf{T}_i and \mathbf{T}_{i+1} can be obtained by transforming (λ_i, θ_i) and $(\lambda_{i+1}, \theta_{i+1})$ to the global frame. PCR_i can then be obtained by using the Eq. (4.1).

The obtained PCR_i is then compared with τ_{PCR} . If $\text{PCR}_i < \tau_{PCR}$, \mathbf{P}_{i+1} is taken as the final position of the next CC point and the corresponding posture is also taken to complete the CL data. Otherwise, \mathbf{P}_{i+1} is changed to a new position closer to \mathbf{P}_i . This is done by reducing the distance between \mathbf{P}_i and \mathbf{P}_{i+1} by a small proportion (e.g., 5%) from the current step-forward length and regenerate \mathbf{P}_{i+1} from \mathbf{P}_i . This iterative process continues until $\text{PCR}_i < \tau_{PCR}$ is satisfied. In this process, the principle of “maximum machining strip width” is followed when choosing the posture for the new CC point, and the objective of “smooth cutter dynamics” is achieved by changing the position (as well as posture) of the new CC point. It is worth to mention that there is a possibility that there would be no CC point could satisfy $\text{PCR}_i < \tau_{PCR}$, especially on some regions with sharp curvature changes. To adapt to this case, the maximum times of the iterations N_{max} (e.g., $N_{max} = 20$) is used. That is, if the time of iteration reaches the specified maximum value, a comparison between all the PCRs evaluated during the iterations is conducted and the one with the smallest PCR is set as the final valid next CC point. The overall algorithm to generate the CL data on a single path is described as follows:

Algorithm: Searching for the CC points and their postures on a single tool-path

Input: (a) *A set of sampled points $\{P_j\}$ ($j = 0, 1, \dots, n$) representing a NURBS surface region*

(b) *The path cutting plane $y = y_i$*

(c) *The profile tolerance τ*

(d) *The posture change rate tolerance τ_{PCR}*

Output: *A set of CC points and their postures on the path*

BEGIN

- (1) Search for the boundary points of the surface curve on plane $y = y_i$.
- (2) Set one boundary point as the current CC point \mathbf{P}_i . Transfer the cutting direction in the local frame α_i , then optimal posture (λ_i, θ_i) can be identified using the quick interpolation approach presented in 4.3.2. The cutter axis in the global frame \mathbf{T}_i is calculated.
- (3) Estimate the maximum step-forward length L_i using Eq. (4.7). Find the L_{imax} and the initial position of the next CC point \mathbf{P}_{i+1} iteratively using the procedures introduced in Section 4.5.1. Set $k = 0$.
- (4) Calculate PCR_i .

IF $PCR_i < \tau_{PCR}$

\mathbf{P}_{i+1} and its posture form the next CL.

ELSE IF $k < N_{max}$ (where N_{max} is a pre-set maximum number of iterations, e.g.

20)

Record the pair $\{\mathbf{P}_{i+1}, PCR_i\}_k$. Set $L_i = 0.95 L_i$. Find the new position of the next CC point \mathbf{P}_{i+1} by following the procedure in Eqs. (4.8) and (4.9). Go to (4).

ELSE

Set $\mathbf{P}_{i+1,k}$ as the valid next CC point \mathbf{P}_{i+1} where k is the one achieves the minimum value of $\text{PCR}_{i,k}$, e.g., $\mathbf{P}_{i+1} = \{ \mathbf{P}_{i+1,k} \mid \text{PCR}_{i,k} = \min\{\text{PCR}_{i+1}, \text{PCR}_i\}_k \}$.

END IF

- (5) Save \mathbf{P}_{i+1} together with $(\lambda_{i+1}, \theta_{i+1})$ and \mathbf{T}_{i+1}
- (6) If \mathbf{P}_{i+1} does not reach the curve segment boundary, set it as the current CC point and go back to (3).
- (7) Output all CC points and their postures on the path. Stop.

END

4.6 Step-Over Calculation

After obtaining all CL data on the current path, the next step is to calculate the step-over for setting the position of the next cutting plane. The process of step-over calculation at a CC point is similar to our previous method (Li, 2007). For clarity, a relatively brief description is given here.

The general idea of calculating step-over is to maximize the path interval such that the resultant scallop-height is just within the given tolerance h . To maximum path interval Δy_j at CC point \mathbf{P}_j on the current path, the machining strip width at \mathbf{P}_j must be maximized. As shown in Figure 4.6a, when a fillet-end cutter moves through a CC point \mathbf{P}_{cc} with posture (λ, θ) along the feeding direction f , the cutter's *effective cutting edge* is a circular curve through \mathbf{P}_{cc} on the cutter's filleted portion and normal to the cutter axis. The cutter swept curve on the normal plane is the projection curve of the effective cutting edge on that plane. The machined surface S_M is bounded within the offset surface, a surface with a distance of h from the design surface S_D . Therefore, the machining strip width can be approximated by calculating the intersection between the cutter swept curve and the surface curve with an offset of h along the

feeding direction. As shown in Figure 4.6b, \mathbf{P}_a and \mathbf{P}_b are the intersection points of the effective cutting shape and the offset part surface S_h on $Y'_L-Z'_L$ plane. The machining strip width is referred as the distance of elliptical cross section \mathbf{P}_b - \mathbf{P}_{cc} - \mathbf{P}_a along the axis Y'_L (Sheltami *et al.*, 1998).

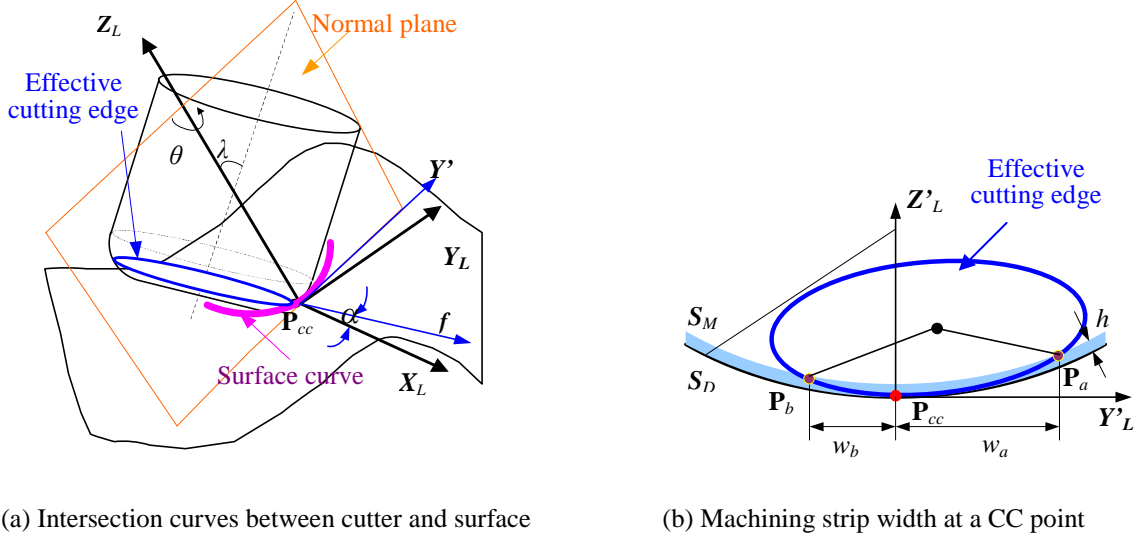


Figure 4.6 Evaluation of machining-strip width at point Pcc

To satisfy the accuracy requirement, the machining strips of the two adjacent paths have to be overlapping to satisfy the scallop-height tolerance h , as shown in Figure 4.7a. On the other hand, the overlapping percentage should be kept as low as possible to maximize machining efficiency. Thus, the path interval Δy_j at \mathbf{P}_j should be close to but no larger than the projection of the connected machining strips at \mathbf{P}_j and \mathbf{P}_{j+1} along Y -axis as:

$$\Delta y_j \leq \omega_{b,jy} + \omega_{a,j+1y} \leq (1 + \varepsilon) \Delta y_j \quad (4.12)$$

Where \mathbf{P}_{j+1} is the corresponding CC point on the next path; $\omega_{b,jy}$ and $\omega_{a,j+1y}$ are the projection of machining strip width at \mathbf{P}_j and \mathbf{P}_{j+1} on the Y -axis direction, respectively; and ε is pre-defined very small value, e.g., 0.05.

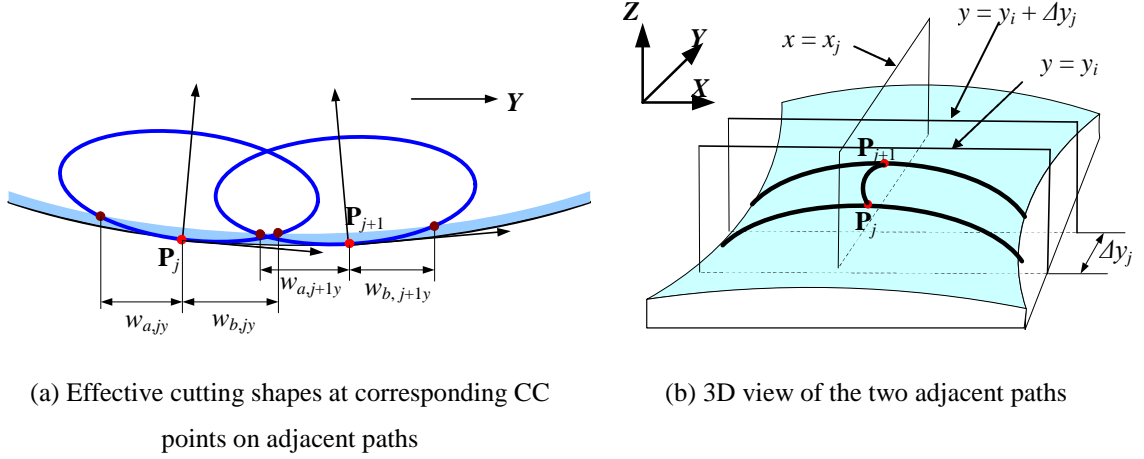


Figure 4.7 Calculation of path interval between two adjacent paths at a CC point \mathbf{P}_j

An iterative algorithm is designed to find \mathbf{P}_{j+1} with the largest allowable path interval. Firstly, from $\mathbf{P}_j(u_j, v_j)$, an estimated point $\mathbf{P}_{j+1}(u_{j+1}, v_{j+1})$ is found with an initial path interval $\Delta y_j = \omega_{b,jy} + \omega_{a,jy}$ as (see Figure 4.7b):

$$\begin{aligned} S_x(u_{j+1}, v_{j+1}) &= x_j \\ S_y(u_{j+1}, v_{j+1}) &= y_i + \Delta y_j \end{aligned} \quad (4.13)$$

Where (x_j, y_i, z_j) are the coordinates of \mathbf{P}_j in the machine frame. Secondly, the cutter posture is specified at \mathbf{P}_{j+1} with regard to its feeding direction using the algorithm in Section 4.3.2 and the machining strip width is then calculated. If condition (4.12) is not satisfied, the value of Δy_j is accordingly changed, and this process of finding \mathbf{P}_{j+1} from Eq. (4.13) and evaluating the machining strip width at \mathbf{P}_{j+1} is repeated until the maximum allowable Δy_j is obtained. The detailed algorithm is given as follows:

Algorithm: Calculating the path-interval at a CC point

- Input:** (a) A CC point \mathbf{P}_{cc} and the corresponding posture
 (b) A-maps at sampled surface points
 (c) Scallop height tolerance h

Output: The path interval at \mathbf{P}_{cc} : Δy_{cc}

BEGIN

- (1) Calculate the machining strip width w_a and w_b at \mathbf{P}_{cc} , and convert them as w_{ay} and w_{by} along \mathbf{Y} -axis. Set $\Delta y_{cc} = w_{ay} + w_{by}$.
 - (2) Determine the adjacent estimated point \mathbf{P}_{cc1} with Δy_{cc} , using Eq. (4.12).
 - (3) Determine the cutter posture at \mathbf{P}_{cc1} based on A-maps at sampled surface points, using the algorithm introduced in Section 4.3.2.
 - (4) Calculate the machining strip width w_{a1} and w_{b1} at \mathbf{P}_{cc1} , and convert them as w_{ay1} and w_{by1} along \mathbf{Y} -axis.
 - (5) Adaptively adjust the value of Δy_{cc} according to the value of $w_{by} + w_{ay1}$:

IF $\Delta y_{cc} \leq w_{by} + w_{ay1} \leq (1 + \varepsilon) \Delta y_{cc}$

go to (6).

ELSE IF $\Delta y_{cc} > w_{by} + w_{ay1}$

decrease Δy_{cc} with a small step and go to (2).

ELSE IF $w_{by} + w_{ay1} > (1 + \varepsilon) \Delta y_{cc}$

increase Δy_{cc} with a small step and go to (2).

END IF
 - (6) Output Δy_{cc} as the path interval at \mathbf{P}_{cc} . Stop.
- END**

Based on the step-over calculation at a CC point, the largest allowable path interval for a path, Δy , can be obtained by calculating the maximum allowable path interval Δy_j at each CC point on the current path and taking the minimum one, i.e., $\Delta y = \min\{\Delta y_j | j = 1, \dots, n_i\}$, where n_i is the number of CC points on the current cutting path. Δy is set as the step-over and the next CL path is then generated.

4.7 The Overall Algorithm for CL-Path Generation

Combined the above algorithms on the determination of cutter posture, the generation of CC points with a cutting plane, and the calculation of path interval, the overall algorithm for iso-planar CL-path generation is given as follows:

Algorithm: Generating CL-paths to finish the surface region

Input: (a) *A set of sampled points $\{P_j\}$ ($j = 0, 1, \dots, n$) representing a NURBS surface region*

(b) *A fillet-end cutter (R, r_f, L)*

(c) *A-maps at all sampled points*

(d) *Machining profile tolerance τ and scallop-height tolerance h*

(e) *Cutter posture change rate limit τ_{PCR}*

Output: *CL-paths with a set of CL data*

BEGIN

- (1) Select the optimal cutting direction using the method presented in Section 4.4. Record the normal vector for the path cutting planes as \mathbf{n} .
- (2) Set a cutting frame with axis \mathbf{Y} aligned with \mathbf{n} . Transform $\{P_j\}$ from the global frame to the cutting frame. And find the minimum y_{min} and maximum y_{max} for $S(u,v)$. Set $y_i = y_{min} + d$, where d is a small value (e.g., $0.1R$) between the first tool-path and the surface edge.
- (3) Plan the path on plane $y = y_i$ and put the CC points into the set $\{\mathbf{P}_{ij}, j = 1, \dots, n_i\}$, using the algorithm in Section 4.5.
- (4) Calculate the maximum path interval $\Delta y_{ij}, j = 1, \dots, n_i$, at each CC point on the current path using the algorithm in Section 4.6.
- (5) Set the path interval $\Delta y_i = \min\{\Delta y_{ij}, j = 1, \dots, n_i\}$.
- (6) Set $y_i = y_i + \Delta y_i$.

IF $y_i \leq y_{max}$

Generate the next CL path using the algorithm in Section 4.5. Go back to (4).

ELSE

Go to (7).

ENDIF

- (7) Convert the CL data from the cutting frame to the global frame and output the CL data.

END

4.8 A Comparison Case Study

The aforementioned method for optimal CL-path generation has been implemented using C++ and OpenGL. In this section, an example for optimal CL-path generation to machine (finishing cut) a sculptured surface is presented to demonstrate the effectiveness of the developed method. Furthermore, a comparison is conducted between the proposed method and our previous optimization method (Li, 2007). The example shows that the generated CL path can achieve both smooth cutter dynamics and high cutting efficiency.

Figure 4.8 illustrates the geometry of a chair-shape surface patch. Figure 4.8a shows the u , v distribution of the NURBS surface in the global frame X - Y - Z , while Figure 4.8b gives the surface decomposition, which consists of concave, convex, and saddle regions. It was sampled uniformly along u and v directions into 201×201 points. A fillet-end cutter $T(R, r_f, L) = (6\text{mm}, 0.5\text{mm}, 60\text{mm})$ was selected to mill this surface. The A-map of the cutter at every sampled point was then obtained by setting the out surface tolerance of 0.05mm for rear gouging calculation and stock surface offset 0.2mm for global collision calculation.

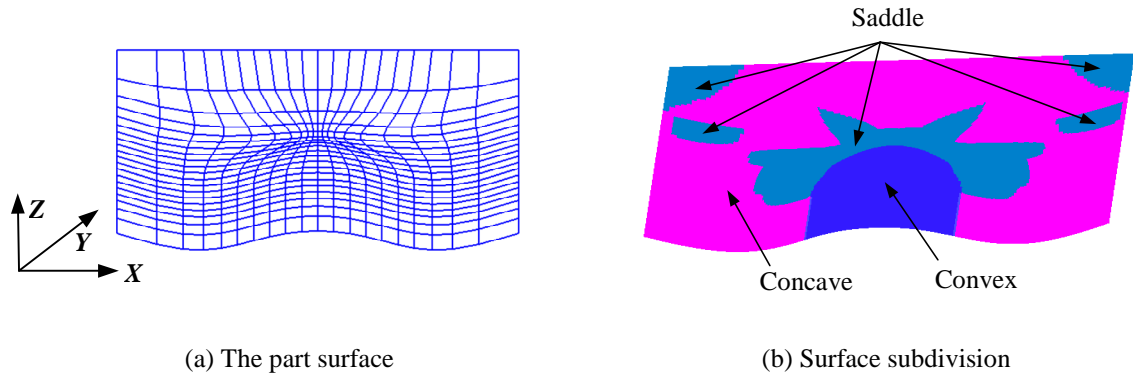


Figure 4.8 The part surface

During the CL path generation, both the profile tolerance and the scallop-height tolerance were specified as 0.1 mm. From X -axis of the global frame, the cutting direction within the range of $[0^\circ, 360^\circ]$ has been uniformly sampled into 73 discrete angles with an increment of 5° . Following the steps in the MMSW-PCR method, along each sampled cutting direction, the average machining strip width is calculated, as shown in Figure 4.9. The direction with angle 270° from X -axis has the maximum machining strip width and thus is taken as the optimal cutting direction. It is worth to mention the direction with angle 0° has the minimum machining strip width.

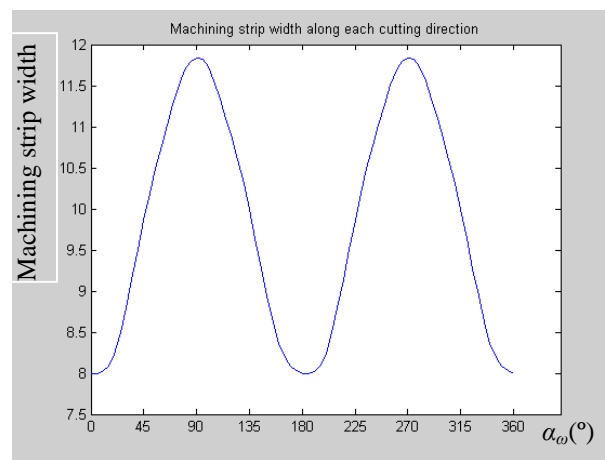


Figure 4.9 Average machining strip widths along all the cutting directions

Along the optimal cutting direction, the CLs are generated with posture change control such that $\tau_{PCR} = 0.1/\text{mm}$. The generated CL-paths are shown in Figure 4.10a (only positional information is shown). For comparison, the PCR-MMSW method developed earlier (Li, 2007), in which the cutting direction is selected based on minimum average PCR and CLs (CC points, postures, and path interval) based on MMSW was also used to generate the CL-path. The selected optimal cutting direction from PCR-MMSW method is 0° . The resulted CL-paths are shown in Figure 4.10b.

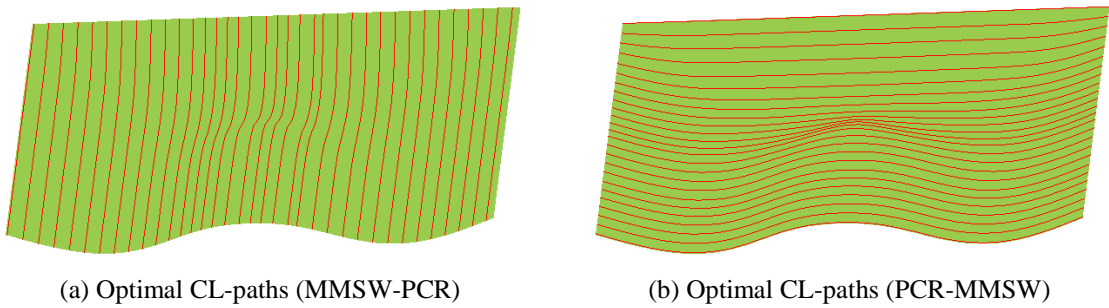


Figure 4.10 The CL-paths generation from the two methods

Table 4.1 shows the comparison results of these two generated CL-paths, under different criteria. It can be seen that the CL-paths from the MMSW-PCR have fewer CC points. In addition, it has a much shorter overall length than CL-paths from the PCR-MMSW. As for the average PCR over the whole CL-path, the MMSW-PCR CL-paths are also much smoother than the PCR-MMSW CL-paths. In summary, for this case study, the MMSW-PCR method outperforms the PCR-MMSW method. However, it is also noted that one drawback of the MMSW-PCR is that the CL-paths have more paths than the PCR-MMSW CL-paths (36 vs. 25), which may result in more air-travel time.

Table 4.1 The comparison between the MMSW-PCR and the PCR-MMSW CL-paths

	Optimal cutting direction	Number of CC points	Total path length (mm)	Average PCR (mm-1)
MMSW-PCR	270°	752	4296.35	0.00764
PCR-MMSW	0°	867	5155.28	0.01042

Furthermore, CL paths are generated by using MMSW-PCR method along all the sampled feeding directions. The path lengths and PCRs are shown in Figure 4.11a and b. It can be seen that the actual path length along 270° is quite close to the smallest length, while PCR along 270° is the smallest one. That is, the presented method is proved to be optimal in generating CL paths that are able to achieve an efficient machining with smooth cutter dynamics.

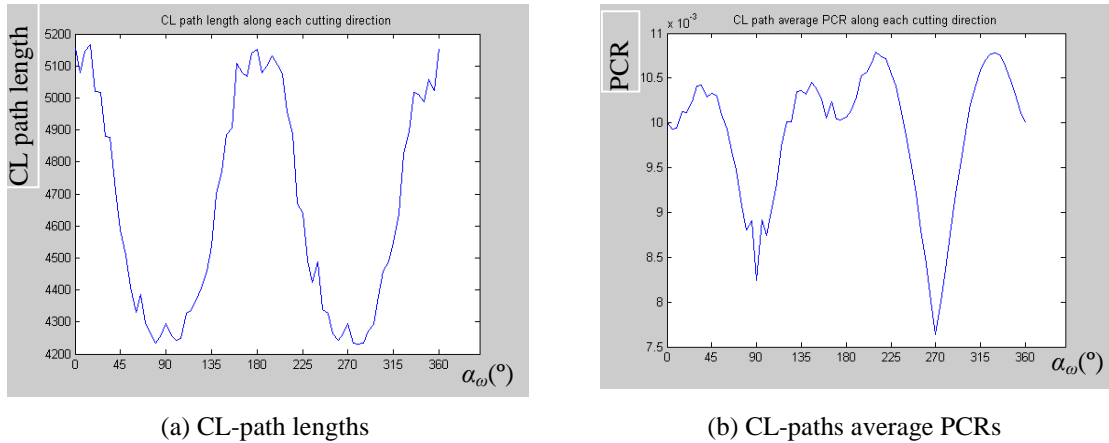


Figure 4.11 CL-paths comparison along all the cutting directions

4.9 Summary

This chapter presents a new approach to generate the 5-axis iso-planar CL-path for finishing a NURBS surface region using a single cutter. Based on the A-maps of the sampled points resulted from the cutter selection stage, optimization strategies such as maximum machining strip width and smooth cutting dynamics have been incorporated into the CL-path generation process.

Firstly, since the A-map construction is performed with respect to all possible directions instead of a fixed feeding direction, it is possible to select the optimal cutting direction based on a specified objective by trying all possible direction. Here, the optimal cutting direction is defined as the one resulting in maximum average machining strip width for all sampled points. Approximation methods have been developed to estimate the machining strip width at a point, which provides an effective tool for selecting a cutting direction that favors cutting efficiency.

Secondly, on each cutting plane along the cutting direction, along each path, CC points are generated in a way such that the posture change rate between two neighboring CC points is kept within a tolerance to smooth the cutting dynamics. In addition, the cutter posture for every CC point is selected by following the heuristic that favors cutting efficiency by maximizing the machining strip width. As such, the two objectives are served in a balanced manner.

The presented method can achieves the maximum machining strip width in a global way and smooth posture change rate in a local way. Compared to our previous work on iso-planar CL-path generation which considers the posture change rate over the whole surface, the proposed method is more reasonable as posture change should be considered locally, i.e., keeping the maximum change within a specified tolerance.

It is worthy to mention the advantage of using A-map information of the sampled points resulted from the cutter selection. Due to the high density of the sampled points, the cutter's posture can be obtained with the interpolation of neighboring points' A-maps. In this way, the overall computation load is reduced significantly. Moreover, the two important planning tasks for 5-axis machining: cutter selection and CL-path generation, are carried out in an integrated manner through the use of the common data source: the A-maps.

CHAPTER 5

MULTI-CUTTER MACHINING: CL PATH GENERATION, SYSTEM IMPLEMENTATION, AND TESTING

For a given surface and a set of cutters, multi-cutter CL path generation is to generate interference-free CL paths on each surface region with its assigned cutter that meets the accuracy requirement. The combinations of surface region and cutter (*eAR*/cutter) resulted from the optimal multi-cutter set selection (presented in Chapter 3) serves as an input. In addition, the method of iso-planar CL path optimization on a surface region given in Chapter 4 is employed to achieve the objectives of maximum cutting efficiency and smooth cutter dynamics within the surface tolerance requirement. Therefore, with the *eAR*/cutter information, by extending the iso-planar CL path optimization algorithm, this chapter presents the method on iso-planar CL paths generation for multi-cutter set machining. The proposed method includes cutting direction selection and CL data generation.

Furthermore, by implementing the aforementioned algorithms in a Visual C++ environment, the integrated process planning system for 5-axis machining has been developed. This chapter also gives an overall introduction of the implemented system. For better demonstration, the interfaces on A-map calculation, multi-cutter selection, and multi-cutter CL path generation are shown in a sequential view with an illustrative example.

5.1 Background

The use of multi-cutter machining is to achieve improved cutting efficiency compared to single cutter machining. However, similar to the current status of 5-axis multi-cutter selection, there is hardly any reported study on 5-axis multi-cutter tool path generation. Therefore, this research focuses on developing a systematic method on CL path generation for 5-axis multi-cutter machining.

The nature of multi-cutter machining is to use a set of cutters together to machine a whole surface. That is, the surface is decomposed into a set of individual regions, and each region is assigned with a suitable cutter. The CL paths are then planned on each region to achieve certain objectives. In Chapter 3, the methodology of optimal multi-cutter set selection for 5-axis sculptured surface finish-cut is described. The output not only includes the optimal cutter set, but also each *eAR* with its assigned cutter (*eAR*/cutter), i.e., $\{eAR_i/T_i, i = 1, 2, \dots, N\}$, where N is the total number of the *eAR*s. In addition, Chapter 4 presents an optimization method for single-cutter iso-planar CL paths generation for a given *eAR*/cutter (isolated). The method includes two phases: (1) selection of optimal cutting direction that achieves maximum machining strip width and (2) generation of CL data within the pre-defined profile tolerance while keeping the PCR along each CL path under control. For multi-cutter CL paths generation, a number of *eAR*/cutter sets (related) need to be planned, and this single-cutter CL path planning method can be partially employed in this planning process.

5.2 Iso-Planar CL Paths Generation in Multi-Cutter Machining

The goal of the multi-cutter CL path generation method is set as to automatically generate the tool paths that can achieve both maximum cutting

efficiency and smooth cutting dynamics. Following this principle, the multi-cutter iso-planar CL path generation problem can be defined as “given a set of *eAR*/cutter with each *eAR* represented by a unique sampled surface point set, the *A*-map for each sampled point in its *eAR*/cutter, the machining profile tolerance, and the maximum allowable PCR, generate the CL path for each *eAR*/cutter that can achieve maximum cutting efficiency and smooth cutting dynamics within the tolerance requirement”.

From the definition, the flowchart of the multi-cutter CL path generation method is shown in Figure 5.1. It consists of two steps. Firstly, the cutting direction for each *eAR*/cutter is selected aiming at the maximum cutting efficiency. Since all the *eAR*s are inter-related, it may not be a good idea to treat each *eAR*/cutter as an isolated case. Secondly, for each *eAR*, followed the selected cutting direction, the boundary of the cutting region is extracted and CL paths are planned to achieve smooth PCR within the surface finish requirement. This process is repeated until the CL paths are planned for all the *eAR*s.

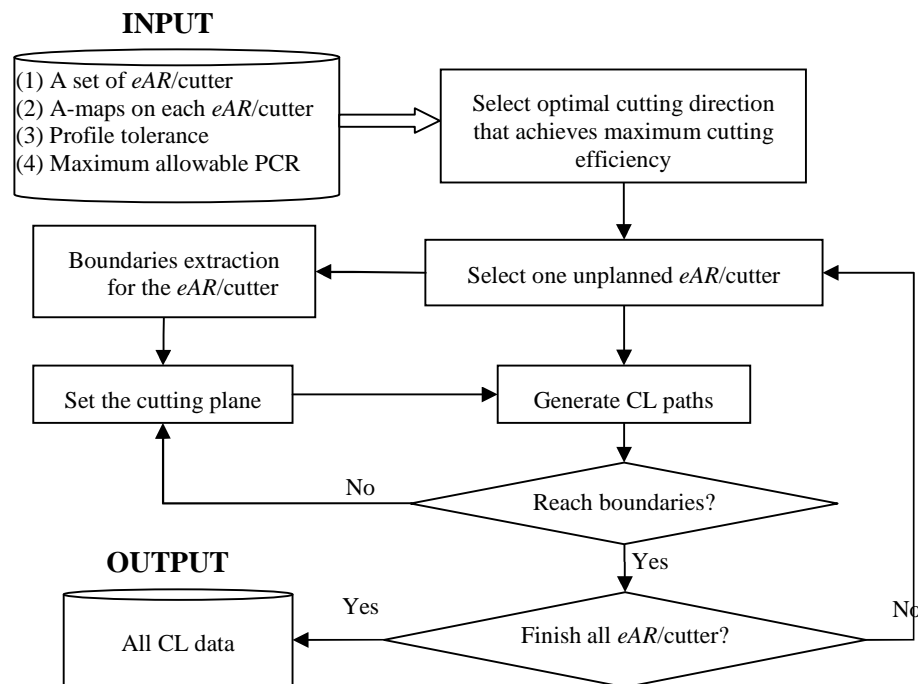


Figure 5.1 Flowchart of 5-axis multi-cutter iso-planar CL path generation

5.2.1 Optimal cutting direction selection

As presented in the last chapter, the single-cutter optimal cutting direction is selected in four steps: (1) uniformly sample the range of the cutting direction, (2) at each sampled surface point, calculate the machining strip width along all the sampled cutting directions, (3) along each sampled cutting direction, the *average machining strip width* is calculated by summing up that at each sampled surface point over the region, and (4) the optimal cutting direction is chosen as the one that achieves the maximum average machining width.

For multi-cutter tool-path generation, there are two approaches with regard to the ‘optimal’ cutting direction selection: (1) the *individual* approach – an optimal cutting direction for each *eAR*/cutter and (2) the *collective* approach – an optimal cutting direction for all the *eAR*/cutter, i.e., the whole surface. Figure 5.2 shows a surface represented by its *eAR*/cutter resulted from multi-cutter selection. Under the *individual* approach, each *eAR*/cutter is treated independently. The single-cutter CL path generation method can be used here for finding the optimal cutting direction for each *eAR*/cutter. This results in different optimal cutting directions for different *eAR*s (even using the same cutter), as shown in Figure 5.2a. Under the *collective* approach, on each *eAR*/cutter, the machining strip widths at every sampled point along all the sampled cutting directions are evaluated firstly. Along each cutting direction, the average machining strip width is then calculated among all the sampled points over the whole surface. The sampled direction with the maximum average machining strip width is chosen as the optimal cutting direction (see Figure 5.2b). In this way, the ranking of the cutting directions is conducted in the surface level, instead of each individual *eAR* level. Intuitively, the resulted cutting efficiency from the *individual* approach is better than that of the *collective* approach. However, with the *individual*

approach, a potential surface smoothness problem may arise in the areas between the boundaries of adjacent eAR s (due to different cutting directions). Therefore, the *collective* approach is taken here.

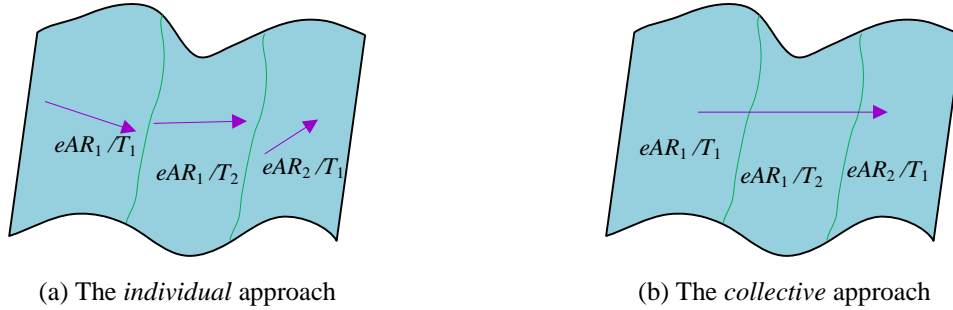


Figure 5.2 Cutting direction selection approaches

The overall algorithm on optimal cutting direction selection for multi-cutter machining is given as follows:

Algorithm: Finding the optimal cutting direction for multi-cutter machining

- Input:** (a) A set of sampled points $\{\mathbf{P}_j\}$ ($j = 0, 1, \dots, n$) representing a NURBS surface
- (b) A set of $eAR/cutter$ $\{eAR_i/T_i, i = 1, 2, \dots, N\}$, with each eAR being represented by a subset of $\{\mathbf{P}_j\}$
- (c) A-map at each sampled point with its corresponding cutter in eAR_i/T_i
- (d) Tool path direction angle range $[\alpha_{\min}, \alpha_{\max}]$

Output: Optimal cutting direction α_{ω}

BEGIN

- (1) Set $k = 1$.
- (2) Pick eAR_k/T_k and set it as C- eAR .
- (3) Pick a sampled point \mathbf{P}_j inside C- eAR .
- (4) Uniformly sample $[\alpha_{\min}, \alpha_{\max}]$ into $(m+1)$ angles, set $i = 0$.

- (5) Set $\alpha_{\omega i} = \alpha_{\omega min} + (\alpha_{\omega max} - \alpha_{\omega min})(i/m)$. Transform $\alpha_{\omega i}$ into the local frame as α_i .
 - (6) Obtain (θ_i, λ_i) by choosing θ_i from the A-map that is closest to the α_i , and λ_i on the lower bound of A-map at θ_i . Calculate the machining strip width W_{ij} and save the data pair: $(\alpha_{\omega i}, W_{ij})$. Set $i = i + 1$. If $i \leq m$, go to (5).
 - (7) Mark \mathbf{P}_j as checked.
IF there is still any unchecked sampled point in C-*eAR*, go to (3)
ELSE
 Mark C-*eAR* as checked.
ENDIF
 - (8) Set $k = k + 1$. If $k \leq N$, go to (2).
 - (9) Set $i = 0$.
 - (10) $\alpha_{\omega i} = \alpha_{\omega min} + (\alpha_{\omega max} - \alpha_{\omega min})(i/m)$.
 - (11) Calculate the average of the W_{ij} over all the points in $\{\mathbf{P}_j\}$ and record it as $Ave(W)_i$. Save the data pair $(\alpha_{\omega i}, Ave(W)_i)$. Set $i = i + 1$. If $i \leq m$, go to (10).
 - (12) Set $\alpha_{\omega i}$ which achieves the maximum value of $Ave(W)_i$ the optimal feeding direction α_{ω} , e.g., $\alpha_{\omega} = \{ \alpha_{\omega} | Ave(W)_{\omega} = \max\{ Ave(W)_i \} \}$. Stop.
- END**

5.2.2 CL data generation

With the cutting direction selected, from the largest cutter to the smallest one, CL paths are generated for each of the *eARs*. For a given *eAR*, the algorithm to generate the CL paths is basically the same as the one described in Sections 4.5 and 4.6 except that the boundary of an *eAR* is represented by sampled points (not a continuous curve). Therefore, this algorithm of CL data generation for multi-cutter machining is only briefly described as follows:

- (1) For the sake of convenience, all the sampled points are represented in the cutting frame (see Figure 5.3, where f is the cutting direction) in which Y -axis is in line with the normal of the cutting plane.
- (2) Start from eAR_1/T_1 (see Figure 5.3a), the boundary of the eAR is represented by the connected sampled points. From the boundaries information, the minimum y_{\min} and maximum y_{\max} of the eAR are obtained.
- (3) Along the cutting direction, the first cutting plane (for the first CL path) $y = y_0$ is selected with a small offset (e.g., $0.1R$) from y_{\min} (see Figure 5.3a). The first CC point is the first intersection point (lower end) between the cutting plane and the eAR boundary. The selection of the cutter posture at this CC point and the determination of the next CL is the same as the one in Section 4.5.1. In this way, the first CL path is generated.
- (4) After completing the generation of one CL path on cutting plane $y = y_i$, the step-over Δy_i for the next path (see Figure 5.3b) is calculated using the method presented in Section 4.6. The next cutting plane is then set plane $y = y_i + \Delta y_i$ and the CLs for the next path are generated by following the procedure in (3). This process is repeated until the cutting plane exceeds y_{\max} . The CL paths for this eAR are said generated.
- (5) The above steps are repeated to generate the CL paths for the remaining eAR s, one at a time. Finally, the CL data are converted from the cutting frame to the global frame.

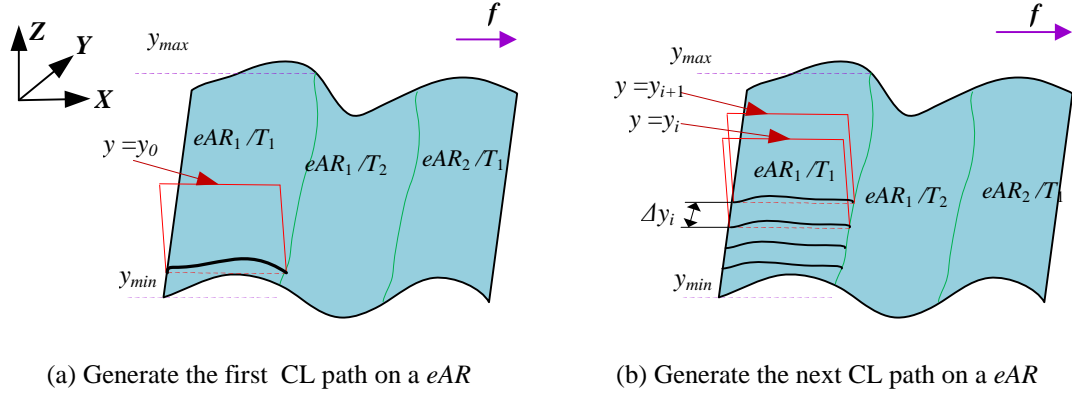


Figure 5.3 CL path generation for multi-cutter machining

The overall algorithm on CL paths generation is given as follows:

Algorithm: CL paths generation for multi-cutter

- Input:** (a) A set of sampled points $\{\mathbf{P}_j\}$ ($j = 0, 1, \dots, n$) representing a surface
- (b) A set of eAR/cutter $\{eAR_i/T_i, i = 1, 2, \dots, N\}$, with each eAR being represented by a subset of $\{\mathbf{P}_j\}$
- (c) A-map at each sampled point for the cutter set
- (d) Optimal path direction α_ω
- (e) Machining profile tolerance τ and h
- (f) Cutter posture change rate τ_{PCR}

Output: CL-paths with a set of CL data

BEGIN

- (1) Transfer the sampled points into the cutting frame.
- (2) Set $k = 1$.
- (3) Pick eAR_k/T_k and set it as C-eAR.
- (4) Extract the boundaries for C-eAR represented by a subset of $\{\mathbf{P}_j\}$. Find y_{min} and y_{max} . Set $d = 0.1R$, where R is the major radius of T_k .
- (5) Set $y_j = y_{min} + d$ and plan the CL path on the cutting plane $y = y_j$ by following:

- (a) Set the first CC point on the boundary and record as the current CC point \mathbf{P}_i .
 - (b) Estimate the maximum step-forward length L_i and the initial \mathbf{P}_{i+1} . Set $p = 0$.
 - (c) Calculate PCR_i .

IF $\text{PCR}_i < \tau_{\text{PCR}}$

Record \mathbf{P}_{i+1} and its posture as the next CL.

ELSE IF $p < N_{\text{max}}$ (where N_{max} is the maximum number of iterations, e.g. 20)

Record the pair $\{\mathbf{P}_{i+1}, \text{PCR}_i\}_p$ and set $p = p + 1$. Set $L_i = 0.95 L_i$ and calculate the new position of \mathbf{P}_{i+1} . Go to (c).

ELSE

Set $\mathbf{P}_{i+1,p}$ as the valid next CC point \mathbf{P}_{i+1} where p is the one achieves the minimum value of $\text{PCR}_{i,p}$, e.g., $\mathbf{P}_{i+1} = \{ \mathbf{P}_{i+1,p} \mid \text{PCR}_{i,p} = \min\{\text{PCR}_{i+1}, \text{PCR}_i\}_p \}$.

END IF
 - (d) If \mathbf{P}_{i+1} does not reach the boundary, set it as \mathbf{P}_i and go to (b).
 - (6) Calculate the maximum allowable step-over length d and set $y_j = y_{\min} + d$.

IF $y_j < y_{\max}$

Go to (5).

ELSE

The path generation for C-*e*AR is complete. Go to (2).

ENDIF
 - (7) Set $k = k + 1$. If $k \leq N$, go to (3).
 - (8) Transfer the CL paths into the global frame and output the CL paths.
- END**

5.2.3 Case study on multi-cutter CL path generation

The proposed algorithm on multi-cutter CL path generation has been implemented using C++ and OpenGL. The case study presented in multi-cutter selection (shown in Figure 3.10) is used here for multi-cutter CL paths generation to show the completeness and effectiveness of the developed algorithms. For further verification on the efficiency of multi-cutter machining, the iso-planar CL paths for the following two modes were generated: (1) using optimal cutter set $\{T_1, T_5, T_8\}$ and (2) using T_8 only (Note T_8 is the largest accessible cutter to the whole surface). The profile tolerance and the scallop height tolerance were set as 0.1 mm. The PCR tolerance is set as 0.6. The generated tool-paths for these two cases are shown in Figures 5.4a and 5.4b, respectively.

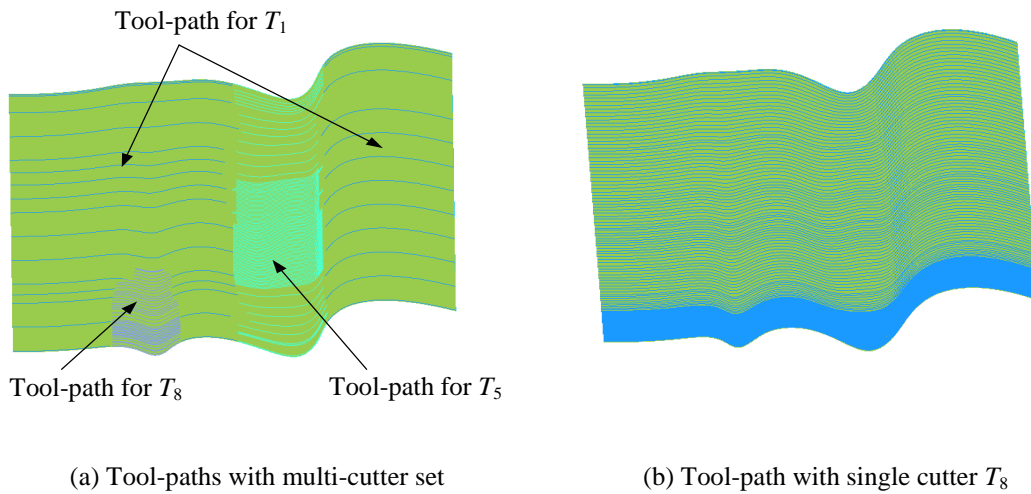


Figure 5.4 Iso-planar tool-paths with *multi-cutter* and *single-cutter* modes

Table 5.1 shows the comparison results of these two generated CL paths in terms of total path lengths and PCR. The total path length indicates the quality of machining efficiency and the PCR (maximum and average) indicates the smoothness of the CL paths. It can be seen that the overall CL path length for using $\{T_1, T_5, T_8\}$ is 8215.43mm, while the overall CL path length for using C_8 only is 33787.70mm.

Assuming the same feed rate is allied to both cases, using $\{T_1, T_5, T_8\}$ can achieve a reduction of machining time by 76%. Even considering the cutter change time and air travel time of the cutters, this saving is still considered quite significant. Therefore, it is concluded that in this case study, the use of the optimal multi-cutter set outperformed the use of the single optimal cutter in terms of machining time. It is also noted that the average PCR along each CL path is very much comparable. This is understandable since the CL path generation algorithm used for each cutter/surface is the same. On the other hand, it is noted that the maximum PCR for every CL path exceeded the PCR tolerance. This is caused by the sharp curvature changes of the surface at certain portions and it is virtually impossible to avoid it completely. In our algorithm, the search will stop after the pre-set maximum number of iterations is reached, and the CL with the least PCR among all the iterations is chosen.

Table 5.1 Tool-path comparison

	Cutter	Number of passes	Number of CC points	Path length (mm)	Average PCR	Maximum PCR
Multi-cutter machining	T_1	33	391	1957.87	0.06	0.8
	T_5	114	1568	5047.75	0.13	0.66
	T_8	41	853	1209.82	0.15	1.22
	(T_1, T_5, T_8)	188	2812	8215.43	0.13	1.22
Single cutter machining	T_8	163	22716	33787.70	0.06	1.27

5.3 An Integrated Process Planning System for Multi-Cutter 5-axis Machining

The prototype integrated process planning system for 5-axis sculptured surface machining with multi-cutter has been finally developed using C++ with OpenGL. The system contains the algorithms on A-map construction, cutter selection and surface partition, and CL path generation. This chapter gives an overview on the running of the system coupled with an comprehensive case study.

As shown in Figure 5.5, the part model is composed of machining surface and un-machining surfaces. The complex geometric features of the model are designed purposely to verify the developed algorithms. On the machining surface, the ditch underneath the overhang has non-uniform cross-sectional curvatures, where the curvature increases from one end to the other end. This feature is designed to induce gouging and drastic posture changes in the machining process. The arch-shaped un-machined part above the machining surface serves as an obstacle that may cause severe collision problems. The detailed surface information is given in Appendix A.

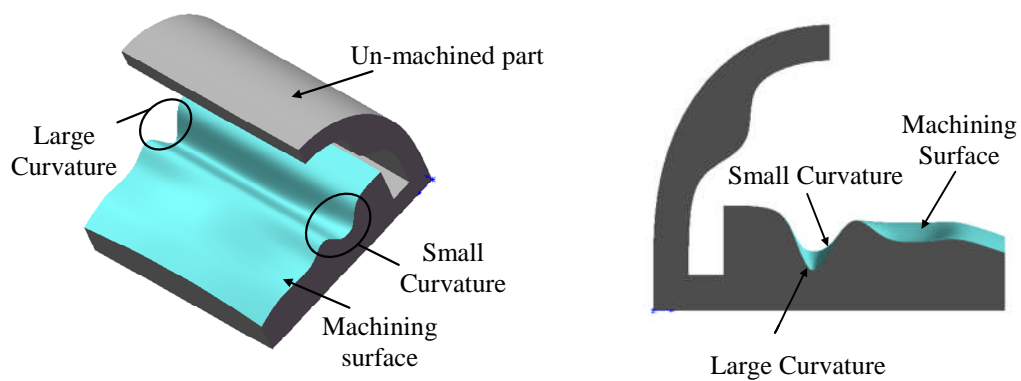


Figure 5.5 The part model

5.3.1 The main interface

The main interface of the system is shown in Figure 5.6. For each option, the functionality is briefly described as follows:

- File: allow user to import and export the data, e.g., to input the surface, cutter library, to save/load the A-maps, load the cutter/*eAR*, save the CL-path etc.
- View: to display different views of the surface, including zoom in/out and rotate.
- Surface: to display the decomposed surfaces, e.g., surface decomposition by its geometry features (convex, saddle or concave), surface decomposition for a given cutter's accessibility, and etc.

- Tolerance: to specify the surface offsets for A-maps calculation and tolerances for tool path generation.
- Point-accessibility: to display a given cutter's accessibility at a specified surface point.
- Cutter: to search for a single optimal cutter or an optimal multi-cutter set.
- Single-Paths: to generate the CL path for a single specified cutter.
- Multi-Paths: to generate the CL paths for a multi-cutter set machining.

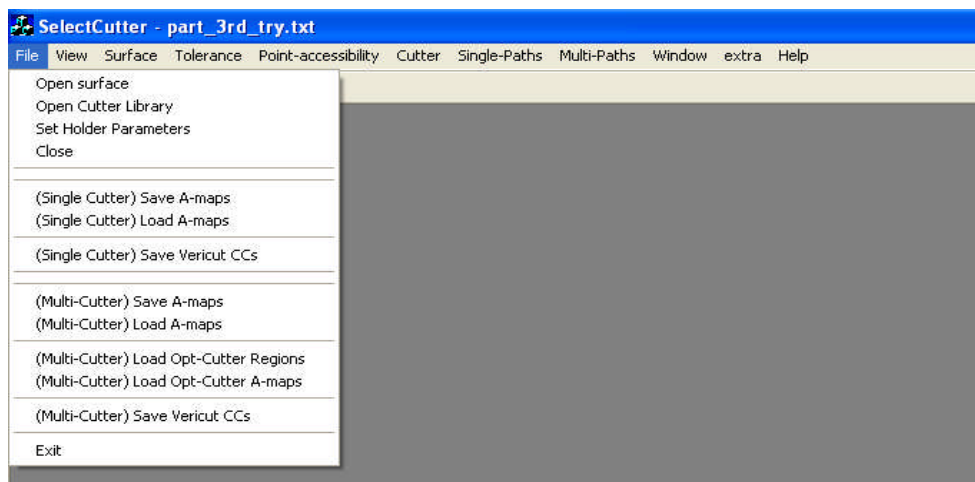


Figure 5.6 Main menu bar

5.3.2 The input to the system

The inputs for a process planning task include the design model, stock model, cutter library, profile tolerance, scallop height tolerance, and PCR tolerance. Under the “File”, the “Open surface” allows the user to browse for the file of the desired part. The “Open cutter library” allows the user to input the cutter library file. The result is shown in Figure 5.7.

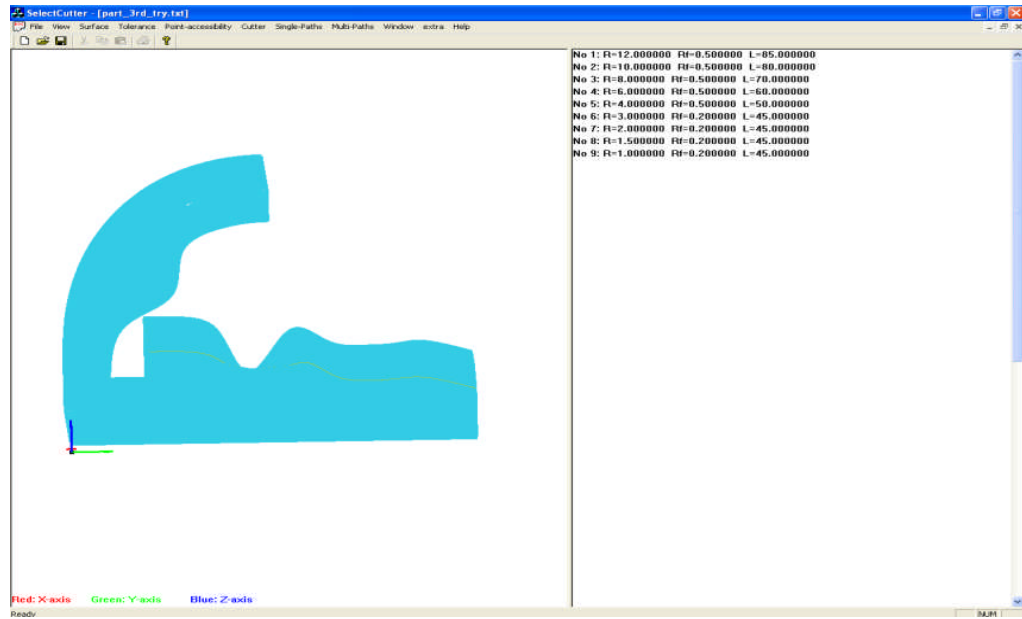


Figure 5.7 System input: part surface and cutter library

Subsequently, the pre-set tolerances are input into the system using the “Tolerance” (see Figure 5.8). The first one, shown in Figure 5.9a, gives the option to specify the offsets value from the design surface for the part surfaces used in A-map calculation. In this case study, the surface with 0.1mm offset is used for gouging checking and the one with 0.15mm offset is used for global collision checking. The second one (Figure 5.9b) is used to specify the profile tolerance, scallop height tolerance, and PCR tolerance used in tool path generation. Here we use 0.1mm as profile tolerance and also for scallop height, while 0.4 for PCR limit along each path.

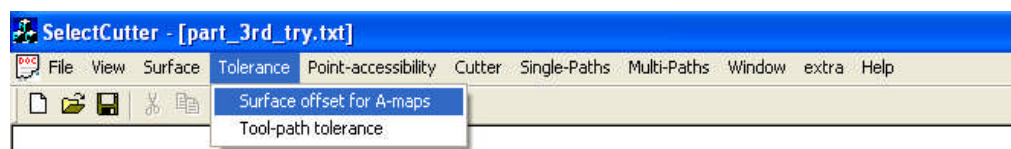
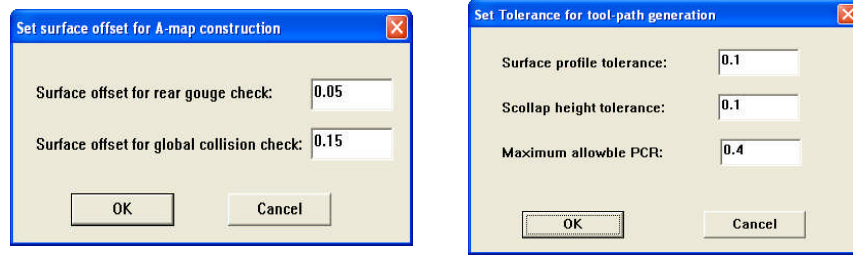


Figure 5.8 Access to system input: tolerances



(a) Surface offsets for A-maps calculation (b) Tolerance for tool-path generation

Figure 5.9 System input: tolerances

5.3.3 Display of a cutter with a specified posture at a surface point

To verify the A-map construction for a cutter at a specified surface point, a visual display of the cutter position on the surface is provided in the system. After selecting “Point-accessibility”, the window in Figure 5.10 pops up to input the point of interest together with the cutter index. There are two ways to specify the point: by sampled point index or by (u, v) . Here the point $(u, v) = (0.4, 0.1)$ is used. The initial value of θ can be specified. The corresponding cutter position at $\lambda=0$ is shown in Figure 5.11. The accessible range of λ for LG-free, RG-free and GC-free is calculated individually and the common accessible λ range is given as $[67.763089^\circ, 90^\circ]$. It can be seen that interference at this posture exists. To further verify the accessible posture range, the cutter is rotated to an accessible posture $(0^\circ, 68^\circ)$ (shown in Figure 5.12). It can be seen there is no overlapped area between the cutter and the surface to cause any type of interference. In addition, the intersection area calculation also justifies this result.

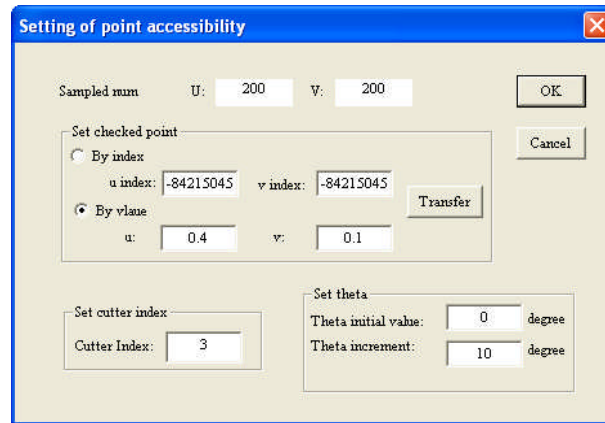


Figure 5.10 Specifying the point of interest

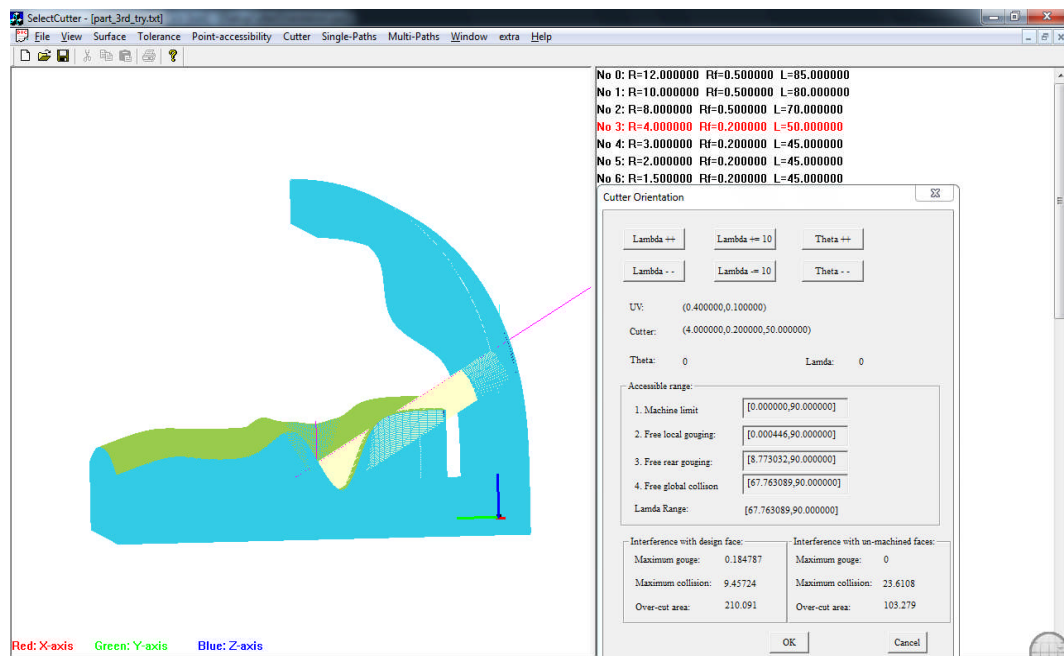


Figure 5.11 Cutter with posture (0,0) at point (0.4, 0.1)

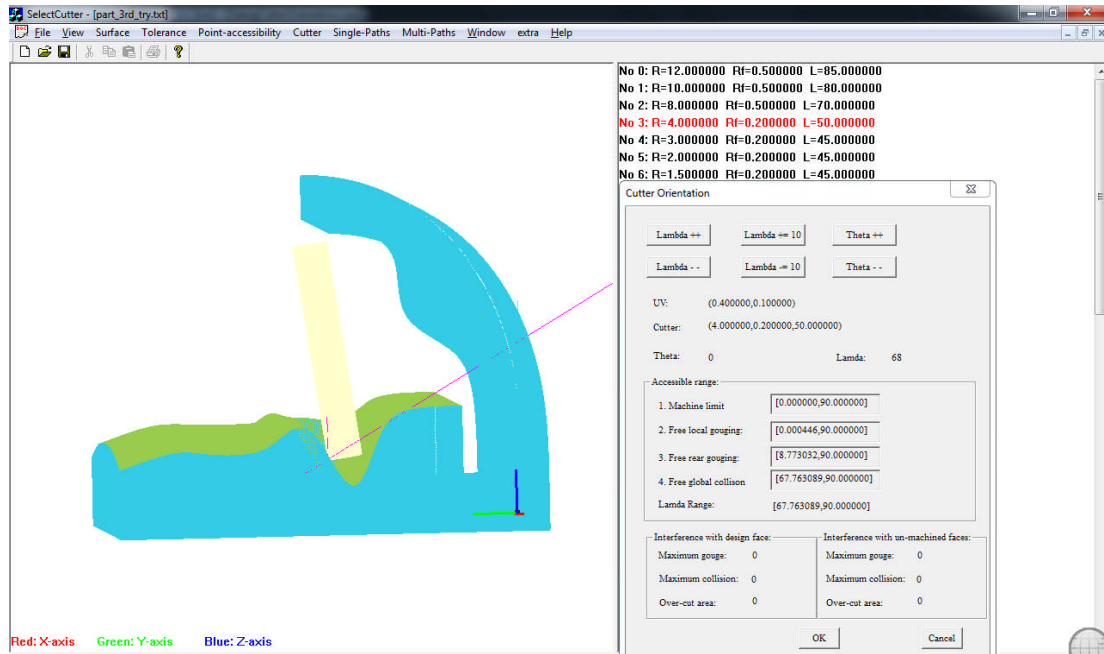


Figure 5.12 Cutter with posture (0, 68°) at point (0.4, 0.1)

5.3.4 Optimal multi-cutter set selection

During A-map construction, the system may take quite some time, depending on the number of cutters and the number of the sampled surface points. Since the A-map construction needs to be completed for every cutter at every sampled point, the A-map is computed and saved, one cutter/point at a time, by using “File->(Multi-cutter) Save A-maps” in the menu.

To start the multi-cutter set selection, the saved A-maps for all the cutter/point are first retrieved using command “File->(Multi-cutter) Load A-maps”. By clicking “Cutter->Select multi-cutter”, a dialogue box (shown in Figure 5.13) pops up for specifying MR_{min} . The output of the multi-cutter selection is displayed in Figure 5.14. In addition to the optimal cutter set, the display also includes the percentage of each cutter's $eARs$ and the estimated total tool path length. In this case, the optimal multi-cutter set is found as $\{T_5, T_9\}$ (the largest cutter that can finish the whole surface is T_9). In this cutter set, $T_5.eARs$ and $T_9.eARs$ take 78.63% and 21.37% of the whole surface,

respectively. Based on the tool-path length estimation model, compared to machining with a single cutter T_9 , the reduced machining time by using $\{T_5, T_9\}$ is also displayed, which is up to 45.44%. The eAR /cutter information is then saved for tool-path generation in the later stage.

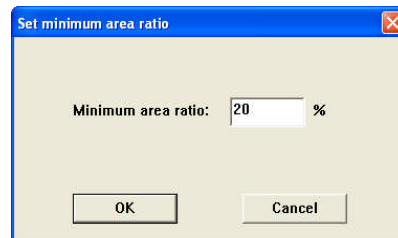


Figure 5.13 Dialog box: specifying MR_{min}

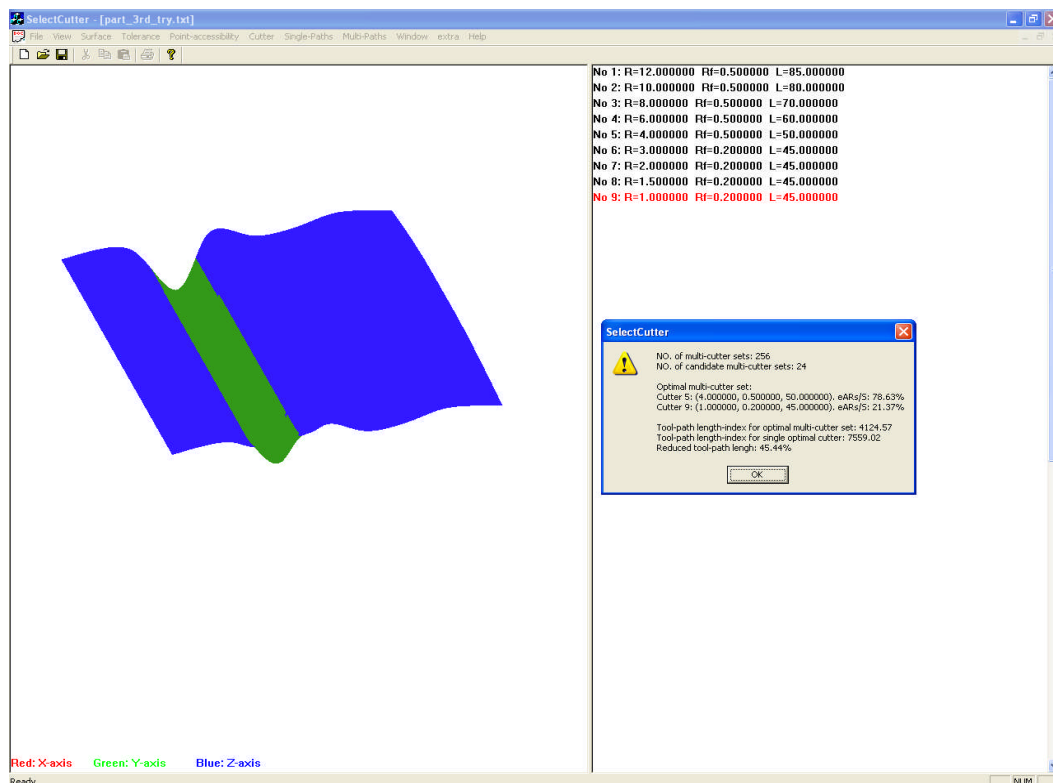


Figure 5.14 Output of the optimal multi-cutter set

5.3.5 Multi-cutter CL path generation

After the multi-cutter set selection, the next step is to generate the CL paths for this cutter set. Firstly, the eAR /cutter for the optimal multi-cutter set is retrieved using “File->(Multi-Cutter) Load Opt-Cutter Regions” and the A-maps are retrieved by “File->(Multi-Cutter) Load Opt-Cutter A-maps”. Subsequently, the iso-planar CL paths can be generated by using the options under “Multi-Paths” (see Figure 5.15).

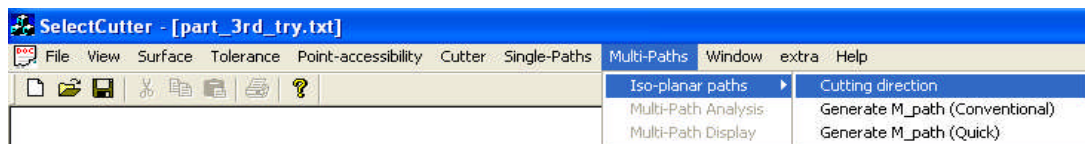


Figure 5.15 Generate CL paths for optimal multi-cutter set

Firstly, the optimal cutting direction is selected by using “Multi-Paths->Cutting direction”. The selection result is shown in a pop-up window (see Figure 5.16).

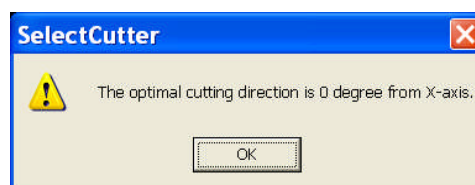


Figure 5.16 The identified optimal cutting direction

Secondly, the CL paths are generated by using either “Generate M_path (Coventional)” or “Generate M_path (Quick)” as shown in Figure 5.14. The main difference between these two modes is the way of computing the optimal cutter posture at a CC point. Under the “Quick” mode, the CC point’s cutter posture is obtained using the interpolation of its neighboring points’ known A-maps; while under “Conventional”, the posture needs to be re-calculated. Due to the heavy computation load in the A-map construction, the “Conventional” mode takes much

longer time compared to the “Quick” one. Here, the “Quick” mode is selected, and a dialog box (see Figure 5.17) for cutting direction specification pops up. It provides two options for tool-path generation: using the optimal cutting direction or any user-specified direction. This option offers more flexibility on tool-path generation.

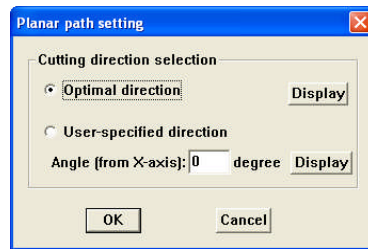


Figure 5.17 Specify the cutting direction

The CL paths for $\{T_5, T_9\}$ are then generated and the results are displayed in Figure 5.18. By clicking “Multi-Paths->Multi-Path Analysis”, the overall CL path information for a particular cutter can be displayed in the right bottom panel.

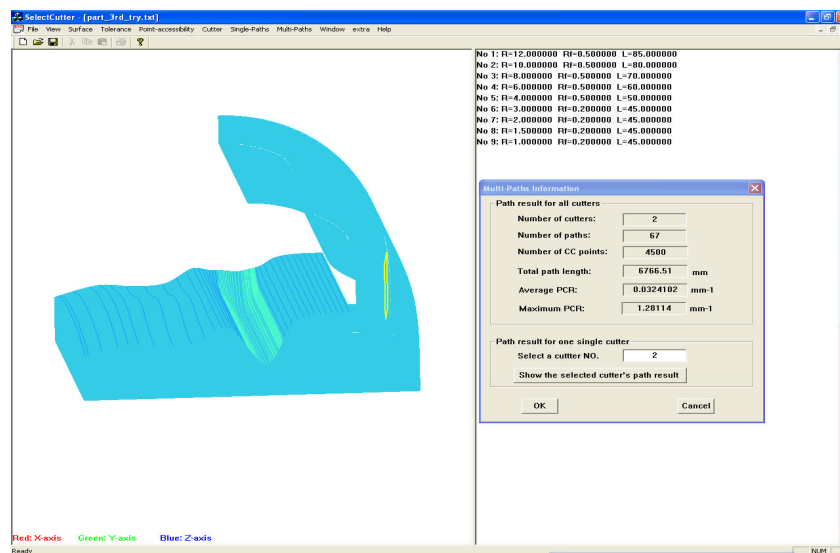


Figure 5.18 The generated iso-planar CL paths for multi-cutter set machining

For direct comparison, the similar process is also applied to obtain the CL path for using the single largest accessible cutter (T_9). The result is shown in Figure 5.19.

Table 5.2 shows the summary of the comparison results between these two generated CL paths in terms of total path lengths and PCR. It can be seen that the overall CL path length using $\{T_5, T_9\}$ is much less than the overall CL path length for using T_9 . Under the assumption of same feeding rate and neglecting the time for tool change and air travel, the actual time save estimated directly from the tool-path length reduction is as much as 45.13%.

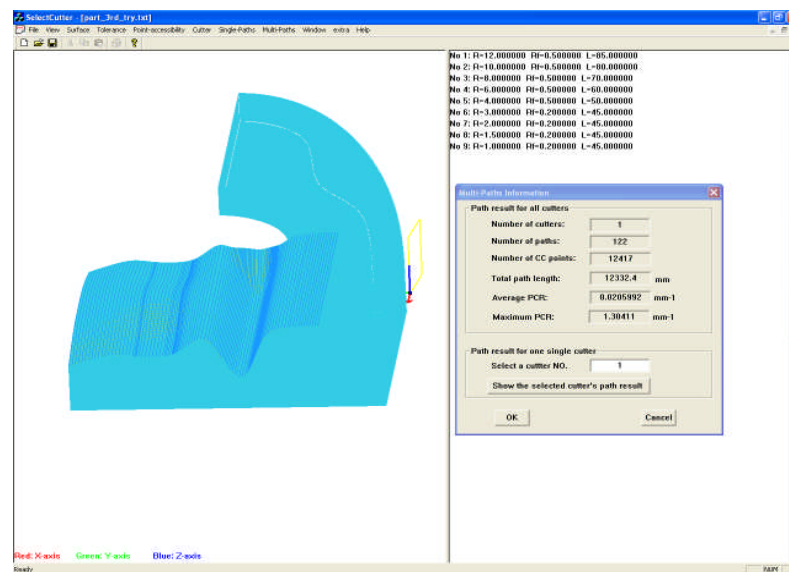


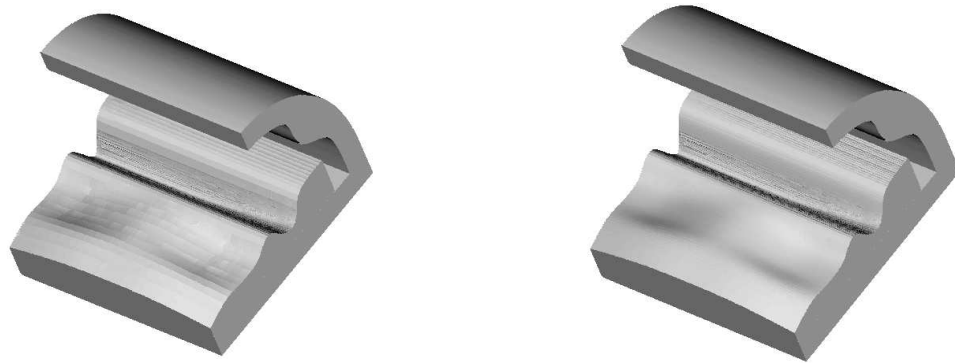
Figure 5.19 Iso-planar CL paths using a single accessible cutter (T_9)

Table 5.2 CL path comparison

	Cutter	Number of passes	Path length (mm)	Average PCR	Maximum PCR
Multi-cutter machining	T_5	31	3108.32	0.011	0.371
	T_9	36	3658.19	0.037	1.281
	(T_5, T_9)	67	6766.51	0.032	1.281
Single cutter machining	T_9	122	12332.40	0.021	1.304

Finally, the generated CL data can be exported by clicking on “File->(Multi-Cutter) Save Vericut CCs” for further simulation verification. In this research, the machining simulation experiment is conducted in the VERICUT[®] environment. This software can simulate the CNC machines as they behave on the shop floor and detect

errors and potential problems before the program goes out to the shop floor. For this case study, machining simulations by using the generated CL data from both multi-cutter machining and single cutter machining are conducted. The resulted machined surfaces are shown in Figure 5.20a and 5.20b, respectively. The AUTODIFF module is then run for the two machined surfaces, which showed that in both cases, machining errors are limited within the specified tolerance.



(a) Machining simulation result with multi-cutter

(b) Machining simulation result with single cutter

Figure 5.20 Machining simulation result for multi-cutter and single cutter modes

CHAPTER 6

CONCLUSION AND FUTURE WORK

The main objective of this study is to develop new methodology towards process planning optimization for 5-axis sculptured surface milling (finish cut). The specific process planning issues addressed in this thesis include multi-cutter selection and CL path generation. As a result, an algorithm for the selection of the optimal multi-cutter set that can achieve maximum cutting efficiency has been developed. The output of this algorithm includes a list of cutters, and the effective cutting areas assigned for each individual cutter. Furthermore, for a given cutter/cutting-area, an algorithm for the generation of iso-planar CL path has been developed. This algorithm has two incorporated objectives: maximum cutting efficiency and smooth cutter dynamics. The main work of this research is concluded and the possible directions for future work are presented in this chapter.

6.1 Conclusions

The main achievements of this study include the following aspects: (1) improvements on the algorithm of A-map construction; (2) identification of the cutting area for a cutter; (3) selection of the optimal multi-cutter set; (4) optimization on iso-planar tool-path generation; and (5) integration process planning for multi-cutter set machining.

- Since A-map plays a fundamental role in the optimization of process planning, some improvements on the algorithm of calculating A-map at a surface point have been implemented. In our previous study, the A-map was constructed based on the nominal surface (design surface). To make the A-map result more reliable, the effects of surface tolerance and stock surface have been taken into consideration. The improved A-map construction algorithm has been proved to be able to achieve a more reliable result. In addition, the computation efficiency has been improved by adding two quick filters on the A-map calculation for LG avoidance.
- For a given cutter, based on the A-map information, a method for identifying its cutting areas has been developed. There steps were involved. Firstly, based on the A-map information, the cutter's accessible points (*APs*) are extracted. Secondly, with the set of *APs*, the boundaries of the accessible areas are traced and represented in a set of connected *APs*. Note that the points on the boundaries were interference-free. Thirdly, by analyzing the inclusiveness of these boundaries, the cutting areas are identified and represented by a set of *APs*. In addition, refinements are applied on the *APs* such that the boundaries are smoother.

Furthermore, give a multi-cutter set, a method for identifying effective cutting region (*eAR*) for each cutter has also been developed. In this method, for a cutter set ordered from large to small, the *eAR* extraction is achieved in three steps. Firstly, from the A-maps information, the set of effective accessible points (*eAPs*) for each cutter are extracted in a sequential way. Secondly, for each cutter, the boundaries of the *eARs* are traced and represented in a set of connected *eAPs*. Finally, for each cutter, by analyzing the inclusiveness of its boundaries, every *eAR* is identified and represented by a set of unique *eAPs*.

- To take the advantage of the cutting efficiency potential of larger cutters while maintaining the machined surface accuracy, the study has proposed an algorithm for multi-cutter set selection for finish cut of sculptured surfaces. In this algorithm, the feasible cutters are extracted from the cutter library, followed by the construction of all the valid cutter sets. The candidate cutter sets are then extracted from the valid sets by satisfying the condition of the minimum accessible area of each *eAR*. For each candidate multi-cutter set, to evaluate the cutting efficiency, a novel tool path length estimation model for 5-axis machining has been developed. In this model, the path length is estimated from the theoretical analysis of machining strip width without generating the actual tool path. By using this model, the cutting efficiency of different multi-cutter sets can then compared and the optimal multi-cutter set identified.

The main advantage of this cutter selection algorithm is that it separates the multi-cutter selection task from the CL path generation task. This has been reflected in several aspects. Firstly, only the comparison between the cutter geometry and part surface geometry is involved in the cutter selection. There is no specification on the cutting direction. Secondly, for each candidate multi-cutter set, the cutting time is estimated from the proposed model on calculating the tool-path length index without generating the actual too-path. Without this model, the cutting time had to be evaluated from the actual CL path for each candidate cutter set, which would suffer from heavy computational load and lead the multi-cutter selection algorithm impractical. The practice of selecting the cutters before generating the CL paths provides full flexibility to the subsequent tool path optimization task. On the other hand, the output of the multi-cutter selection algorithm, e.g., A-maps and the set of *eAR*/cutter, also provide useful information for CL path generation.

- Given a cutter/cutting region and the A-maps, an optimization algorithm has been developed to generate the iso-planar CL paths, aiming at maximum cutting efficiency and smooth cutter dynamics. The proposed method included two steps: (1) identifying the optimal cutting direction, and (2) generating the CL paths on each cutting plane. By using the A-map information, the objective of maximum cutting efficiency is achieved by selecting the optimal cutting direction and specifying the cutter posture at each CC point that resulted in maximum cutting efficiency, while the smooth cutter dynamics is achieved by keeping the posture change rate between two consecutive CC points along a path under control.

Compared to our previous method, which took the optimal cutting direction as the one that achieves smooth cutting dynamics over the whole surface, the presented method is more reasonable as the cutting dynamics is mainly affected by the posture change along one single path. In addition, the case study has shown that machining with the generated CL-paths achieved both good machining efficiency and smooth cutter dynamics.

- The integrated process planning system has been implemented in the Visual C++ and OpenGL environment. Various case studies have been conducted that proved the developed algorithms are able to select a good cutter set and generate CL paths with very good cutting efficiency and smoothness. Compared with machining using a single cutter, the use of multi-cutter set can improve the cutting efficiency tremendously. The developed process planning system has provided a powerful tool to make the multi-cutter practice possible.

6.2 Future Work

Some future work is recommended as follows:

- In the algorithm for GC avoidance, the effect of the cutter's holder is not considered due to the unknown geometry of the holder. However, if the geometry of the holder is given with a regular shape, an algorithm that is similar to global collision avoidance of the cutter's body could be developed to incorporate the constraints of the holder.
- Due to the approximations involved in the tool path length estimation model, the path length might have some discrepancies with the actual length. A more powerful and robust tool-path estimation method is needed.
- For maximum machining strip width (W), the proposed method on optimal posture selection at a CC point is specified by two steps: (1) specifying θ as the one closest to the feeding angle α , named as θ_α ; and (2) specifying λ as the lower-bound at θ_α within A-maps, named as λ_{θ_α} . This method is simple and possible to obtain the near-optimal posture. However, there could be another posture (with a different θ and a different λ) that achieves a larger W due to the fact that, W is determined by both θ and λ while in this algorithm θ_α was specified without considering the effect of λ on W . Thus, a more accurate optimal posture could be selected by comparing W on all the sampled θ , instead of only θ_α . This is also practically possible since the known A-map could largely alleviate the computational load in interference checking.
- Only iso-planar CL path pattern is addressed in this study. For some kind of surfaces, to achieve a certain machining objectives, other path patterns, such as contour and constant scallop height might be a more suitable choice. One possible future work direction could be the investigation of the optimal path patterns based on the analysis of the surface features.

REFERENCES

- Aomura S, and Uehara T, 1990, Self-intersection of an offset surface, *Computer-Aided Design*, 22(7): 417–421.
- Arya S, Cheng S. and Mount D., 2001, Approximate algorithm for multiple-tool milling, *International Journal of Computational Geometry and Applications*, 11(3): 339–372.
- Balasubramaniam M, Laxmiprasad P, Sarma SE, and Shaikn Z, 2000, Generating 5-axis NC roughing paths directly from a tessellated representation, *Computer-Aided Design*, 32(4): 261-277.
- Balasubramaniam M, Sarma SE, and Marciniak K, 2003, Collision-free finishing toolpaths from visibility data, *Computer-Aided Design*, 35(4): 359-374.
- Chen T, Ye PQ, and Wang JS, 2005, Local interference detection and avoidance in five-axis NC machining of sculptured surfaces, *International journal of Advanced Manufacturing Technology*, 25: 343-349.
- Chiou CJ, 2004, Accurate tool position for five-axis ruled surface machining by swept envelope approach, *Computer-Aided Design*, 36(10): 967-974.
- Chiou CJ, and Lee YS, 2002, A machining potential field approach to tool path generation for multi-axis sculptured surface machining, *Computer-Aided Design*, 34(5): 357-371.
- Choi BK, and Jerard RB, 1998, *Sculptured Surface Machining: Theory and Application*, Kluwer Academic Publishers, Dordrecht/Boston/London.

- Choi BK, Kim DH, and Jerard RB, 1997, C-space approach to tool-path generation for die and mold machining, *Computer-Aided Design*, 29(9): 657-669.
- D'Souza RM, Sequin C, and Wright PK, 2004, Automated tool sequence selection for 3-axis machining of free-form pockets, *Computer-Aided Design*, 36(7): 595-605.
- Elber G, 1995, Freeform surface region optimization for 3-axis and 5-axis milling. *Computer-Aided Design*, 27 (6): 455-470.
- Guggenheimer HW, 1977, *Differential Geometry*, Dover, New York.
- Held M, 1991, A geometry-based investigation of tool-path generation for zigzag pocket machining, *Visual Computer*, 7: 296-308.
- Hwang JS, 1992, Interference-free tool-path generation in the NC machining of parametric compound surfaces, *Computer-Aided Design*, 24(12): 667-676,.
- Jensen CG, and Anderson DC, 1993, Accurate tool placement and orientation for finish surface machining. *Journal of Design and Manufacturing*, 3: 251-261.
- Jensen CG, Red WE, and Pi J, 2002, Tool selection for five-axis curvature matched machining, *Computer-Aided Design*, 34(3): 251-266,.
- Jun CS, Cha K, and Lee YS, 2003, Optimizing Tool Orientations for 5-Axis Machining by Configuration-Space Search Method, *Computer-Aided design*, 35(6): 549-566.
- Kim BH, and Choi BK, 2002, Machining efficiency comparison direction-parallel tool-path with contour-parallel tool-path, *Computer-Aided Design*, 34(2): 89-95.
- Lauwers B, Dejonghe P, and Kruth JP, 2003, Optimal and collision free tool posture in five-axis machining through the tight integration of tool path generation and machine simulation, *Computer-Aided Design*, 35(5): 421-432.

- Lee YS, 1998, Non-isoparametric tool path planning by machining strip evaluation for 5-axis sculptured surface machining, *Computer-Aided Design*, 30(7): 559-570.
- Lee YS, and Chang TC, 1992, A contour growth method from inters of hunting planes to evaluate machining information. *Journal of Design and Manufacturing*, 2(3): 119–133.
- Lee YS, and Chang TC, 1995, 2-Phase approach to global tool interference avoidance in 5-axis machining, *Computer-Aided Design*, 27(10): 715-729.
- Lee YS, and Chang TC, 1996 Automatic cutter selection for 5-axis sculptured surface machining, *International Journal of Production Research*, 34(4): 977-998.
- Li H, and Feng HY, 2004, Efficient five-axis machining of free-form surfaces with constant scallop height tool-paths, *International Journal of Production Research*, 42: 2403-2417.
- Li. LL, 2007, *Process planning for five-axis milling of sculptured surfaces*, PHD Thesis, National University of Singapore.
- Li SX, and Jerard B, 1994, 5-axis machining of sculptured surfaces with a flat-end cutter, *Computer-Aided Design*, 26(6): 165-178.
- Li. LL, and Zhang YF, 2006, An integrated approach towards process planning for 5-axis milling of sculptured surfaces, *Computer-Aided Design and Application*, 3: 249-258.
- Lim T, Corney J, and Clark DE R, 2000, Exact Tool Sizing for Feature Accessibility, *International Journal of Advanced Manufacture Technology*, 16: 791–802.
- Mahadevan B, Sarma SE, Marciniak K, 2003, Collision-free finishing toolpaths from visibility data. *Computer-Aided Design*, 35(4): 359-374.

- Marciniak K, 1987, Influence of surface shape on admissible tool positions in 5-axis milling, *Computer-Aided Design*, 19(5): 233-236.
- Morishige, K, and Takeuchi, Y, 1999, Tool path generation using C-space for 5-axis control machining, *ASME Journal of Manufacturing Science and Engineering*, 121: 144-149.
- Pi J, Red WE, and Jensen CG, 1998, Grind-free tool-path generation for five-axis surface machining, *Computer Integrated Manufacturing Systems*, 11: 227-350.
- Park SC, 2003, Tool-path generation for Z-constant contour machining, *Computer-Aided Design*, 35(1): 27-36.
- Park SC, and Choi BK, 2000, Tool-path planning for direction-parallel area milling, *Computer-Aided Design*, 32(1): 17-25.
- Park SC, and Choi BK, 2001, Boundary extraction algorithm for cutting area detection. *Computer-Aided Design*, 33(8): 571–579.
- Radzevich SP, 2005, A cutting-tool-dependent approach for partitioning of sculptured surface, *Computer-Aided Design*, 37(7): 767–778.
- Rao N, Bedi S, and Buchal R, 1996, Implementation of the principle-axis method for machining of complex surfaces. *International Journal of Advanced Manufacture Technology*, 11: 249-257.
- Sheltami K, Bedi S, and Ismail F, 1998, Swept volumes of toroidal cutters using generating curves, *International Journal of Machine Tools and Manufacture*, 38: 855-870.
- Sun G, Sequin CH, and Wright PK, 2001, Operation decomposition for freeform surface features in process planning, *Computer-Aided Design*, 33(9): 621–636.

- Sun YF, Nee AYC, and Lee KS, 2004, Modifying free-formed NURBS curves and surfaces for offsetting without local self-intersection, *Computer-Aided Design*, 36(12): 1161–1169.
- Veeramani D, and Gau Y, 1997, Selection of an optimal set of cutting tool sizes for 2.5D pocket machining, *Computer-Aided Design*, 29(12): 869–877.
- Wang XC, and Yu Y, 2003, An approach to interference-free cutter position for five-axis free-form surface side finishing milling. *Journal of Materials Processing Technology*, 123: 191–196.
- Yang DCH, and Han Z, 1999, Interference detection and optimal tool selection in 3-axis NC machining of free-form surfaces. *Computer-Aided Design*, 31(5): 303–315.
- Yang JZ, Wang QF, Huang ZD, and Chen G, 2007, Cutting area extraction from a Z-map model. *International Journal of Advanced Manufacture Technology*, 33: 1010–1016.
- Yao Z, Gupta SK, Nau DS, 2003, Algorithms for selecting cutters in multi-part milling problems, *Computer-Aided Design*, 35(9): 825–839.
- Ye T, and Xiong CH, 2008, Geometric parameter optimization in multi-axis machining, *Computer-Aided Design*, 40(8): 879–890.
- Yoon JH, Pottmann H, and Lee YS, 2003, Locally optimal cutting positions for 5-axis sculptured surface machining, *Computer-Aided Design*, 35(1): 69–81.

APPENDIX A

SURFACE WITH STOCK DATA

Note:

The 1st line in the data presents the total number of the un-machined surfaces. The following lines describe the surface one by one, with the first surface as the design surface, followed by each un-machined surface. For each surface,

- (1) The 1st line presents the first surface degrees along u and v direction, respectively.
- (2) The 2nd line presents the numbers (n_i and n_j) of control points along u and v direction, respectively.
- (3) Beginning from 3rd line, the control points $\mathbf{P}_{ij}(x, y, z, w)$ ($i = 0, \dots, n_i; j = 0, \dots, n_j$) are presented. The unit in this example is in cm.
- (4) The 2nd last section presents the knots along u direction.
- (5) The last section presents the knots along v direction.

16

3 3
20 14

10 2 2.913170099 1
9.722221692 2 2.913170099 1
9.166665 2 2.913170099 1
8.333330424 2 2.913170099 1
7.499999951 2 2.913170099 1
6.666669771 2 2.913170099 1
5.833335031 2 2.913170099 1
5 2 2.913170099 1
4.166664969 2 2.913170099 1
3.333330229 2 2.913170099 1
2.500000049 2 2.913170099 1
1.666669576 2 2.913170099 1
0.8333349999 2 2.913170099 1
0.2777783076 2 2.913170099 1
-2.9416e-012 2 2.913170099 1
10 2.204393777 2.915878037 1

9.722221899 2.204393622 2.915878048 1
9.166665495 2.204393569 2.915878052 1
8.3333321 2.204393898 2.915878025 1
7.5000005 2.204393503 2.915878031 1
6.666669452 2.204393915 2.915877977 1
5.833335111 2.204393498 2.915877984 1
5 2.204393917 2.915877943 1
4.166664889 2.204393498 2.915877984 1
3.333330548 2.204393915 2.915877977 1
2.4999995 2.204393503 2.915878031 1
1.6666679 2.204393898 2.915878025 1
0.8333345049 2.204393569 2.915878052 1
0.2777781006 2.204393622 2.915878048 1
-6.098828e-010 2.204393777 2.915878037 1
10 2.61318133 2.921293912 1
9.722222312 2.613180864 2.921293946 1
9.166666484 2.613180707 2.921293958 1
8.333335449 2.613181692 2.921293876 1
7.500001597 2.613180507 2.921293895 1
6.666668813 2.613181745 2.921293732 1
5.83333527 2.613180493 2.921293755 1
5 2.613181749 2.92129363 1
4.16666473 2.613180493 2.921293755 1
3.333331187 2.613181745 2.921293732 1
2.499998403 2.613180507 2.921293895 1
1.666664551 2.613181692 2.921293876 1
0.8333335156 2.613180707 2.921293958 1
0.2777776875 2.613180864 2.921293946 1
-9.457128e-010 2.61318133 2.921293912 1
10 3.167914959 2.880674848 1
9.722223029 3.167916823 2.880674713 1
9.166668677 3.167917449 2.880674664 1
8.333339619 3.167913514 2.880674995 1
7.500003914 3.167918251 2.880674918 1
6.666670791 3.167913299 2.880675568 1
5.833334776 3.167918308 2.880675479 1
5 3.167913285 2.880675976 1
4.166665224 3.167918308 2.880675479 1
3.333329209 3.167913299 2.880675568 1
2.499996086 3.167918251 2.880674918 1
1.666660381 3.167913514 2.880674995 1
0.8333313229 3.167917449 2.880674664 1
0.2777769705 3.167916823 2.880674713 1
2.54551e-010 3.167914959 2.880674848 1
10 3.549860847 2.765740659 1
9.722223353 3.549860218 2.765741165 1
9.166670158 3.549860006 2.76574135 1
8.333339777 3.549861335 2.765740108 1
7.50000538 3.549859735 2.765740395 1
6.666669692 3.549861407 2.765737957 1
5.833335051 3.549859716 2.765738291 1
5 3.549861412 2.76573643 1
4.166664949 3.549859716 2.765738291 1
3.333330308 3.549861407 2.765737957 1
2.49999462 3.549859735 2.765740395 1
1.666660223 3.549861335 2.765740108 1

0.8333298417 3.549860006 2.76574135 1
0.2777766465 3.549860218 2.765741165 1
-6.88978e-011 3.549860847 2.765740659 1
10 3.868261913 2.458232289 1
9.722223248 3.868262567 2.4582304 1
9.166669694 3.868262787 2.45822971 1
8.33333966 3.868261406 2.458234347 1
7.500005141 3.868263068 2.458233275 1
6.666669993 3.868261331 2.458242378 1
5.833334975 3.868263088 2.458241131 1
5 3.868261326 2.45824808 1
4.166665025 3.868263088 2.458241131 1
3.333330007 3.868261331 2.458242378 1
2.499994859 3.868263068 2.458233275 1
1.66666034 3.868261406 2.458234347 1
0.8333303058 3.868262787 2.45822971 1
0.2777767521 3.868262567 2.4582304 1
1.87645e-011 3.868261913 2.458232289 1
10 4.184857794 1.551778031 1
9.722223346 4.184855809 1.551778722 1
9.166670069 4.184855142 1.551779162 1
8.333339971 4.184859334 1.551775282 1
7.500004634 4.184854287 1.551763119 1
6.666669888 4.184859563 1.551746043 1
5.833335002 4.184854226 1.551733602 1
5 4.184859578 1.55172481 1
4.166664998 4.184854226 1.551733602 1
3.333330112 4.184859563 1.551746043 1
2.499995366 4.184854287 1.551763119 1
1.666660029 4.184859334 1.551775282 1
0.8333299305 4.184855142 1.551779162 1
0.2777766537 4.184855809 2.107895244 1
-5.6257e-012 4.184857794 2.424641211 1
10 4.527830825 0.885131424 1
9.722223058 4.527831357 0.8851305473 1
9.166669032 4.527831536 0.8851294784 1
8.333338845 4.527830413 0.885140362 1
7.500006901 4.527831765 0.8851900862 1
6.666670006 4.527830351 0.8852492853 1
5.833334972 4.527831781 0.8853002974 1
5 4.527830347 0.8853285151 1
4.166665028 4.527831781 0.8853002974 1
3.333329994 4.527830351 0.8852492853 1
2.499993099 4.527831765 0.8851900862 1
1.666661155 4.527830413 0.885140362 1
0.8333309675 4.527831536 0.9498333578 1
0.2777769419 4.527831357 1.441247069 1
1.4744e-012 4.527830825 2.287161122 1
10 4.98911296 1.530460145 1
9.722222205 4.989112818 1.530462961 1
9.166665914 4.98911277 1.530466797 1
8.333335683 4.989113071 1.530427143 1
7.500000308 4.989112709 1.530240409 1
6.666669114 4.989113087 1.530020688 1
5.833335195 4.989112704 1.529829081 1
5 4.989113089 1.529725001 1

4.166664805 4.989112704 1.529829081 1
 3.333330886 4.989113087 1.530020688 1
 2.499999692 4.989112709 1.530240409 1
 1.666664317 4.989113071 1.530427143 1
 0.8333340857 4.98911277 1.530466797 1
 0.2777777949 4.989112818 2.086579483 1
 -4.148e-013 4.98911296 2.403323326 1
 10 5.439217334 2.381143639 1
 9.722222091 5.439217373 2.381133255 1
 9.166666925 5.439217385 2.381118978 1
 8.33333026 5.439217305 2.381266712 1
 7.500002119 5.439217402 2.381963923 1
 6.666669354 5.4392173 2.382783608 1
 5.833335135 5.439217403 2.383499024 1
 5 5.4392173 2.383887124 1
 4.166664865 5.439217403 2.383499024 1
 3.333330646 5.4392173 2.382783608 1
 2.499997881 5.439217402 2.381963923 1
 1.66666974 5.439217305 2.381266712 1
 0.8333330748 5.439217385 2.381118978 1
 0.277777909 5.439217373 2.381133255 1
 4.725e-013 5.439217334 2.381143639 1
 10 5.738593689 2.541991168 1
 9.722221496 5.738593678 2.542029894 1
 9.166664037 5.738593675 2.54208316 1
 8.333330508 5.738593697 2.541531879 1
 7.499998865 5.738593671 2.53892977 1
 6.66666998 5.738593698 2.53587075 1
 5.833334979 5.73859367 2.533200696 1
 5 5.738593698 2.531752375 1
 4.166665021 5.73859367 2.533200696 1
 3.33333002 5.738593698 2.53587075 1
 2.500001135 5.738593671 2.53892977 1
 1.666669492 5.738593697 2.541531879 1
 0.8333359629 5.738593675 2.54208316 1
 0.2777785039 5.738593678 2.542029894 1
 -2.771e-013 5.738593689 2.541991168 1
 10 6.00936988 2.450891688 1
 9.722222127 6.009369883 2.450747109 1
 9.166666905 6.009369884 2.45054841 1
 8.33333026 6.009369878 2.452605763 1
 7.500002126 6.009369885 2.462316998 1
 6.666669354 6.009369878 2.473733391 1
 5.833335135 6.009369885 2.483698194 1
 5 6.009369878 2.489103376 1
 4.166664865 6.009369885 2.483698194 1
 3.333330646 6.009369878 2.473733391 1
 2.499997874 6.009369885 2.462316998 1
 1.66666974 6.009369878 2.452605763 1
 0.8333330954 6.009369884 2.45054841 1
 0.277777873 6.009369883 2.450747109 1
 1.403e-013 6.00936988 2.450891688 1
 10 6.312990434 2.197362194 1
 9.722225252 6.312990433 2.196060587 1
 9.166664516 6.312990433 2.194231906 1
 8.333336078 6.312990435 2.209166898 1

7.500000171 6.312990433 2.281840607 1
 6.66666914 6.312990435 2.367435868 1
 5.833335188 6.312990433 2.442340982 1
 5 6.312990435 2.483011288 1
 4.166664812 6.312990433 2.442340982 1
 3.33333086 6.312990435 2.367435868 1
 2.499999829 6.312990433 2.281840607 1
 1.666663922 6.312990435 2.209166898 1
 0.833335484 6.312990433 2.194231906 1
 0.2777747482 6.312990433 2.196060587 1
 -3.846e-013 6.312990434 2.197362194 1
 10 6.691968144 1.93134571 1
 9.722231609 6.691968144 1.929152299 1
 9.166665005 6.691968144 1.926653034 1
 8.333339848 6.691968143 1.955179315 1
 7.500006751 6.691968144 2.09049268 1
 6.666670083 6.691968143 2.249469772 1
 5.833334953 6.691968144 2.388287682 1
 5 6.691968143 2.463552001 1
 4.166665047 6.691968144 2.388287682 1
 3.333329917 6.691968143 2.249469772 1
 2.499993249 6.691968144 2.09049268 1
 1.666660152 6.691968143 1.955179315 1
 0.8333349948 6.691968144 1.926653034 1
 0.2777683909 6.691968144 1.929152299 1
 8.9e-013 6.691968144 1.93134571 1
 10 7.26866292 1.87770076 1
 9.722245686 7.26866292 1.874080095 1
 9.166659921 7.26866292 1.869104973 1
 8.333342977 7.26866292 1.897844249 1
 7.500003386 7.26866292 2.046506762 1
 6.666670082 7.26866292 2.222344243 1
 5.833334953 7.26866292 2.376871085 1
 5 7.26866292 2.461007926 1
 4.166665047 7.26866292 2.376871085 1
 3.333329918 7.26866292 2.222344243 1
 2.499996614 7.26866292 2.046506762 1
 1.666657023 7.26866292 1.897844249 1
 0.8333400792 7.26866292 1.869104973 1
 0.2777543137 7.26866292 1.874080095 1
 -4.8583e-012 7.26866292 1.87770076 1
 10 7.864990093 1.901217158 1
 9.722205927 7.864990093 1.902023975 1
 9.166676914 7.864990093 1.905541566 1
 8.333336655 7.864990093 1.94224343 1
 7.500007596 7.864990093 2.083847612 1
 6.666669857 7.864990093 2.247051028 1
 5.833335009 7.864990093 2.38692078 1
 5 7.864990093 2.461831487 1
 4.166664991 7.864990093 2.38692078 1
 3.333330143 7.864990093 2.247051028 1
 2.499992404 7.864990093 2.083847612 1
 1.666663345 7.864990093 1.94224343 1
 0.8333230859 7.864990093 1.905541566 1
 0.2777940727 7.864990093 1.902023975 1
 1.80442e-011 7.864990093 1.901217158 1

10 8.277171232 1.973770444 1
9.722150991 8.277171232 1.996697824 1
9.166699021 8.277171232 2.043257232 1
8.33332684 8.277171232 2.124121432 1
7.500002686 8.277171232 2.247517807 1
6.666668473 8.277171232 2.367231419 1
5.833335355 8.277171232 2.450631284 1
5 8.277171232 2.488481999 1
4.166664645 8.277171232 2.450631284 1
3.333331527 8.277171232 2.367231419 1
2.499997314 8.277171232 2.247517807 1
1.66667316 8.277171232 2.124121432 1
0.8333009792 8.277171232 2.043257232 1
0.2778490088 8.277171232 1.996697824 1
-6.74989e-011 8.277171232 1.973770444 1
10 8.704645385 1.998969124 1
9.722308113 8.704645385 2.043893781 1
9.166626847 8.704645385 2.133831106 1
8.333340968 8.704645385 2.258575793 1
7.499999106 8.704645385 2.362644497 1
6.666670148 8.704645385 2.437481326 1
5.833334936 8.704645385 2.463186344 1
5 8.704645385 2.463299819 1
4.166665064 8.704645385 2.463186344 1
3.333329852 8.704645385 2.437481326 1
2.500000894 8.704645385 2.362644497 1
1.666659032 8.704645385 2.258575793 1
0.8333731529 8.704645385 2.133831106 1
0.2776918873 8.704645385 2.043893781 1
2.521743e-010 8.704645385 1.998969124 1
10 9.323841037 1.826155413 1
9.722356604 9.323841038 1.875276116 1
9.166601899 9.323841038 1.973590648 1
8.333347252 9.323841037 2.10677658 1
7.499994949 9.323841038 2.207240575 1
6.666671067 9.323841037 2.273582189 1
5.833334707 9.323841037 2.2883079 1
5 9.323841038 2.281228876 1
4.166665293 9.323841037 2.2883079 1
3.333328933 9.323841037 2.273582189 1
2.500005051 9.323841038 2.207240575 1
1.666652748 9.323841037 2.10677658 1
0.8333981009 9.323841038 1.973590648 1
0.2776433959 9.323841038 1.875276116 1
-9.409566e-010 9.323841037 1.826155413 1
10 9.774613679 1.66872048 1
9.722349845 9.774613679 1.717282276 1
9.166605362 9.774613679 1.814478877 1
8.333346378 9.774613679 1.946539711 1
7.499995513 9.774613679 2.047484185 1
6.666670942 9.774613679 2.114958688 1
5.833334738 9.774613679 2.131148253 1
5 9.774613679 2.125028247 1
4.166665262 9.774613679 2.131148253 1
3.333329058 9.774613679 2.114958688 1
2.500004487 9.774613679 2.047484185 1

1.666653622 9.774613679 1.946539711 1
0.833394638 9.774613679 1.814478877 1
0.277650155 9.774613679 1.717282276 1
-6.063248e-010 9.774613679 1.66872048 1
10 10 1.590003014 1
9.722346497 10 1.638285283 1
9.166607078 10 1.734923025 1
8.333345944 10 1.866421266 1
7.499995794 10 1.967605992 1
6.66667088 10 2.035646936 1
5.833334754 10 2.05256843 1
5 10 2.046927932 1
4.166665246 10 2.05256843 1
3.33332912 10 2.035646936 1
2.500004206 10 1.967605992 1
1.666654056 10 1.866421266 1
0.833392922 10 1.734923025 1
0.2776535026 10 1.638285283 1
1.8e-015 10 1.590003014 1

0
0
0
0
0.055556
0.111111
0.166667
0.222222
0.277778
0.333333
0.388889
0.444444
0.5
0.555556
0.611111
0.666667
0.722222
0.777778
0.833333
0.888889
0.944444

1
1
1
1

0
0
0
0
0.083333
0.166667
0.25
0.333333
0.416667
0.5

0.583333
 0.666667
 0.75
 0.833333
 0.916667
 1
 1
 1
 1

 3 3
 20 3

 1.8e-015 10 1 1
 1.8e-015 10 1.196667671 1
 1.8e-015 10 1.393335342 1
 1.8e-015 10 1.590003014 1
 1.75e-015 9.851851852 1 1
 -2.021071e-010 9.826105794 1.222906827 1
 -4.0421595e-010 9.800359737 1.445813654 1
 -6.063248e-010 9.774613679 1.66872048 1
 1.655555556e-015 9.555555556 1 1
 -3.136510963e-010 9.478317383 1.275385138 1
 -6.273038481e-010 9.40107921 1.550770275 1
 -9.409566e-010 9.323841037 1.826155413 1
 1.52962963e-015 9.111111111 1 1
 8.405911975e-011 8.975622536 1.332989708 1
 1.681167099e-010 8.840133961 1.665979416 1
 2.521743e-010 8.704645385 1.998969124 1
 1.418518519e-015 8.666666667 1 1
 -2.249868765e-011 8.536834855 1.324590148 1
 -4.499879383e-011 8.407003044 1.649180296 1
 -6.74989e-011 8.277171232 1.973770444 1
 1.321296296e-015 8.222222222 1 1
 6.015614198e-012 8.103144846 1.300405719 1
 1.20299071e-011 7.98406747 1.600811439 1
 1.80442e-011 7.864990093 1.901217158 1
 1.237037037e-015 7.777777778 1 1
 -1.618608642e-012 7.608072825 1.29256692 1
 -3.238454321e-012 7.438367873 1.58513384 1
 -4.8583e-012 7.26866292 1.87770076 1
 1.164814815e-015 7.333333333 1 1
 2.974432099e-013 7.119544937 1.31044857 1
 5.937216049e-013 6.90575654 1.62089714 1
 8.9e-013 6.691968144 1.93134571 1
 1.103703704e-015 6.888888889 1 1
 -1.274641975e-013 6.696922737 1.399120731 1
 -2.560320988e-013 6.504956586 1.798241462 1
 -3.846e-013 6.312990434 2.197362194 1
 1.052777778e-015 6.444444444 1 1
 4.746851852e-014 6.29941959 1.483630563 1
 9.388425926e-014 6.154394735 1.967261125 1
 1.403e-013 6.00936988 2.450891688 1
 1.011111111e-015 6 1 1
 -9.169259259e-014 5.912864563 1.513997056 1
 -1.843962963e-013 5.825729126 2.027994112 1

-2.771e-013 5.738593689 2.541991168 1
 9.777777778e-016 5.555555556 1 1
 1.581518519e-013 5.516776149 1.460381213 1
 3.153259259e-013 5.477996741 1.920762426 1
 4.725e-013 5.439217334 2.381143639 1
 9.518518519e-016 5.111111111 1 1
 -1.376320988e-013 5.070445061 1.467774442 1
 -2.762160494e-013 5.029779011 1.935548884 1
 -4.148e-013 4.98911296 2.403323326 1
 9.324074074e-016 4.666666667 1 1
 4.920882716e-013 4.620388053 1.429053707 1
 9.832441358e-013 4.574109439 1.858107415 1
 1.4744e-012 4.527830825 2.287161122 1
 9.185185185e-016 4.222222222 1 1
 -1.874620988e-012 4.209767413 1.474880404 1
 -3.750160494e-012 4.197312604 1.949760808 1
 -5.6257e-012 4.184857794 2.424641211 1
 9.092592593e-016 3.777777778 1 1
 6.255439506e-012 3.807939156 1.48607743 1
 1.250996975e-011 3.838100535 1.972154859 1
 1.87645e-011 3.868261913 2.458232289 1
 9.037037037e-016 3.333333333 1 1
 -2.296533086e-011 3.405509171 1.58858022 1
 -4.593156543e-011 3.477685009 2.17716044 1
 -6.88978e-011 3.549860847 2.765740659 1
 9.009259259e-016 2.888888889 1 1
 8.485093395e-011 2.981897579 1.626891616 1
 1.69700967e-010 3.074906269 2.253783232 1
 2.54551e-010 3.167914959 2.880674848 1
 9e-016 2.444444444 1 1
 -3.15237e-010 2.500690073 1.640431304 1
 -6.304749e-010 2.556935702 2.280862608 1
 -9.457128e-010 2.61318133 2.921293912 1
 9e-016 2.148148148 1 1
 -2.032936667e-010 2.166896691 1.638626012 1
 -4.065882333e-010 2.185645234 2.277252025 1
 -6.098828e-010 2.204393777 2.915878037 1
 9e-016 2 1 1
 -9.799333333e-013 2 1.637723366 1
 -1.960766667e-012 2 2.275446733 1
 -2.9416e-012 2 2.913170099 1

0
 0
 0
 0
 0.055556
 0.111111
 0.166667
 0.222222
 0.277778
 0.333333
 0.388889
 0.444444
 0.5
 0.555556

0.611111
 0.666667
 0.722222
 0.777778
 0.833333
 0.888889
 0.944444
 1
 1
 1
 1
 0
 0
 0
 0
 1
 1
 1
 1
 3 3
 20 3
 10 10 1.590003014 1
 10 10 1.393335342 1
 10 10 1.196667671 1
 10 10 1 1
 10 9.774613679 1.66872048 1
 10 9.774613679 1.445813654 1
 10 9.774613679 1.222906827 1
 10 9.774613679 1 1
 10 9.323841037 1.826155413 1
 10 9.323841037 1.550770275 1
 10 9.323841037 1.275385138 1
 10 9.323841037 1 1
 10 8.704645385 1.998969124 1
 10 8.704645385 1.665979416 1
 10 8.704645385 1.332989708 1
 10 8.704645385 1 1
 10 8.277171232 1.973770444 1
 10 8.277171232 1.649180296 1
 10 8.277171232 1.324590148 1
 10 8.277171232 1 1
 10 7.864990093 1.901217158 1
 10 7.864990093 1.600811439 1
 10 7.864990093 1.300405719 1
 10 7.864990093 1 1
 10 7.26866292 1.87770076 1
 10 7.26866292 1.58513384 1
 10 7.26866292 1.29256692 1
 10 7.26866292 1 1
 10 6.691968144 1.93134571 1
 10 6.691968144 1.62089714 1
 10 6.691968144 1.31044857 1
 10 6.691968144 1 1

10 6.312990434 2.197362194 1
10 6.312990434 1.798241462 1
10 6.312990434 1.399120731 1
10 6.312990434 1 1
10 6.00936988 2.450891688 1
10 6.00936988 1.967261125 1
10 6.00936988 1.483630563 1
10 6.00936988 1 1
10 5.738593689 2.541991168 1
10 5.738593689 2.027994112 1
10 5.738593689 1.513997056 1
10 5.738593689 1 1
10 5.439217334 2.381143639 1
10 5.439217334 1.920762426 1
10 5.439217334 1.460381213 1
10 5.439217334 1 1
10 4.98911296 1.530460145 1
10 4.98911296 1.353640097 1
10 4.98911296 1.176820048 1
10 4.98911296 1 1
10 4.527830825 0.885131424 1
10 4.527830825 0.9234209494 1
10 4.527830825 0.9617104747 1
10 4.527830825 1 1
10 4.184857794 1.551778031 1
10 4.184857794 1.36785202 1
10 4.184857794 1.18392601 1
10 4.184857794 1 1
10 3.868261913 2.458232289 1
10 3.868261913 1.972154859 1
10 3.868261913 1.48607743 1
10 3.868261913 1 1
10 3.549860847 2.765740659 1
10 3.549860847 2.17716044 1
10 3.549860847 1.58858022 1
10 3.549860847 1 1
10 3.167914959 2.880674848 1
10 3.167914959 2.253783232 1
10 3.167914959 1.626891616 1
10 3.167914959 1 1
10 2.61318133 2.921293912 1
10 2.61318133 2.280862608 1
10 2.61318133 1.640431304 1
10 2.61318133 1 1
10 2.204393777 2.915878037 1
10 2.204393777 2.277252025 1
10 2.204393777 1.638626012 1
10 2.204393777 1 1
10 2 2.913170099 1
10 2 2.275446733 1
10 2 1.637723366 1
10 2 1 1

0
0
0

0
0.055556
0.111111
0.166667
0.222222
0.277778
0.333333
0.388889
0.444444
0.5
0.555556
0.611111
0.666667
0.722222
0.777778
0.833333
0.888889
0.944444

1
1
1
1

0
0
0
0
1
1
1
1

3 3
14 3

1.8e-015 10 1.590003014 1
1.8e-015 10 1.393335342 1
1.8e-015 10 1.196667671 1
1.8e-015 10 1 1
0.2776535026 10 1.638285283 1
0.2776949277 10 1.425523522 1
0.2777363527 10 1.212761761 1
0.2777777778 10 1 1
0.833392922 10 1.734923025 1
0.8333730591 10 1.489948683 1
0.8333531962 10 1.244974342 1
0.8333333333 10 1 1
1.666654056 10 1.866421266 1
1.666658259 10 1.577614178 1
1.666662463 10 1.288807089 1
1.666666667 10 1 1
2.500004206 10 1.967605992 1
2.500002804 10 1.645070661 1
2.500001402 10 1.322535331 1

2.5 10 1 1
 3.33332912 10 2.035646936 1
 3.333330525 10 1.690431291 1
 3.333331929 10 1.345215645 1
 3.333333333 10 1 1
 4.166665246 10 2.05256843 1
 4.16666572 10 1.701712287 1
 4.166666193 10 1.350856143 1
 4.166666667 10 1 1
 5 10 2.046927932 1
 5 10 1.697951955 1
 5 10 1.348975977 1
 5 10 1 1
 5.833334754 10 2.05256843 1
 5.83333428 10 1.701712287 1
 5.833333807 10 1.350856143 1
 5.833333333 10 1 1
 6.66667088 10 2.035646936 1
 6.666669475 10 1.690431291 1
 6.666668071 10 1.345215645 1
 6.666666667 10 1 1
 7.499995794 10 1.967605992 1
 7.499997196 10 1.645070661 1
 7.499998598 10 1.322535331 1
 7.5 10 1 1
 8.333345944 10 1.866421266 1
 8.333341741 10 1.577614178 1
 8.333337537 10 1.288807089 1
 8.333333333 10 1 1
 9.166607078 10 1.734923025 1
 9.166626941 10 1.489948683 1
 9.166646804 10 1.244974342 1
 9.166666667 10 1 1
 9.722346497 10 1.638285283 1
 9.722305072 10 1.425523522 1
 9.722263647 10 1.212761761 1
 9.722222222 10 1 1
 10 10 1.590003014 1
 10 10 1.393335342 1
 10 10 1.196667671 1
 10 10 1 1

0
 0
 0
 0
 0.083333
 0.166667
 0.25
 0.333333
 0.416667
 0.5
 0.583333
 0.666667
 0.75
 0.833333

0.916667

1
1
1
1

0
0
0
0
1
1
1
1

3 3
14 3

1.8e-015 10 1 1
9e-016 7.333333333 1 1
9e-016 4.666666667 1 1
9e-016 2 1 1
0.277777778 10 1 1
0.277777778 7.333333333 1 1
0.277777778 4.666666667 1 1
0.277777778 2 1 1
0.833333333 10 1 1
0.833333333 7.333333333 1 1
0.833333333 4.666666667 1 1
0.833333333 2 1 1
1.666666667 10 1 1
1.666666667 7.333333333 1 1
1.666666667 4.666666667 1 1
1.666666667 2 1 1
2.5 10 1 1
2.5 7.333333333 1 1
2.5 4.666666667 1 1
2.5 2 1 1
3.333333333 10 1 1
3.333333333 7.333333333 1 1
3.333333333 4.666666667 1 1
3.333333333 2 1 1
4.166666667 10 1 1
4.166666667 7.333333333 1 1
4.166666667 4.666666667 1 1
4.166666667 2 1 1
5 10 1 1
5 7.333333333 1 1
5 4.666666667 1 1
5 2 1 1
5.833333333 10 1 1
5.833333333 7.333333333 1 1
5.833333333 4.666666667 1 1
5.833333333 2 1 1
6.666666667 10 1 1
6.666666667 7.333333333 1 1

6.66666667 4.66666667 1 1
 6.66666667 2 1 1
 7.5 10 1 1
 7.5 7.33333333 1 1
 7.5 4.66666667 1 1
 7.5 2 1 1
 8.33333333 10 1 1
 8.33333333 7.33333333 1 1
 8.33333333 4.66666667 1 1
 8.33333333 2 1 1
 9.16666667 10 1 1
 9.16666667 7.33333333 1 1
 9.16666667 4.66666667 1 1
 9.16666667 2 1 1
 9.72222222 10 1 1
 9.72222222 7.33333333 1 1
 9.72222222 4.66666667 1 1
 9.72222222 2 1 1
 10 10 1 1
 10 7.33333333 1 1
 10 4.66666667 1 1
 10 2 1 1

 0
 0
 0
 0
 0.083333
 0.166667
 0.25
 0.333333
 0.416667
 0.5
 0.583333
 0.666667
 0.75
 0.833333
 0.916667
 1
 1
 1
 1

 0
 0
 0
 0
 1
 1
 1
 1

 3 3
 14 3

 10 2 2.913170099 1

10 2 2.275446733 1
10 2 1.637723366 1
10 2 1 1
9.722222222 2 2.913170099 1
9.722222222 2 2.275446733 1
9.722222222 2 1.637723366 1
9.722222222 2 1 1
9.166666667 2 2.913170099 1
9.166666667 2 2.275446733 1
9.166666667 2 1.637723366 1
9.166666667 2 1 1
8.333333333 2 2.913170099 1
8.333333333 2 2.275446733 1
8.333333333 2 1.637723366 1
8.333333333 2 1 1
7.5 2 2.913170099 1
7.5 2 2.275446733 1
7.5 2 1.637723366 1
7.5 2 1 1
6.666666667 2 2.913170099 1
6.666666667 2 2.275446733 1
6.666666667 2 1.637723366 1
6.666666667 2 1 1
5.833333333 2 2.913170099 1
5.833333333 2 2.275446733 1
5.833333333 2 1.637723366 1
5.833333333 2 1 1
5 2 2.913170099 1
5 2 2.275446733 1
5 2 1.637723366 1
5 2 1 1
4.166666667 2 2.913170099 1
4.166666667 2 2.275446733 1
4.166666667 2 1.637723366 1
4.166666667 2 1 1
3.333333333 2 2.913170099 1
3.333333333 2 2.275446733 1
3.333333333 2 1.637723366 1
3.333333333 2 1 1
2.5 2 2.913170099 1
2.5 2 2.275446733 1
2.5 2 1.637723366 1
2.5 2 1 1
1.666666667 2 2.913170099 1
1.666666667 2 2.275446733 1
1.666666667 2 1.637723366 1
1.666666667 2 1 1
0.833333333 2 2.913170099 1
0.833333333 2 2.275446733 1
0.833333333 2 1.637723366 1
0.833333333 2 1 1
0.277777778 2 2.913170099 1
0.277777778 2 2.275446733 1
0.277777778 2 1.637723366 1
0.277777778 2 1 1
-2.9416e-012 2 2.913170099 1

-1.9611e-012 2 2.275446733 1
 -9.805e-013 2 1.637723366 1
 9e-016 2 1 1

 0
 0
 0
 0
 0.083333
 0.166667
 0.25
 0.333333
 0.416667
 0.5
 0.583333
 0.666667
 0.75
 0.833333
 0.916667
 1
 1
 1
 1

 0
 0
 0
 0
 1
 1
 1
 1
 3 3
 8 3

 10 5 7 1
 6.666666667 5 7 1
 3.333333333 5 7 1
 0 5 7 1
 10 4.580728757 7.003353363 1
 6.666666667 4.580728757 7.003353363 1
 3.333333333 4.580728757 7.003353363 1
 -1.8e-015 4.580728757 7.003353363 1
 10 3.712954569 6.844516785 1
 6.666666667 3.712954569 6.844516785 1
 3.333333333 3.712954569 6.844516785 1
 1.8e-015 3.712954569 6.844516785 1
 10 2.62092373 6.223420308 1
 6.666666667 2.62092373 6.223420308 1
 3.333333333 2.62092373 6.223420308 1
 -1.24e-014 2.62092373 6.223420308 1
 10 2.918061909 4.509191523 1
 6.666666667 2.918061909 4.509191523 1
 3.333333333 2.918061909 4.509191523 1
 2.32e-014 2.918061909 4.509191523 1
 10 1.231783584 3.950119887 1

6.66666667 1.231783584 3.950119887 1
 3.333333333 1.231783584 3.950119887 1
 -7.1e-015 1.231783584 3.950119887 1
 10 1.003115699 2.56030384 1
 6.66666667 1.003115699 2.56030384 1
 3.333333333 1.003115699 2.56030384 1
 1.07e-014 1.003115699 2.56030384 1
 10 0.9985505318 1.542444275 1
 6.66666667 0.9985505318 1.542444275 1
 3.333333333 0.9985505318 1.542444275 1
 -3.6e-015 0.9985505318 1.542444275 1
 10 1 1 1
 6.66666667 1 1 1
 3.333333333 1 1 1
 0 1 1 1

0
 0
 0
 0
 0.166667
 0.333333
 0.5
 0.666667
 0.833333

1
 1
 1
 1

0
 0
 0
 0
 1
 1
 1
 1

3 3
 3 3

10 5 8 1
 6.66666667 5 8 1
 3.333333333 5 8 1
 0 5 8 1
 10 5 7.66666667 1
 6.66666667 5 7.66666667 1
 3.333333333 5 7.66666667 1
 0 5 7.66666667 1
 10 5 7.333333333 1
 6.66666667 5 7.333333333 1
 3.333333333 5 7.333333333 1
 0 5 7.333333333 1
 10 5 7 1
 6.66666667 5 7 1

3.333333333 5 7 1
0 5 7 1

0
0
0
0
1
1
1
1
1

0
0
0
0
1
1
1
1

3 3
8 3

10 0 1 1
6.666666667 0 1 1
3.333333333 0 1 1
0 0 1 1
10 0 1.54036882 1
6.666666667 0 1.54036882 1
3.333333333 0 1.54036882 1
0 0 1.54036882 1
10 -0.001364444769 2.618820312 1
6.666666667 -0.001364444769 2.618820312 1
3.333333333 -0.001364444769 2.618820312 1
-1.8e-015 -0.001364444769 2.618820312 1
10 0.2621222832 4.229131229 1
6.666666667 0.2621222832 4.229131229 1
3.333333333 0.2621222832 4.229131229 1
-3.6e-015 0.2621222832 4.229131229 1
10 0.908387104 5.724129758 1
6.666666667 0.908387104 5.724129758 1
3.333333333 0.908387104 5.724129758 1
1.78e-014 0.908387104 5.724129758 1
10 1.953308639 6.995845805 1
6.666666667 1.953308639 6.995845805 1
3.333333333 1.953308639 6.995845805 1
-2.66e-014 1.953308639 6.995845805 1
10 3.375937522 7.812003569 1
6.666666667 3.375937522 7.812003569 1
3.333333333 3.375937522 7.812003569 1
1.07e-014 3.375937522 7.812003569 1
10 4.462676433 8 1
6.666666667 4.462676433 8 1
3.333333333 4.462676433 8 1
0 4.462676433 8 1

10 5 8 1
6.666666667 5 8 1
3.333333333 5 8 1
0 5 8 1

0
0
0
0
0.166667
0.333333
0.5
0.666667
0.833333

1
1
1
1

0
0
0
0
1
1
1
1

3 3
3 3

0 -1e-016 0 1
0 -1e-016 0.333333333 1
0 -5.551115123e-017 0.666666667 1
0 0 1 1
3.333333333 -1e-016 0 1
3.333333333 -1e-016 0.333333333 1
3.333333333 -5.551115123e-017 0.666666667 1
3.333333333 0 1 1
6.666666667 -1e-016 0 1
6.666666667 -1e-016 0.333333333 1
6.666666667 -5.551115123e-017 0.666666667 1
6.666666667 0 1 1
10 -1e-016 0 1
10 -1e-016 0.333333333 1
10 -5.551115123e-017 0.666666667 1
10 0 1 1

0
0
0
0
1
1
1
1

0
0
0
0
1
1
1
1
1

3 3
3 3

-9e-016 0 0 1
3.333333333 0 0 1
6.666666667 0 0 1
10 0 0 1
-9e-016 3.333333333 0 1
3.333333333 3.333333333 0 1
6.666666667 3.333333333 0 1
10 3.333333333 0 1
0 6.666666667 0 1
3.333333333 6.666666667 0 1
6.666666667 6.666666667 0 1
10 6.666666667 0 1
0 10 0 1
3.333333333 10 0 1
6.666666667 10 0 1
10 10 0 1

0
0
0
0
1
1
1
1
1

0
0
0
0
1
1
1
1

3 3
3 3

0 10 0 1
3.333333333 10 0 1
6.666666667 10 0 1
10 10 0 1
0 10 0.333333333 1

3.33333333 10 0.333333333 1
 6.66666667 10 0.333333333 1
 10 10 0.333333333 1
 0 10 0.66666667 1
 3.33333333 10 0.66666667 1
 6.66666667 10 0.66666667 1
 10 10 0.66666667 1
 0 10 1 1
 3.33333333 10 1 1
 6.66666667 10 1 1
 10 10 1 1

 0
 0
 0
 0
 1
 1
 1
 1

 0
 0
 0
 0
 1
 1
 1
 1

 3 3
 3 3

 0 10 1 1
 3.33333333 10 1 1
 6.66666667 10 1 1
 10 10 1 1
 0 7 1 1
 3.33333333 7 1 1
 6.66666667 7 1 1
 10 7 1 1
 -9e-016 4 1 1
 3.33333333 4 1 1
 6.66666667 4 1 1
 10 4 1 1
 -9e-016 1.000261351 0.9990284962 1
 3.33333333 1.000261351 0.9990284962 1
 6.66666667 1.000261351 0.9990284962 1
 10 1.000261351 0.9990284962 1

 0
 0
 0
 0
 1
 1

1
 1

 0
 0
 0
 0
 1
 1
 1
 1

 3 3
 8 3

 10 5 8 1
 10 5 7.666666667 1
 10 5 7.333333333 1
 10 5 7 1
 10 4.462676433 8 1
 10 4.502027208 7.667784454 1
 10 4.541377982 7.335568909 1
 10 4.580728757 7.003353363 1
 10 3.375937522 7.812003569 1
 10 3.488276538 7.489507974 1
 10 3.600615553 7.167012379 1
 10 3.712954569 6.844516785 1
 10 1.953308639 6.995845805 1
 10 2.175847002 6.738370639 1
 10 2.398385366 6.480895473 1
 10 2.62092373 6.223420308 1
 10 0.908387104 5.724129758 1
 10 1.578278706 5.319150346 1
 10 2.248170307 4.914170935 1
 10 2.918061909 4.509191523 1
 10 0.2621222832 4.229131229 1
 10 0.5853427168 4.136127448 1
 10 0.9085631505 4.043123668 1
 10 1.231783584 3.950119887 1
 10 -0.001364444769 2.618820312 1
 10 0.3334622697 2.599314822 1
 10 0.6682889841 2.579809331 1
 10 1.003115699 2.56030384 1
 10 0 1.54036882 1
 10 0.3328501773 1.541060638 1
 10 0.6657003545 1.541752457 1
 10 0.9985505318 1.542444275 1
 10 0 1 1
 10 0.3333333333 1 1
 10 0.6666666667 1 1
 10 1 1 1

 0
 0
 0
 0

0.166667
 0.333333
 0.5
 0.666667
 0.833333
 1
 1
 1
 1

 0
 0
 0
 0
 1
 1
 1
 1

 3 3
 8 3

 0 0 1 1
 0 0.3333333333 1 1
 0 0.6666666667 1 1
 0 1 1 1
 0 0 1.54036882 1
 -1.2e-015 0.3328501773 1.541060638 1
 -2.4e-015 0.6657003545 1.541752457 1
 -3.6e-015 0.9985505318 1.542444275 1
 -1.8e-015 -0.001364444769 2.618820312 1
 2.4e-015 0.3334622697 2.599314822 1
 6.5e-015 0.6682889841 2.579809331 1
 1.07e-014 1.003115699 2.56030384 1
 -3.6e-015 0.2621222832 4.229131229 1
 -4.8e-015 0.5853427168 4.136127448 1
 -5.9e-015 0.9085631505 4.043123668 1
 -7.1e-015 1.231783584 3.950119887 1
 1.78e-014 0.908387104 5.724129758 1
 1.96e-014 1.578278706 5.319150346 1
 2.14e-014 2.248170307 4.914170935 1
 2.32e-014 2.918061909 4.509191523 1
 -2.66e-014 1.953308639 6.995845805 1
 -2.19e-014 2.175847002 6.738370639 1
 -1.71e-014 2.398385366 6.480895473 1
 -1.24e-014 2.62092373 6.223420308 1
 1.07e-014 3.375937522 7.812003569 1
 7.7e-015 3.488276538 7.489507974 1
 4.8e-015 3.600615553 7.167012379 1
 1.8e-015 3.712954569 6.844516785 1
 0 4.462676433 8 1
 -6e-016 4.502027208 7.667784454 1
 -1.2e-015 4.541377982 7.335568909 1
 -1.8e-015 4.580728757 7.003353363 1
 0 5 8 1

0 5 7.666666667 1
 0 5 7.333333333 1
 0 5 7 1

0
 0
 0
 0
 0.166667
 0.333333
 0.5
 0.666667
 0.833333

1
 1
 1
 1

0
 0
 0
 0
 1
 1
 1
 1

3 3
 3 3

10 10 0 1
 10 6.666666667 0 1
 10 3.333333333 0 1
 10 1e-016 0 1
 10 10 0.333333333 1
 10 6.666666667 0.333333333 1
 10 3.333333333 0.333333333 1
 10 1e-016 0.333333333 1
 10 10 0.666666667 1
 10 6.666666667 0.666666667 1
 10 3.333333333 0.666666667 1
 10 5.551115123e-017 0.666666667 1
 10 10 1 1
 10 6.666666667 1 1
 10 3.333333333 1 1
 10 0 1 1

0
 0
 0
 0
 1
 1
 1
 1

0
0
0
0
1
1
1
1
1

3 3
3 3

0 10 1 1
0 6.666666667 1 1
-9e-016 3.333333333 1 1
-9e-016 0 1 1
0 10 0.666666667 1
0 6.666666667 0.666666667 1
-9e-016 3.333333333 0.666666667 1
-9e-016 0 0.666666667 1
0 10 0.333333333 1
0 6.666666667 0.333333333 1
-9e-016 3.333333333 0.333333333 1
-9e-016 0 0.333333333 1
0 10 0 1
0 6.666666667 0 1
-9e-016 3.333333333 0 1
-9e-016 0 0 1

0
0
0
0
1
1
1
1
1

0
0
0
0
1
1
1
1
1

APPENDIX B

PART OF CL DATA FOR MULTI-CUTTER MACHINING IN VERICUT

```

SPINDL/4000
FEDRAT/750
COOLNT/ON
RAPID
CUTTER/8      0.5      3.5      0.5      0      0      100
FROM/0        50      170      0      0      1
RAPID

RAPID
GOTO/198.762  20.6276 170      0      0      1
RAPID
GOTO/198.762  20.6276 47.1367 0.997532      0.000901519      0.0702125
RAPID
GOTO/-0.75444720.4473 33.1042 0.997532      0.000901519      0.0702125
SPINDLE/ON
GOTO/-0.74444720.4473 33.0942 0.997532      0.000901519      0.0702125
GOTO/3.08125  20.3695 33.0501 0.98886 0.0841291      0.122789
GOTO/7.08229  20.3702 33.0504 0.988948      0.0835719      0.122464
GOTO/11.0823  20.3702 33.0504 0.988948      0.0835693      0.122466
GOTO/15.2565  20.4476 33.0944 0.997553      0.000492099      0.0699169
GOTO/19.2543  20.4468 33.0939 0.997505      0.00149633      0.0705817
GOTO/23.0794  20.3686 33.0495 0.98872 0.0849424      0.123353
GOTO/27.0823  20.3702 33.0504 0.988949      0.0835497      0.122463
GOTO/31.0823  20.3702 33.0504 0.98895 0.0835553      0.122454
GOTO/35.0794  20.3686 33.0495 0.98872 0.0849554      0.123352
GOTO/39.2566  20.4476 33.0944 0.997554      0.000483512      0.0699044
GOTO/43.0823  20.3702 33.0504 0.988949      0.083574      0.122457
GOTO/47.0823  20.3702 33.0504 0.988949      0.0835738      0.122455
GOTO/51.2564  20.4476 33.0944 0.997551      0.000491769      0.0699454
GOTO/55.2297  20.4476 33.0885 0.99698 0.000390461      0.0776597
GOTO/59.0198  20.365  33.0328 0.986553      0.0831291      0.140719
GOTO/63.1625  20.4489 33.0729 0.995275      -0.00125282      0.0970828
GOTO/67.1122  20.4478 33.0602 0.993741      -4.00142e-005      0.11171
GOTO/71.0552  20.4492 33.0449 0.991739      -0.00163304      0.128261
GOTO/74.9772  20.4495 33.0223 0.988534      -0.00193772      0.150984
GOTO/78.8732  20.4503 32.9891 0.983401      -0.00267079      0.181428
GOTO/82.7255  20.4506 32.9358 0.974374      -0.00290325      0.224914
GOTO/86.5086  20.4518 32.8435 0.957216      -0.00373716      0.289351
GOTO/90.1717  20.4542 32.6641 0.920447      -0.00505622      0.390835

```

GOTO/93.6456	20.4592	32.2756	0.832697		-0.00716029	0.553682
GOTO/96.9909	20.4692	31.4942	0.640087		-0.00994313	0.768238
GOTO/200	20.4692	31.4942	0.640087		-0.00994313	0.768238
SPINDLE/OFF						
RAPID						
GOTO/196.991	20.4692	170	0	0	1	
RAPID						
GOTO/196.036	39.0702	170	0	0	1	
RAPID						
GOTO/196.036	39.0702	62.7868	0.985222		0.0849738	0.148716
RAPID						
GOTO/-1.01869	22.0754	33.0537	0.985222		0.0849738	0.148716
SPINDLE/ON						
GOTO/-1.00869	22.0754	33.0437	0.985222		0.0849738	0.148716
GOTO/2.99133	22.0755	33.0437	0.985225		0.0849596	0.148706
GOTO/7.18163	22.1632	33.0965	0.9958	-0.000731412	0.0915554	
GOTO/10.9914	22.0755	33.0437	0.985228		0.0849406	0.148697
.....						
GOTO/84.0879	99.536	18.7584	0.162373		0.329928	0.929937
GOTO/88.0465	99.5381	18.1557	0.171726		0.328406	0.928795
GOTO/91.9967	99.5338	17.5102	0.177179		0.32685	0.928319
GOTO/95.9399	99.5321	16.8361	0.179393		0.325328	0.928429
GOTO/96.5532	99.5291	16.7305	0.179452		0.325116	0.928492
GOTO/200	99.5291	16.7305	0.179452		0.325116	0.928492
SPINDLE/OFF						
RAPID						
GOTO/196.553	99.5291	170	0	0	1	
COOLNT/OFF						
RAPID						
GOTO/050	170	0	0	1		
FINI						
SPINDL/4000						
FEDRAT/750						
COOLNT/ON						
RAPID						
CUTTER/2	0.2	0.8	0.2	0	0	100
FROM/0	50	170	0	0	1	
RAPID						
RAPID						
GOTO/125.829	80.3238	170	0	0	1	
RAPID						
GOTO/125.829	80.3238	174.661	0.632313		0.199663	0.748542
RAPID						
GOTO/-0.64394440.3913	24.9628	0.632313			0.199663	0.748542
SPINDLE/ON						
GOTO/-0.63394440.3913	24.9528	0.632313			0.199663	0.748542
GOTO/0.228678	40.4472	24.4723	0.659842		0.399445	0.636437
GOTO/1.05905	40.3824	23.9194	0.541316		0.569718	0.618384

GOTO/2.34112	39.9877	23.5919	0.40097	0.750521	0.525302
GOTO/3.33629	39.969	23.1282	0.319485	0.807939	0.495141
GOTO/4.27853	39.9868	22.694	0.263601	0.846502	0.462546
GOTO/5.2989	39.9921	22.2756	0.211068	0.874349	0.436995
GOTO/6.26041	40.0135	21.8989	0.172269	0.894106	0.413399
GOTO/7.04935	40.0201	21.6135	0.146525	0.900162	0.41017
GOTO/7.57142	41.2232	21.0765	0.980888	0.164397	-0.104078
GOTO/8.47831	41.2233	20.7864	0.979903	0.195055	-0.0417468
GOTO/9.4968	41.245	20.5855	0.982056	0.154362	-0.10834
GOTO/10.4364	41.2405	20.4067	0.98175	0.182719	-0.0527285
GOTO/11.3529	41.2162	20.2686	0.967271	0.253226	0.0162113
GOTO/12.2163	41.1416	20.1523	0.907474	0.405391	0.110221
GOTO/13.2241	41.1565	20.0949	0.916045	0.381909	0.122503
GOTO/14.0637	41.0104	20.003	0.77365	0.592066	0.225661
GOTO/15.0652	41.016	19.9885	0.776881	0.585976	0.230411
GOTO/15.975	40.877	19.9279	0.625906	0.724493	0.288708
GOTO/16.9434	40.8036	19.8973	0.542086	0.780628	0.311068
GOTO/17.9213	40.7326	19.868	0.45841	0.82563	0.328932
GOTO/18.9108	40.6843	19.8479	0.400207	0.851355	0.33916
GOTO/19.9071	40.662	19.8382	0.372865	0.861982	0.34345
GOTO/20.9072	40.6625	19.8377	0.373238	0.861788	0.343533
GOTO/21.9067	40.6597	19.8361	0.36962	0.863084	0.34419

.....