Founded 1905

PROCESS PLANNING FOR FIVE-AXIS MILLING OF SCULPTURED SURFACES

LI LINGLING

(B.Eng., M.Eng.)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGPAORE

2007

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my supervisor, A/Prof. Zhang Yunfeng, from the Department of Mechanical Engineering at the 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 a research scholarship from the National University of Singapore, and for the support by ASTAR of Singapore under the project R265-000-176-305.

Special thanks are given to A/Prof Fuh Ying Hsi, A/Prof. Wong Yoke San and A/P A Senthil Kumar for their suggestion of this research. And I also wish to thank all my fellow graduate students for their support and encouragement, and a pleasant research environment.

Finally, I thank my parents and husband for their kindness and love. Without their deep love and constant support, I cannot smoothly complete the project.

TABLE OF CONTENTS

SUMMARY

This thesis studies the automated process planning for finish cut of sculptured surfaces using a 5-axis milling machine. The objective is to automatically carry out the process planning tasks, including cutter selection and tool-path generation, in an integrated and efficient way based on the concept of cutter accessibility.

Firstly, a unique algorithm is developed to evaluate the accessibility of a cylindrical fillet-end cutter to a point on a surface by considering machine axis limits, avoidance of local-gouging, rear-gouging, and global-collision. The accessibility map (A-map) is formed through geometric analysis and represented in terms of ranges of tilting and rotational angles. The checking of cutter accessibility is performed with respect to all possible directions instead of a fixed feeding direction, which is adopted by most other approaches in the literature.

Secondly, an intelligent method is developed to efficiently select the optimal cutter from the available ones with respect to cutting efficiency, by checking cutter's accessibility to the sampled points on a given part surface. Two techniques are presented to alleviate the extensive computation load for cutter selection. The first is surface decomposition, which divides the surface into interference-prone regions and interference-free regions based on the geometry of both cutter and part surfaces. The accessibility checking is carried out only within the interference-prone regions. The second is accessibility comparison between cutters, which can reduce the redundancy when the search procedure is applied from a larger cutter to a smaller one. Moreover,

the checking results in cutter selection can be subsequently employed in tool-path generation tasks.

Thirdly, efficient algorithms are developed for the tasks of tool-path generation, including determining the path direction and generating the cutter location (CL) data. They are based on the A-map at each sampled surface point, obtained in cutter selection. To begin with, the optimal path direction is identified by an optimization approach aiming at minimizing cutter posture change rate during the machining of the whole surface. In addition, the A-maps are also utilized to obtain the optimal tool paths with respect to the largest cutting strip. An interpolation approach is proposed to obtain the cutter postures thus reducing the computation load significantly.

Finally, computer implementation and illustrative examples are performed to demonstrate the validity, efficacy and robustness of the developed methods.

LIST OF TABLES

LIST OF FIGURES

LIST OF GLOSSARY

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

R: Cutter major radius

r_f : Cutter minor radius

L: Cutter length

k: Total number of discrete sampled θs over the rotational angle range $[\theta_{\min}, \theta_{\max}]$

m: the number of sampled points on the surface for interference checking

n: the number of checked points for a cutter's accessibility to a surface

$\boldsymbol{f}$: feeding direction

($\boldsymbol{X}_L$, $\boldsymbol{Y}_L$, $\boldsymbol{Z}_L$): The local coordinate frame at a surface point

($\boldsymbol{X}_T$, $\boldsymbol{Y}_T$, $\boldsymbol{Z}_T$): The tool coordinate frame at a surface point

A-map: Accessibility map (accessible range) of a cutter to a surface point

APR: Accessible posture range

CA algorithm: Cutter accessibility algorithm

CAM: Computer-Aided Machining

CC point: Cutter contact point to the surface

CL: Cutter location

CMM: Coordinate measuring machine

DOF: Degree of freedom

GC: Global-collision

LG: Local-gouging

ML:	Machine axis limits
NC machine:	Numerically controlled machine
NURBS:	Non-Uniform Rational B-Spline
PCR:	Posture change rate
RG:	Rear-gouging

CHAPTER 1

INTRODUCTION

Sculptured surface machining has been a major contributor in the process of bringing new products to the market place. A great variety of industries, from ship building to aerospace, all rely on this technology for producing dies and molds used in manufacturing the part components. The use of numerically controlled (NC) machines for tooling making has become a vital part of the product development process.

This chapter introduces the technology of 5-aixs NC milling in sculptured surface machining as well as automated process planning, one of the critical challenges for successful 5-axis cutting. Further, based on the discussion of the-state-of-art in commercial systems and published work, the motivation of this thesis is presented and followed by the detailed description of the research scope.

1.1 Sculptured Surfaces

Sculptured surfaces, also called freeform surfaces, are commonly employed in product design to enhance the aesthetic appeal or meet functional demands for complex elements in industry. Irregular curvature distribution, one of the main features for sculptured surfaces, contributes to the difficulty of direct machining from the design concepts to the surfaces. Thus, the original design concepts of sculptured surfaces are generally embodied in a master model, sculptured by the skilled hands of an artisan in an easily workable material like clay or wood. The master model is then stored as “database” for mass-producing of the product, by the use of a tracing mill

where the master model is traced by a stylus while a cutting tool machines a duplicate in the steel mould. The arrival of the computer revolution in the 1960s radically changed this process. The master model can be stored in a digital form through data capturing by a coordinate measuring machine (CMM). The data is then fitted to the surface with a strictly mathematical expression, generally in parametrical form such as Coons, Bezier, B-spline, and recently NURBS. Further, there is now a trend toward eliminating the clay and wood master model in favor of the virtual creative space in Computer-Aided Machining (CAM). With the increasing application of sculptured surfaces, the machining of sculptured surfaces has become one of the critical issues in the process of new products.

In the 1950s, the increased need for precision-machining of aircraft parts led to the development of NC milling machines. Nowadays, sculptured surfaces are often produced on a NC machine due to its superior accuracy, efficiency and ability to operating a much broader range of materials (Beaman *et al.*, 1997). In general, three metal removal stages are required to generate the final shape of a finished-part from a raw stock (Warkentin *et al*., 1996):

(1) Roughing: to remove the bulk waste material and sculpt the surface shape,

(2) Finishing: to produce the final shape of the finished part (with some cusps),

(3) Polishing or grinding: to remove the cusp left and keep the surface errors within the desirable tolerance.

As much as 25% to 38% of the total machining time is spent on hand polishing which aims to eliminate the cusp left in the finishing stage (Manson, 1995). Thus, the efficiency and accuracy in finish cutting is a critical issue to the efficient manufacturing of sculptured surfaces with NC machines.

1.2 Five-axis NC Milling

To achieve successful metal cutting, several important criteria need to be followed in the finishing stage (Li and Jerard, 1994):

(1) Accuracy: the shape errors introduced by machining should be bounded, and machined surfaces must be interference-free.

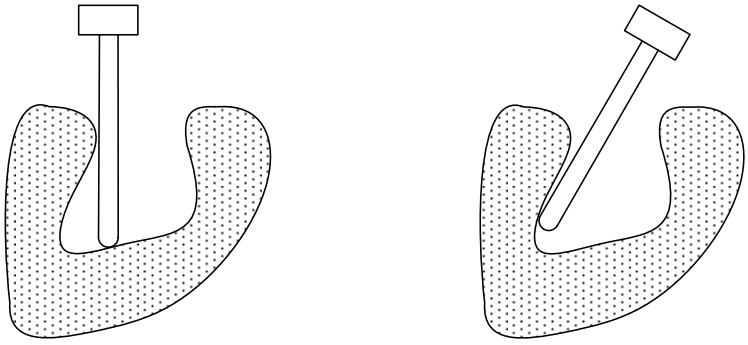
(2) Efficiency: three important measures need to be considered: (a) increased programmer productivity with resultant speedup in the product-development process, (b) algorithm efficiency for machining data generation, and (c) machining time to produce the finished part.

(3) Robustness: tool-paths should not be constrained to only a specific topology or a specific direction. Also, a robust system should be versatile in selection of cutter sizes and shapes and in tool-path planning.

Traditionally, 3-axis NC machines with ball-end tools are employed for sculptured surface machining. In a 3-axis machine, tools can move with a fixed axis direction to any point in its workspace with three translational freedoms. It is somewhat easy to position tools relative to the machined surface and to generate simple codes because of the relatively simple tool movement without revolution. However, with the growing requirement of complex components in industry, the whole process with 3-axis mode becomes inefficient and the resultant surface finish inaccurate (Vickers and Quan, 1989). To meet the demand for better accuracy and efficiency, 5-axis machining with flat-end or fillet-end tools have been introduced for sculptured surface machining.

In 5-axis machining, the machine can not only move a tool to any point in its workspace, but also be able to position it in any arbitrary orientation relative to the surface with two additional rotational degrees of freedom (DOFs). In theory, 5-axis

machining provides many advantages over 3-axis mode. Firstly, with two rotational DOFs, 5-axis machining provides more flexibility to deal with geometrically and topologically complex surfaces. As shown in Figure 1.1a, a tool in a 3-axis machine is positioned in a fixed direction during one setup. This means that only those parts of a model that are visible from a particular direction can be milled. Inaccessible regions have to be machined by configuring the cutter setup again along other directions. However, with two rotational DOFs, a cutter in a 5-axis machine, shown in Figure 1.1b, can adjust its angles during one setup to reach into the areas that are not accessible by a cutter in a 3-axis machine. The preparatory work is reduced and so is the machining time and cost for finishing a sculptured surface. Meanwhile, the machining accuracy is increased from fewer position reconfiguration set-up processes.



(a) 3-axis machining (b) 5-axis machining

Figure 1.1 Comparison of the accessible regions between 3-axis and 5-axis milling

Secondly, 5-axis machining can also improve the finished surface quality and increase material removal rate due to the close match of the tool cutting edge to the surface shape. In 3-axis milling, cutter axis direction is fixed and the ball-end tool geometry is unchangeable with respect to the changed surface features during the whole machining process, resulting in large scallops left after finishing. However,

compared to 3-axis milling, the tool orientation in 5-axis mode can be adjusted to fit the required surface geometry, leading to much smaller scallop and less hand polishing work. Some related works have shown the effectiveness and efficiency of 5-axis finish machining (Vickers and Quan, 1989; Mullins *et al.*, 1993; Li and Jerard, 1994). Pi *et al*. (1998) proposed a new method of grind-free finish machining for a 5-axis machine. The resultant surface has scallops that are within the design surface profile and has no need of hand polishing. Therefore, the machining time in 5-axis machining from a stock material to the finished part is greatly shortened (Gray *et al.*, 2001). In summary, fewer set-ups, faster material-removal rate, and improved surface finish can be achieved in 5-axis machining in theory.

In practice, however, automated ***process planning*** has been the main bottleneck preventing the wide application of 5-axis milling machines in sculptured surface machining. With two additional rotational DOFs than 3-axis mode, tool orientations on a 5-axis machine have to be specified during the whole machining process, leading to intensive computational time in process planning. In addition, the tool is prone to interfere with the non-machined surface portions since both translational and rotational movement results in very complex tool trajectory and swept volume. Thus, cumbersome and complicated algorithms are required to detect and correct the interference and to ensure successful machining. They also contribute to computation time-cost for the preparation of machining data. To summarize, the process planning is complicated and time-consuming in 5-axis sculptured surface machining. There is a need for faster planning techniques to improve both the computation and machining efficiency.

1.3 Process Planning for Sculptured Surface Machining

During process planning, various geometric (e.g., the cutter size and shape, the tool-path topology and distribution) and non-geometric (e.g., dynamitic) parameters should be considered for successful machining. This work only investigates the geometric aspects in the process planning for sculptured surface machining on a 5-axis NC machine.

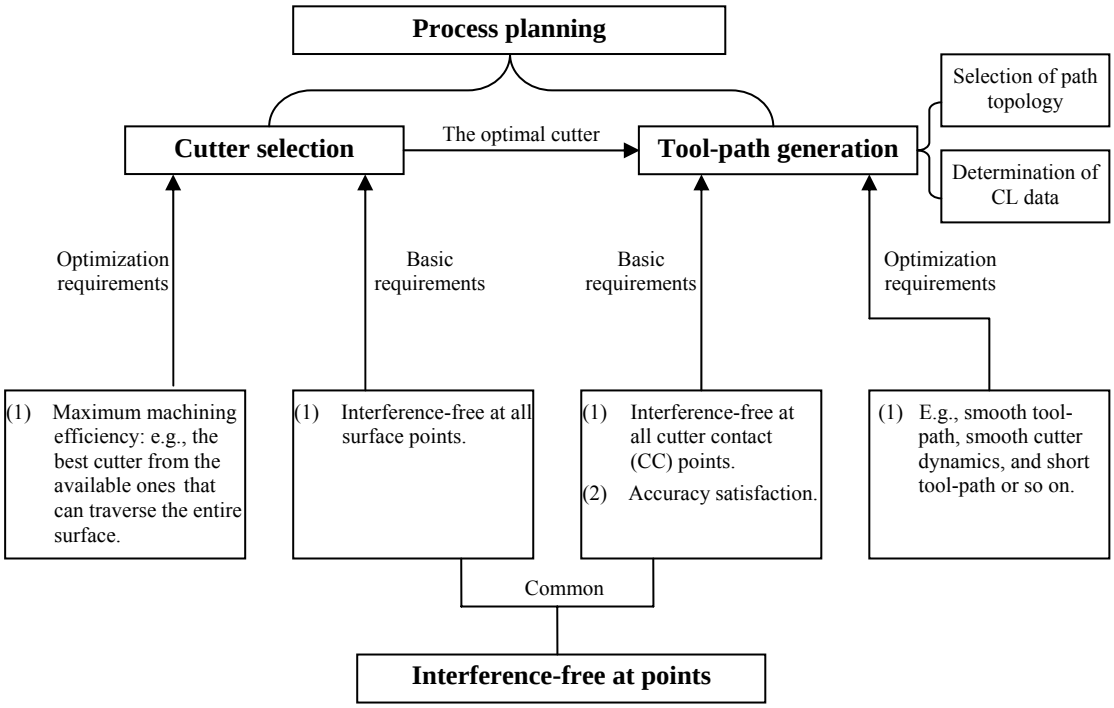


Figure 1.2 Process planning in 5-axis NC milling

The process planning tasks for 5-axis machining (finish cut) include ***cutter selection*** and ***tool-path generation***, as shown in Figure 1.2. The former determines the best cutter from the available ones that can traverse the entire surface without interference. The optimization criterion usually refers to that the largest feasible cutter should be chosen thus maximizing the cutting efficiency. The latter selects a tool-path pattern, generates the cutter-contact (CC) points that satisfy the accuracy requirement,

and determines the cutter's posture (orientation) at every CC point without causing any interference. For a successful process planning system, it should be possible to automate these two tasks without assistance of an expert in 5-axis machining.

However, current commercial CAM systems are inadequate in the automatic process planning for 5-axis milling. In these systems (e.g., DelCAM and Unigrpahics), tool-path generation is generally conducted on the basis of some user specifications, such as the cutter size that finish the given surface, path topology, cutting direction, and so on. It is difficult for the user to select the suitable specification for finishing the surface. A trial-and-error approach is usually utilized, i.e., the user picks a set of parameters based on experience and then conducts the tool-path generation program to check their feasibility. Since the resultant tool-path might have to be manually corrected by tweaking the G codes, this trial-and-error approach can be very time-consuming. Alternatively, the user may choose the most conservative parameters to play safe, which will certainly compromise the machining efficiency. Therefore, it is necessary to develop automated approaches for the tasks in process planning to improve the practicality of 5-axis sculptured surface machining.

1.4 State-of-the-art in Process Planning for Sculptured Surface Machining

Since the late of 1980's, numerous amount of work has been published for the automation of process planning. A number of surveys and reviews have been presented on the issues in process planning. Dragomatz and Mann (1997) provided a classified bibliography of literature on NC milling path generation from 220 papers. Jensen and Anderson (1996) presented a mathematical review of methods and algorithms to place the milling cutter for multi-axis machining. Choi and Jerard (1998)

gave an extensive introduction of 5-axis sculptured surface machining from the aspect of the fundamental mathematics, the avoidance of undercutting and overcutting, path simulation and verification and so on.

Recently, there is a large body of work published in the area of the process planning for 5-axis sculptured surface machining. They mainly focus on the following aspects.

- **Interference detection and correction**

Generally, interference in machining can lead to the inaccurate part size, bad machining surface quality, and possible damage of the cutter and/or the machine tool. Thus, interference avoidance is the basic requirement for both cutter selection and tool-path generation in process planning, as shown in Figure 1.2. Much effort has been made on finding the cutter posture for interference avoidance in tool-path generation. The reported work is mainly categorized into two groups: detect-and-correct and accessible range. The former is to repeat the procedure of interference detection and correction until no interference for a cutter posture (Li and Jerard, 1994; Pi *et al.*, 1999). The merit of this approach is easy analysis and reasonable computation efficiency. However, it cannot achieve optimization in tool-path generation, such as the optimization of the cutter posture at the CC point. The latter approach evaluates the feasible range of cutter orientation with which the cutter can access the surface without interference (Choi *et al.*, 1993; Woo, 1994; Lee, 1997; Morishige *et al*., 1999). The accessible range can then be utilized for the optimization of tool-path data. Compared to the detect-and-correct approach, computation efficiency is the main drawback of the latter approach. In general, most of these approaches were formulated with a specified feeding direction, since they were proposed for tool placement in tool-path generation.

- **Cutter selection**

Cutter selection is to select an interference-free cutter with high machining efficiency. Several techniques have been developed for the automation of cutter selection in 3-axis NC machining. For example, cutter selection in roughing cut (Bard and Feo, 1989; Lee *et al.*, 1992; Lee *et al.*, 1994), and in finishing cut (Bala and Chang, 1991). The reported work on cutter selection for 5-axis finish cut machining is limited (Lee and Chang, 1996; Jensen *et al.*, 2002). To some extent, the reported methods for 5-axis cutting are trial-and-error in nature by selecting a cutter and conducting the procedure of tool-path generation, leading to heavy computation or compromise of machining efficiency. So far, there is no effective method that can determine whether a cutter is suitable for finishing a given surface before tool-path generation in 5-axis finish cut.

- **Path topology**

Sculptured surface machining is a point-milling process where a sequence of CC points are traced by milling cutters. When a region is machined by the point-milling method, the pattern of 'tracing' or scanning is called tool-path topology (Marshall and Griffiths, 1994). Many patterns has been studied for surface machining, such as serial-pattern (Ding *et al.*, 2003), radial-pattern (Kim and Choi, 2002), and contour-pattern (Park, 2003). Both the serial-type and radial-type are for machining one area, and the contour-type is for cutting a vertical or slant wall (Choi and Jerard, 1998). Recently, serial-type paths are extensively investigated in theory to improve the machining efficiency, such as non-iso-parametric path (Lee, 1998) and constant scallop path (Li and Feng, 2004). However, the traditional ***XY***-parallel (iso-planar) is still widely employed in practice because of its robustness in almost every scenario, for example, the machining of both the compound surface and trimmed surface.

In iso-planar path generation, cutting direction is a specification that has to be manually or automatically determined before tool-path generation. Lakkaraju and Raman (1990) pointed out that there must exist an optimum path for every shape at a specific orientation. Some work has been reported in the selection of cutting direction in area-machining from different aspects (Held, 1991; Sarma, 1999; Park and Choi, 2000). Unfortunately, these algorithms are limited to two-dimensional area manufacturing with fixed tool axis direction. They cannot be directly extended to 5-axis sculptured surface machining since the dynamically changing cutter posture should be considered for 5-axis machining.

- **Tool-path generation**

In addition to the path topology, the main task in 5-axis tool-path generation is the determination of CL data, including positions and corresponding cutter postures. It is desirable to have the shortest length of paths to cover the surface for high machining efficiency, i.e., with largest path interval between adjacent paths while keeping the machining scallop height within the desirable tolerance. Much work has been done in this direction by investigating the cutting shape at a CC point. Generally, the maximum path interval value is achieved from current CC point by repeating the procedure of (1) searching for the estimated CC point; (2) searching for the suitable cutter posture at this point; and (3) calculating the scallop height value on the basis of cutting shape (Lee, 1998). The procedure is repeated till the scallop height value reaches its maximum possible value, i.e., just bellow the user-defined tolerance. The step for searching cutter posture is essential in this procedure. Since the traditional optimization algorithm (searching from accessible range at this point) for optimal cutter posture is time-consuming, the computation load for tool-path generation can be very extensive.

1.5 Research Motivation

Process planning plays a vital role in achieving efficiency and accuracy for sculptured surface machining. Owing to two rotational axes, process planning becomes complicated and cumbersome for 5-axis milling mode. The process planning tasks include cutter selection and tool-path generation. Currently, there is no commercially available CAM software package that provides comprehensive automated process planning functions for 5-axis milling. The CAM systems are still mainly dependent on user-interaction and often lack flexibility (Chiou and Lee,2002a). On the other hand, there has been much research work aimed at achieving automated process planning in this domain. Generally speaking, the reported work mainly focuses on tool-path generation by considering interference avoidance, scallop-height control, and cutting efficiency, in which the geometric issues have been well studied. However, it is also noticed that cutter selection and tool-path generation are treated as separate tasks, i.e., a cutter and cutter feeding direction is pre-determined before tool-path generation. Even among the limited reported work in cutter selection, a cutter is selected in a trial-and-error manner by repeating tool-path generation procedure, leading to extensive computation load because of the complicated nature of the tool motion in 5-axis cutting. Another drawback for this trial-and-error approach is that the tool-path optimization is not ensured since the tool-path topology should be specified by the user before cutter selection and tool-path generation.

To improve the practicability of 5-axis milling, algorithms should be explored to speed up the computation in process planning and improve the cutting efficiency in sculptured surface machining. Based on this motivation, this research proposes an integrated process planning system, which accomplishes the task of cutter selection and tool-path generation in an integrated and efficient manner.

1.6 Research Objectives and Scope

The general goal of this research is to develop a process planning system to select an optimal cutter and generate efficient tool-paths to finish a sculptured surface in an integrated and efficient way. The cutter considered is a cylindrical cutter with a fillet-end, which also covers the flat-end and ball-end types. Three objectives have been set out as follows:

(1) Propose an approach for interference detection and correction when a cutter is posed at a single surface point,

(2) Develop a methodology for cutter selection before tool-path generation, and

(3) Develop algorithms for generating optimal tool-paths based on the checking result of cutter selection.

The research scope has been recognized as follows:

For the 1st objective, the research scope covers:

- Define cutter accessibility to a surface point using accessibility map (A-map) without considering feeding direction,
- Identify A-map of a cutter based on local surface property of both the part surface and the cutter,
- Identify A-map of a cutter based on global surface property of both the part surface and the cutter.

For the 2nd objective, the research scope covers:

- Identify accessibility of a single cutter to the part surface based on A-map analysis,
- Develop a time-saving method for accessibility evaluation of a single cutter to the part surface based on surface decomposition technique,
- Develop a non-redundant approach for cutter selection from available

ones based on accessiblity comparison between cutters.

For the 3rd objective, the research scope covers:

- Identify and describe the machining strategies in 5-axis milling,
- Identify optimal cutting direction in path planning for a selected tool-path pattern with the criteria of machining strategies,
- Develop an interference-free tool-path generation system in an efficient manner and with high cutting productivity.

1.7 Organization of the Thesis

In this thesis, Chapter 2 presents a point-based algorithm to evaluate the A-map in respect of machine limits, local- and rear-gouging avoidance, and global-collision avoidance through geometric analysis. Chapter 3 discusses an approach to efficiently check the accessibility of a single cutter based on A-map algorithm on the surface point and surface decomposition technique. Further, to speed up the computation in cutter selection from a set of cutters, a non-redundant algorithm is presented in Chapter 4 based on accessibility comparison between cutters. After the process of cutter selection, an algorithm is designed to use the checking result from cutter selection in the determination of optimal cutting direction for iso-planar paths in Chapter 5. In addition, an efficient and optimal algorithm is proposed in Chapter 6 to use the checking result from cutter selection in tool-path generation. An application example is given in Chapter 7 to validate the efficiency and effectiveness of the proposed algorithms. Chapter 8 draws the conclusions by discussing achievements and limitations of the research proposal, and the avenues of the possible future research.

CHAPTER 2

CUTTR ACCESSIBILITY TO A SURFACE POINT

In 5-axis machining, using the two additional revolute axes, the cutter's orientation can be altered to access the complicated sculptured part surface for better cutting. However, the flexibility also brings with it the complexity in process planning. Apart from accuracy concern, cutter accessibility is the most important issue to be considered in the two tasks for 5-axis machining (finish cut): cutter selection and tool-path generation. To be more specific, at a point on the surface, if the cutter has a posture that causes no interference, the cutter is said to be accessible to this point. One of the critical issues addressed in this work is to design an algorithm for finding cutter accessibility to a point which can be utilized in both cutter selection and tool-path generation.

2.1 Introduction

5-axis machining is employed in the situation where increased cutter accessibility can reduce the number of set-up process or where improved surface finish is necessary (Choi *et al.*, 1993). However, owing to the two rotational DOFs, tool movement can become complicated and cumbersome. Additional factors, related to tool positioning and movement, also contribute to making process planning a complex and error prone process. For example, violation of geometric constraints may be placed on the machine's tool axes. Unintentional localized gouging of the part surface by the cutter is another concern that has to be addressed in both cutter

selection and tool-path generation. Gouging must be avoided otherwise the resultant surface would have localized surface flaws and might not satisfy the surface accuracy or texture specification. Further, it is also needed to consider the collision between the cutter and the workpiece, or surrounding objects such as fixtures. Cutter collision might result in the damage of the cutter, machined part, or the machine tool.

In this chapter, new techniques are proposed to check whether a fillet-end cutter can access a point on the sculptured surface without interference. In this work, interference refers to the machine axis limits, local- and rear- gouging, and global-collision.

2.2 Literature Review

To successfully machine a sculptured surface, machine axis limits, avoidance of gouging and collision between the cutter and the part surface must be guaranteed. The two additional axes for rotation in 5-axis machining permits the improvement of the machined surface quality and accessibility (Vickers and Quan, 1989). Each axis of the machine tool has a limit, which is specific to the machine configuration and beyond which any further motion is prevented. These limits introduce a problem for 5-axis machining if a transition from one tool position and orientation to the next lets an axis beyond its limit (usually specified in the control software) (Xu *et al.*, 2002).

Local-gouging refers to removal of excess material in the vicinity of a CC point due to the mismatch in curvatures between the cutter (swept along the path) and the part surface at the CC point. Rao and Sarma (2000) detected and avoided the local-gouging by matching the effective cutting curvature of the flat-end tool swept surface with the normal curvature of the part surfaces. Chiou (2004) presented an approach to determining gouging-free tool position for 5-axis ruled surface machining

based on the explicit analysis of the swept profile. A trial-and-error approach was utilized by repeating the algorithms of calculating tool swept profile, detecting tool positioning error (checking the intersection of the swept profile and the part surface), and repositioning the tool. The most valuable merit for swept profile approach is its accuracy to represent the machined surface. However, computational complexity and efficiency problem are also introduced from the analysis of swept profile for the cutter with simultaneous translational and rotational motion, and the calculation of the distance between two free-form surfaces (swept profile surface and the part surface). Thus, the approaches with swept volume are generally complicated and difficult to be performed. Much effort was made to simplify the analysis by approximating the effects of the cutter swept surface. Many studies use the instantaneous cutter geometry at each surface point to replace the swept surface for local-gouging detection. In curvature matched machining proposed by Pi *et al*. (1999), the effective curvatures of the cutter is evaluated to match the curvatures of the part surface at the normal and osculating planes for gouging avoidance and efficient machining. Yoon *et al.* (2003) pointed out that these algorithms use some rough approximations, such as "effective cutting shape" at two planes in order to determine a locally optimal cutter position. This may lead to unwanted collisions. They presented a local condition for gouging-free 5-axis milling of sculptured surfaces by considering the curvatures of cutter and part surfaces along all possible directions.

Rear-gouging refers to the removal of excess material due to intrusion of the cutter bottom surface into the part surface. It is another source of overcut to affect the machined surface accuracy and must be eliminated for a proper machining of sculptured surfaces. Much effort has been made on the study of rear-gouging avoidance. Li and Jerard (1994) observed that tool movement affects only a small

portion of the tessellated surface and suggested localized interference checking using a bucketing strategy. Once interference is detected, the tool is inclined away from the interference until it barely touches the gouging and colliding triangle facets. The merit of this approach is good computation efficiency. However, the final cutter orientation is searched by a non-deterministic approach and other potential orientations are not considered. One or more of the alternative solutions may be superior in machining when checked with respect to other criteria as well. On the other hand, Lee (1997) presented algorithms of admissible tool orientation control for gouging avoidance in 5-axis machining with a fillet-end cutter. Utilizing a triangular polyhedral description of the surface, gouging avoidance was studied by Xu *et al.* (2002) to evaluate a feasible domain, which can be employed for the optimization of cutter orientation.

For successful machining, interference between non-cutting portions of the tool and the surface, refereed as ***global-collision*** in this work, also should be considered since it leads to the bad surface quality and possible damage of the cutter and the machine tool. Some studies proposed algorithms to avoid collision based on a trial-and-error process, where the provisional determination of a cutter orientation is repeated until collision does not occur. Lee and Chang (1995) proposed a 2-phase approach to solve the problem of global tool collision in 5-axis sculptured surface machining. Convex hull of the sculptured surface is utilized to find a conservative feasible tool orientation. If collision between the convex hull and the tool is detected, checking on interference is further performed and tool orientation is corrected if needed. Similarly as the algorithms for rear-gouging avoidance, the merit of finding collision-free tool orientation by gradually adjusting the orientation is the computation efficiency. However, this method cannot achieve optimal tool orientation and tool-path. An elegant concept known as the *C*-space (configuration space) has been

developed and explored by a few researchers. In *C*-space, each point specifies a particular position in the space. By mapping obstacles to the *C*-space, the collision-free access can be theoretically inferred by simply navigating the point around the obstacles in the *C*-space. Choi *et al.* (1997) stated that even though the term '*C*-space' has rarely been used in die-cavity machining, the *C*-space is not completely new in the field of tool-path generation. For example, the CL-checking approach is similar to the *C*-space approach where the cutter is the moving object, and the design-surface and stock-surface are the obstacles. Morishige *et al.* (1999) used the *C*-space at each CC point to produce a smooth and continuously varying toolpath. Although intuitive and intellectually appealing, a major problem of the *C*-space approach is the computational intractability when mapping obstacles to the *C*-space. Woo (1994) first demonstrated the use of visibility cones, an alternative representation of the configuration, for part accessibility analysis. Balasubramaniam *et al.* (2000) presented a new way of computing discrete visibility information using hidden surface removal methods and used this information to generate collision-free roughing paths. This approach takes advantage of the fast computation of graphics hardware to achieve efficiency. However, visibility is only a necessary condition for interference avoidance in process planning for machining. The tool geometry cannot be modeled as an abstract straight line and its radius (for an end mill) must be accounted for to achieve the subtle geometry required for interference avoidance (Xu *et al*., 2002).

In the reported literature of interference avoidance, most approaches are designed for automated tool-path generation based on a given feeding direction. They can be adopted for interference avoidance for cutter selection by using a trial-and-error approach. However, at the stage of cutter selection in process planning, the tool-path pattern and cutting direction is preferably not fixed, and a more general

algorithm for interference avoidance is needed for optimal machining. Nevertheless, the reported algorithms for avoiding interference in tool-path generation provide useful references to develop a more general cutter accessibility analysis method.

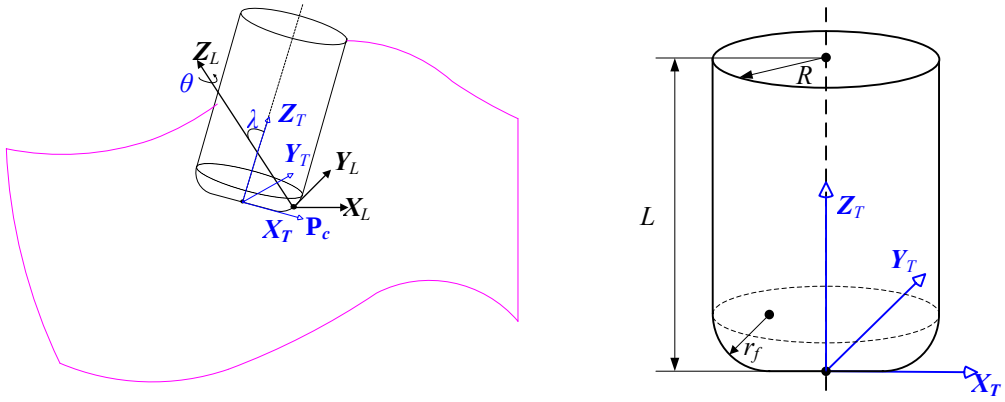
In this chapter, a point-based method is proposed to deal with the cutter accessibility problem without considering the feeding direction. In addition, as mentioned above, although the trial-and-error approach is easy and efficient to handle the interference problem, the accessible range approach is much more effective in the optimization of process planning. Thus, accessible range approach is adopted in this work. Since the feeding direction is not considered, the algorithms proposed can be utilized in both cutter selection and tool-path generation.

2.3 Point-based Cutter Accessibility Checking

In general, there are four attributes to a cutter's accessibility to a point on the surface: the machine axis limits, local-gouging, rear-gouging, and global-collision. In this section, the algorithms to check a given cutter's accessibility in terms of the four attributes are introduced. The objective is to check, at the point, whether there exists a posture at which the cutter is interference-free. Given a point, we firstly identify the accessible posture range of the cutter based on each attribute. If there is no accessible range for an attribute, the search is stopped and the cutter is labeled as non-accessible. The common accessible range among the four ranges, referred to as ***accessibility map*** (A-map) in this work, is then identified and if the common range exists, the cutter is accessible at the point.

Before the algorithms are described, three coordinate frames are firstly introduced: machine frame, local frame and tool frame. Machine frame is the universal coordinate system related to the machine configuration in which the design

surface lies. Local frame is defined according to the surface geometry at the point of interest $\mathbf{P}_c$. As shown in Figure 2.1a, the local frame ($\boldsymbol{X}_L$, $\boldsymbol{Y}_L$, $\boldsymbol{Z}_L$) originates at $\mathbf{P}_c$ with $\boldsymbol{Z}_L$-axis along the normal vector, $\boldsymbol{X}_L$-axis along the surface maximum principal direction, and $\boldsymbol{Y}_L$-axis along the surface minimum principal direction. A cutter's orientation is defined by an angle pair (λ, θ) meaning that the cutter's axis inclines counter-clockwise with λ about $\boldsymbol{Y}_L$-axis and rotates a θ about $\boldsymbol{Z}_L$-axis, where $0° \leq \lambda \leq 90°$ and $0° \leq \theta \leq 360°$. Tool frame ($\boldsymbol{X}_T$, $\boldsymbol{Y}_T$, $\boldsymbol{Z}_T$) is defined with its origin at the cutter bottom centre while its $\boldsymbol{Z}_T$-axis along the cutter axis direction. The intersection line between the bottom plane and the plane defined by the $\boldsymbol{Z}_T$-axis and $\mathbf{P}_c$ defines the $\boldsymbol{X}_T$-axis that points towards $\mathbf{P}_c$. The $\boldsymbol{Y}_T$-axis is defined by $\boldsymbol{Y}_T = \mathbf{Z}_T \times \boldsymbol{X}_T$. θ is 0 when the $\boldsymbol{X}_L$-axis and $\boldsymbol{X}_T$-axis are co-planar, and $\lambda = 0$ when the $\boldsymbol{Z}_L$-axis and $\boldsymbol{Z}_T$-axis are parallel.



(a) Local frame and tool frame (b) Cutter geometry

Figure 2.1 A fillet-end cutter at $\mathbf{P}_c$ in the local frame and tool frame

A cutter is generally represented in the tool frame. As shown in Figure 2.1b, a fillet-end cutter (R, r_f, L) consists of three portions: the cylindrical portion with major radius R, the filleted portion with minor radius r_f, and the circular bottom planar portion with radius $r_1 = R - r_f$. The cutter length is represented by L.

Rear-gouging occurs if cuter bottom surface is underneath the part surface, and global-collision occurs if cutter's non-cutting portion protrudes into the part surface. Hence, the detection of rear-gouging and global-collision is in fact a distance-evaluation problem. However, a numerical method is the only solution to solve this problem involving the evaluation of distance between the cutter surface and the complex sculptured surface. This is very time consuming and sometime leads to no convergence. In this work, instead of distance evaluation between surfaces, the given sculptured surface is sampled to a set of sampled points, and the point-based analysis is developed to obtain the accessible range of rear-gouging and global-collision avoidance. Some reported work (Piegl and Richard, 1995; Piegl and Tiller, 1998) can be employed for sampling of sculptured surfaces.

In the following sections, the algorithms to obtain the accessible range (λ, θ) for a given cutter, if such a range exists, are introduced. Among the 4 attributes, identifying the accessible range based on machine limits is rather straightforward (Xu *et al*., 2002), which is not to be covered here. The discussion focuses on identifying accessible ranges for the avoidance of local-gouging, rear-gouging, and global-collision.

2.3.1 Accessible range for local-gouging avoidance

Local-gouging occurs when the curvatures of the cutter's local surface are less than those of the part surface at the point of interest such that the cutter cuts excess material. Therefore, given a posture (λ, θ) of the cutter, the normal curvatures of the cutter and the part surface at the CC point in every possible direction need to be compared to ensure the prevention of local-gouging. According to Euler's formula (O'Neil, 1966), as shown in Figure 2.2, the normal curvature of the curve at the CC

point ($\mathbf{P}_c$) along any direction $\boldsymbol{x}_\omega$ (the angle between x_ω and $\boldsymbol{X}_L$-axis is ω and $0 \le \omega \le 2\pi$) on the tangent plane is given as,

$$\kappa_{s\omega} = \kappa_{\max} \cos^2 \omega + \kappa_{\min} \sin^2 \omega \tag{2.1}$$

Where $\kappa_{\max}$ and $\kappa_{\min}$ are the maximum and minimum principal curvatures of local part surface at $\mathbf{P}_c$, respectively.

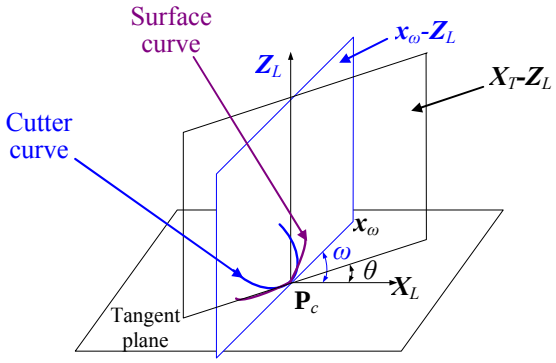


Figure 2.2 The cutter and surface curve on a normal plane containing $\boldsymbol{x}_\omega$ at $\mathbf{P}_c$

For a fillet-end cutter, the cutting edge is located on the filleted portion of the cutter surface. At the CC point, the cutter surface normal coincides with the part surface normal, and the principal curvatures of the cutter surface can be expressed as (Jensen *et al.*, 2002):

$$\kappa_{t\min} = \frac{1}{\dfrac{(R - r_f)}{\sin\lambda} + r_f}$$
$$\kappa_{t\max} = \frac{1}{r_f} \tag{2.2}$$

Where $\kappa_{t\max}$, and $\kappa_{t\min}$ are the normal curvatures of cutter curves at $\mathbf{P}_c$ on the $\boldsymbol{X}_T$-$\boldsymbol{Z}_L$ plane and $\boldsymbol{Y}_T$-$\boldsymbol{Z}_L$ plane, respectively. The normal curvature of the cutter curve on the normal plane containing $\boldsymbol{x}_\omega$ (see Figure 2.2) is then given as:

$$\kappa_{t\omega} = \kappa_{t\max}\cos^2(\omega-\theta) + \kappa_{t\min}\sin^2(\omega-\theta) \tag{2.3}$$

To make sure that the cutter is free of local-gouging at this point, we must have $\kappa_{t\omega}$-$\kappa_{s\omega} > 0$. Therefore, we have,

$$\begin{aligned}\kappa_{t\omega} - \kappa_{sw} = &(\kappa_{t\max}\cos^2\theta + \kappa_{t\min}\sin^2\theta - \kappa_{\max})\cos^2\omega \\ &+ [2(\kappa_{t\max} - \kappa_{t\min})\sin\theta\cos\theta]\sin\omega\cos\omega \\ &+ (\kappa_{t\max}\sin^2\theta + \kappa_{t\min}\cos^2\theta)\sin^2\omega > 0\end{aligned} \tag{2.4}$$

When $\omega = 0$, the following inequality must be satisfied,

$$\kappa_{t\max}\cos^2\theta + \kappa_{t\min}\sin^2\theta - \kappa_{\max} > 0 \tag{2.5}$$

When $\omega \neq 0$, by dropping $sin^2\omega$ from both sides of inequality (2.4), we have,

$$a\times\cot^2\omega + b\times\cot\omega + c > 0 \quad \text{where} \quad \begin{aligned} a &= \kappa_{t\max}\cos^2\theta + \kappa_{t\min}\sin^2\theta - \kappa_{\max} \\ b &= 2(\kappa_{t\max} - \kappa_{t\min})\sin\theta\cos\theta \\ c &= \kappa_{t\max}\sin^2\theta + \kappa_{t\min}\cos^2\theta - \kappa_{\min}\end{aligned} \tag{2.6}$$

It can be seen from Eq. (2.2) that when λ increases ($0°\leq\lambda\leq 90°$), $\kappa_{t\min}$ increases while $\kappa_{t\max}$ remains the same, and therefore, $\kappa_{t\omega}$ increases according to Eq. (2.3). On the other hand, $\kappa_{s\omega}$ remains the same when λ increases. Therefore, $\kappa_{t\omega}$ - $\kappa_{s\omega}$ is an increasing function in terms of λ. To satisfy inequality (2.6), we have,

$$b^2 - 4ac < 0 \tag{2.7}$$

Re-arranging inequalities (2.5) and (2.7) and combining Eq. (2.6), the conditions for no local-gouging are:

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

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

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 $[\lambda_{\theta\text{-}lg}, 90°]$, where $\lambda_{\theta\text{-}lg} = \max(\lambda_1, \lambda_2)$. 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., $[\theta, (\lambda_{\theta\text{-}lg}, 90°)]$, with computation efficiency O(1).

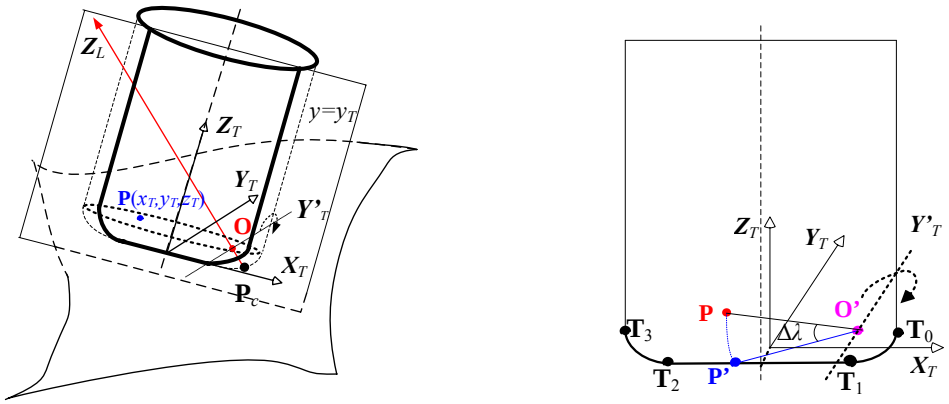
2.3.2 Accessible range for rear-gouging avoidance

For a given θ, we now need to identify an accessible range $[\theta, (\lambda_{\theta\text{-}rg1}, \lambda_{\theta\text{-}rg2})]$ such that cutter bottom surface does not protrude into the part surface. To conduct this search, we first identify all the candidate points on the part surface that have the possibility of causing rear-gouging, thus minimizing the search time. For each rear-gouging candidate point $\mathbf{P}_i | i=1, \ldots, n$, where n is the total number of candidate points, the accessible range $(\lambda_{\theta\text{-}rg1\text{-}i}, \lambda_{\theta\text{-}rg2\text{-}i})$ is then obtained. The common range of all the $(\lambda_{\theta\text{-}rg1\text{-}i}, \lambda_{\theta\text{-}rg2\text{-}i}) | i=1, \ldots, n$, is taken as the $(\lambda_{\theta\text{-}rg1}, \lambda_{\theta\text{-}rg2})$.

Referring to Figure 2.3, with a different posture (λ in this case), the fillet-end cutter will have different contact point on the filleted portion. Its pivot point **O** is along the normal vector of the CC point $\mathbf{P}_c$ with a distance r_f from $\mathbf{P}_c$. It can be easily

shown that, for any point $\mathbf{P}_{tk}$ on the cutter bottom surface (including the planar and the filleted portion), $|\mathbf{OP}_{tk}| \le 2R - r_f$. Thus the candidate points on the part surface for rear-gouging check should be within a distance range of $2R - r_f$ from **O**. In addition, only those points on the surface that are above the tangent plane can possibly cause rear-gouging. Therefore, a candidate point, $\mathbf{P}(x_T, y_T, z_T)$, must satisfy $|\mathbf{OP}| \le 2R - r_f$ and $\mathbf{P}_c\mathbf{P} \bullet \boldsymbol{Z}_L > 0$. Furthermore, there are some other simple criteria that can be used to check whether the candidate point may cause rear-gouging. For example, when θ is fixed, the cutter rotates about axis $\boldsymbol{Y'}_T$ that is parallel to $\boldsymbol{Y}_T$-axis and passes through the pivot point **O**, which means that y_T is constant in the tool frame. Therefore, y_T must be within the range of $-R \le y_T \le R$ if it is rear-gouging prone.

Now, we show under what condition that a rear-gouging prone point, **P**, causes rear-gouging. If we use a plane $y=y_T$ to section the cutter bottom surface, a section curve is produced (see Figure 2.3b), which consists of 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. If **P** is above the section curve, rear-gouging occurs. If we increase λ by rotating the cutter about axis $\boldsymbol{Y'}_T$, **P** tends to move towards underneath the section curve. Therefore, we need to find the minimum λ such that **P** is on the cutter outer surface at position **P**' (see Figure 2.3b).



(a) Gouging-prone **P** (x_T,y_T,z_T) and the cutter

(b) Section curve on cutter at $y=y_T$

Figure 2.3 Identifying cutter posture range for rear-gouging avoidance

Given a candidate point $\mathbf{P}(x_T, y_T, z_T)$, we start with the cutter posture at $\lambda = 0$ and obtain the section curve of the cutter bottom at $y = y_T$, as shown in Figure 2.3b. **O'** is the intersection point between axis $\boldsymbol{Y'}_T$ and plane $y = y_T$. If **P** is below the section curve, the accessible posture range, in terms of λ, is [0, 90°]. Otherwise, we have to adjust angle λ. By rotating the cutter about axis $\boldsymbol{Y'}_T$, it can be seen that **P** will reach the cutter bottom surface at a corresponding point $\mathbf{P}'$, which may fall into different segments in the section curve. Depending on which segment $\mathbf{P}'$ falls into, calculation of the increment $\Delta\lambda$ that moves **P** to $\mathbf{P}'$ is different. We define d as the distance between **P** and axis $\boldsymbol{Y'}_T$, and use d_0, d_1, d_2, and d_3 to represent the distance from points $\mathbf{T}_0$, $\mathbf{T}_1$, $\mathbf{T}_2$, and $\mathbf{T}_3$ to $\boldsymbol{Y'}_T$, respectively. The calculation of the increment $\Delta\lambda$ is given as follows:

(1) When $d_1 \le d \le d_2$, $\mathbf{P}'$ falls between $\mathbf{T}_1$ and $\mathbf{T}_2$, and its coordinates in the tool frame are given as $\mathbf{P}'(x'_T, y'_T, z'_T) = (r_1 - \sqrt{d^2 - r_f^{\,2}}, y_T, 0)$. The increment $\Delta\lambda$ is calculated as:

$$\Delta\lambda = \cos^{-1}\left(\frac{r_f - z_T}{d}\right) - \cos^{-1}\left(\frac{r_f}{d}\right) \tag{2.10}$$

(2) When $d_0 \le d \le d_1$ or $d_2 \le d \le d_3$, **P'** falls between $\mathbf{T}_0$ and $\mathbf{T}_1$, $\mathbf{T}_2$ and $\mathbf{T}_3$, respectively. Its coordinates in the tool frame are $\mathbf{P}'(x'_T, y'_T, z'_T) = \left(a + \frac{a^2 + y_T^{\,2} - r_1^2}{2(r_1 - a)}, y_T, r_f\left(1 - \sqrt{1 - \left(\frac{a^2 + y_T^{\,2} - r_1^2}{2r_f(r_1 - a)}\right)^2}\right)\right)$, where $a = \frac{2r_1^2 + r_f^{\,2} - d^2 - y_T^{\,2}}{2r}$. The increment $\Delta\lambda$ is calculated as:

$$\Delta\lambda = \sin^{-1}\left(\frac{z_T - r_f}{d}\right) + \sin^{-1}\left(\frac{r_f - z'_T}{d}\right) \tag{2.11}$$

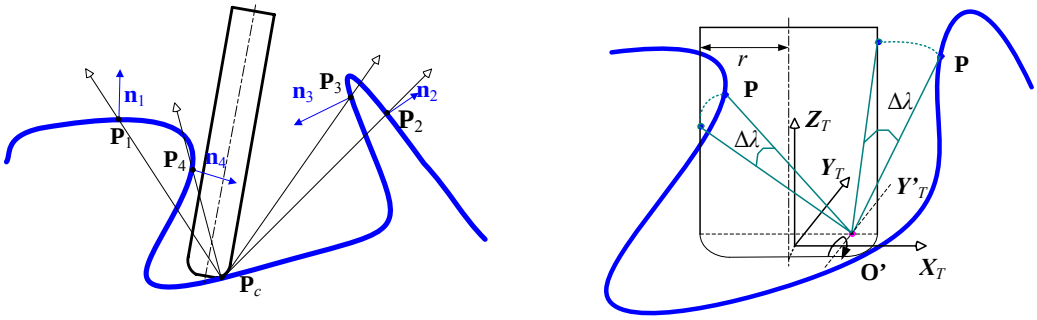
Since the initial λ is set as 0, the minimum inclining angle λ for the cutter to avoid rear-gouging at point **P** is $\Delta\lambda$ and the accessible posture range, in terms of λ, for the cutter at **P** is $[\Delta\lambda, 90°]$, if $\Delta\lambda \leq 90°$. If $\Delta\lambda$ is outside [0, 90°], it means that at θ, rear-gouging cannot be avoided. θ needs to be increased and the search restarts from local-gouging avoidance. Using this method, we can identify the accessible ranges for rear-gouging avoidance at θ for all the rear-gouging candidate points as $\Delta\lambda_i|i=1,\ldots, n$. The overall accessible range for no rear-gouging is $[\theta, (\lambda_{\theta\text{-}rg}, 90°)]$, where $\lambda_{\theta\text{-}rg} = \max\{\Delta\lambda_i|i=1,\ldots, n\}$. 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.

2.3.3 Accessible range for global-collision avoidance

We now proceed to find an accessible range $[\theta, (\lambda_{\theta\text{-}gc1}, \lambda_{\theta\text{-}gc2})]$ such that the cutter shaft does not intersect with the part surface. In theory, given a posture, a point collides with the cutter if it falls "inside" the cutter, in which the cutter length needs to be considered. Here, we consider that the point collides with the cutter if the distance between the point and the cutter axis is less than R, i.e., we assume the length of the cutter to be infinite. This may help to cover the constraints imposed by the cutter's holder, which is not considered here. However, the constraints of the cutter's holder can be easily incorporated into this collision-avoidance algorithm if the geometry of the holder is given. Like the search procedure for rear-gouging, we first identify the candidate points on the part surface that have the possibility of causing global-collision, thus minimizing the search time. For each global-collision candidate point $\mathbf{P}_i|i=1, \ldots, n$, where n is the total number of candidate points, the accessible range $[\lambda_{\theta\text{-}}$

$_{gc1\text{-}i}$, $\lambda_{\theta\text{-}gc2\text{-}i}$] is then obtained. The common range of all the [$\lambda_{\theta\text{-}gc1\text{-}i}$, $\lambda_{\theta\text{-}gc2\text{-}i}$] $|i=1, \ldots, n$, is taken as the [$\lambda_{\theta\text{-}gc1}$, $\lambda_{\theta\text{-}gc2}$].

Referring to Figure 2.4a, it can be seen that in the vicinity of the point $\mathbf{P}_c$, those surface points that "face" the cutter have the possibility to interfere with the cutter shaft. These surface points, such as $\mathbf{P}_3$ and $\mathbf{P}_4$, are identified as collision-prone candidates for analysis. The normal vector $\mathbf{n}_i$ at one specific surface point $\mathbf{P}_i$ can be utilized to determine whether it "faces" the cutter or not, i.e., if $\mathbf{n}_i \bullet \mathbf{P}_c\mathbf{P}_i < 0$, $\mathbf{P}_i$ is "facing" the cutter. In Figure 2.4a, $\mathbf{P}_3$ and $\mathbf{P}_4$ are determined as "facing" the cutter and labeled as collision-prone points, while $\mathbf{P}_1$ and $\mathbf{P}_2$ are collision-free points. On the other hand, y_T must be within the range of $-R \le y_T \le R$ if it is global-collision prone.



(a) Collision-prone and collision-free points (b) Section curve on cutter at $y=y_T$

Figure 2.4 Identifying cutter posture range for global-collision avoidance

Now, we show how to find the collision-free accessible range for a global-collision prone point, $\mathbf{P}(x_T, y_T, z_T)$. We firstly use a plane $y = y_T$ to section the cutter surface (at $\lambda = 0$), a section curve is produced as shown in Figure 2.4b, which shows the same section curve as the one in Figure 2.3b, but it is presented here for a clear discussion on collision avoidance. The radius of the section curve is $r = \sqrt{R^2 - y_T^2}$. In rear-gouging avoidance, we know that if a rear-gouging prone point does not cause

gouging at $\lambda = 0$, its accessible range, in terms of λ, is [0, 90°]. Otherwise, we need to find the minimum $\Delta\lambda$ that leads to the avoidance of rear-gouging. In global-collision avoidance, however, the posture range for every collision prone point is needed to be found even it does not collide with the cutter at $\lambda = 0$. For example, referring to Figure 2.4b, if a point causes collision (e.g., **P** at the rear), i.e., it is inside the section curve (above the cutter bottom portion), we need to find the minimum $\Delta\lambda$ the cutter must be rotated forward to avoid the collision. In this case, the accessible range for no global-collision is [$\Delta\lambda$, 90°]. If the point does not cause collision (e.g., **P** to the front), we need to find the minimum $\Delta\lambda$ the cutter must be rotated forward such that the point touches the cutter shaft. In this case, the accessibility map for no global-collision is [0, $\Delta\lambda$]. The relative positional relationship between **P** and the section curve can be categorized into five cases and the methods that handle the different cases are given as follows:

(1) $z_T < r_f$, **P** is collision-free and its accessibility map is [0, 90°].

(2) $x_T < -r$, and $\mathbf{n} \bullet \boldsymbol{X}_T > 0$ ($z_T \geq r_f$), **P** is collision-free and its accessibility map is [0, 90°].

(3) $x_T \geq -r$, and $\mathbf{n} \bullet \boldsymbol{X}_T > 0$ ($z_T \geq r_f$), global-collision exists. The minimum $\Delta\lambda$ that the cutter must be rotated clockwise to avoid global-collision is:

$$\Delta\lambda = \cos^{-1}\frac{r_1 - x_T}{d} - \cos^{-1}\frac{r_1 + \sqrt{R^2 - {y_T}^2}}{d} \tag{2.12}$$

Where d is the distance from **P** to **O'**. The accessibility map is [$\Delta\lambda$, 90°].

(4) $x_T < r$ and $\mathbf{n} \bullet \boldsymbol{X}_T < 0$ ($z_T \geq r_f$), global-collision exists. The accessibility map is NULL.

(5) $x_T \geq r$, and $\mathbf{n} \bullet \boldsymbol{X}_T < 0$ ($z_T \geq r_f$), **P** is collision-free. The minimum $\Delta\lambda$ that the cutter must be rotated clockwise to cause global-collision is give as:

$$\Delta\lambda = \cos^{-1}\frac{r_1 - x_T}{d} - \cos^{-1}\frac{r_1 - \sqrt{R^2 - y_T^2}}{d} \tag{2.13}$$

The accessibility map is $[0, \Delta\lambda]$.

Using the above method, the accessible ranges for all global-collision prone points can be obtained as $[\lambda_{\theta\text{-}gc1\text{-}i}, \lambda_{\theta\text{-}gc2\text{-}i}]$ $|i=1, \ldots, n$. The overall accessible range for no global-collision is $[\theta, (\lambda_{\theta\text{-}gc1}, \lambda_{\theta\text{-}gc2})]$, where $\lambda_{\theta\text{-}gc1} = \max\{\lambda_{\theta\text{-}gc1\text{-}i}, |i=1, \ldots, n\}$ and $\lambda_{\theta\text{-}gc2} = \min\{\lambda_{\theta\text{-}gc2\text{-}i}, |i=1, \ldots, n\}$. 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 $\lambda_{\theta\text{-}gc1} > \lambda_{\theta\text{-}gc2}$, it means that the cutter is not accessible at this θ. θ needs to be increased and the search restarts from local-gouging avoidance.

2.3.4 The overall search algorithm

For a point on the part surface, the three methods introduced in Sections 2.3.1, 2.3.2, and 2.3.3 can be used to find $[\theta, (\lambda_{\theta\text{-}lg}, 90°)]$, $[\theta, (\lambda_{\theta\text{-}rg}, 90°)]$, and $[\theta, (\lambda_{\theta\text{-}gc1}, \lambda_{\theta\text{-}gc2})]$, representing the accessible ranges for the avoidance of local-gouging, rear-gouging, and global-collision, respectively. If a common region among the three accessible ranges is available, the cutter is accessible to the point. In practice, we need to combine the three methods into an overall search algorithm. It is worth noting that we assume the minimum and maximum values of the tilting angle λ as 0° and 90°, respectively. In practice, this can be generalized by using $\lambda_{\min}$ and $\lambda_{\max}$ instead. Similarly, the minimum and maximum values of the rotational angle θ are $\theta_{\min}$ and $\theta_{\max}$, respectively. The algorithm is described as follows:

Algorithm: ***Finding the accessible posture range of a cutter at a CC point P***

Input: *(a) All the points on the surface* $\{\boldsymbol{P}_k, k = 1, 2, ..., m\}$ *except the CC point* $\boldsymbol{P}$

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

(c) Titling angle range $[\lambda_{\min}, \lambda_{\max}]$, *rotational angle range* $[\theta_{\min}, \theta_{\max}]$

Output: *Accessibility map and the accessibility of the cutter at* $\boldsymbol{P}$

Begin

(1) Uniformly sample $(\theta_{\min}, \theta_{\max})$ into k angles, set $i = 0$.

(2) IF $i \leq (k\text{-}1)$, $\theta_i = \theta_{\min} + (\theta_{\max} - \theta_{\min})(i/(k\text{-}1))$; otherwise, go to (7).

(3) Find the local-gouging free accessible range $[\theta_i, (\lambda_{\theta\text{-}lg}, \lambda_{\max})]$, using the method introduced in Section 2.3.1. If such an accessible range does not exist, $i = i + 1$, go to (2).

(4) Find the rear-gouging free accessible ranges, from $(\lambda_{\theta\text{-}lg}, \lambda_{\max})$, for $\{\mathbf{P}_k, k = 1, 2, \ldots, m\}$. The common accessibility map is taken as $[\theta_i, (\lambda_{\theta\text{-}rg}, \lambda_{\max})]$, note that $\lambda_{\theta\text{-}rg} \geq \lambda_{\theta\text{-}lg}$. If such an accessible range does not exist, $i = i + 1$, go to (2).

(5) Find the global-collision free accessible ranges, from $(\lambda_{\theta\text{-}rg}, \lambda_{\max})$, for $\{\mathbf{P}_k, k = 1, 2, \ldots, m\}$. The common accessible range is taken as $[\theta_i, (\lambda_{\theta\text{-}gc1}, \lambda_{\theta\text{-}gc2})]$, note that $\lambda_{\theta\text{-}gc1} \geq \lambda_{\theta\text{-}rg}$ and $\lambda_{\theta\text{-}gc2} \leq \lambda_{\max}$. If such an accessible range does not exist, $i = i + 1$, go to (2).

(6) Output the accessible range (A-map) and "the cutter is accessible at **P**". Stop.

(7) Output "the cutter is not accessible at **P**". Stop.

End

It can be seen that the algorithm is, to a large extent, numerical in nature, except that the method to find the A-map for the avoidance of local-gouging is

analytical. The computation efficiency is O($k + km + km$) = O(km). Furthermore, the search for the accessible posture range considers only geometric concerns. Some technical concerns, such as the preferable tilting and orientation ranges, also need to be taken into consideration in cutter selection (Jensen *et al.*, 2002) or tool-path generation. This can be incorporated by specifying [λ_{min}, λ_{max}] and [θ_{min}, θ_{max}] before the search starts.

2.4 Summary

A comprehensive method has been presented to identify the A-map (accessible range), in terms of titling and rotational angles, for the accessibility analysis of a given fillet-end cutter to a point on a sculptured surface. The method addressed four major attributes to a cutter's accessibility to a point on the surface in multi-axis machining: (a) the accessible range constraints imposed by the machine tool's configuration; (b) local-gouging prevention between the local surface of the part and the cutter in the vicinity of the CC point; (c) rear-gouging avoidance between the part surface and the cutter; and (d) global-collision avoidance between the cutter shaft and the part. An accessible range is constructed for each of the attribute stated above. The common region among the four accessible ranges is then identified as the A-map to reflect the accessibility of the cutter to the surface point.

CHAPTER 3

CUTTER SELECTION PART 1:

CUTTER ACCESSIBILITY TO A SURFACE

Cutter selection is a very important issue in 5-axis tool-path generation for machining sculptured surfaces. To select the optimal cutter from a set of available ones, it is essential to check whether a cutter is able to finish the entire surface without any interference, i.e., to check the accessibility of a cutter at any point on the surface. In the last chapter, the issues involved in cutter accessibility checking at a surface point have been studied. A comprehensive search algorithm has been developed that is able to find the A-map at a point for a given cutter in terms of the tilting and rotational angles. This point-based solution can be extended to all surface points for the analysis of cutter accessibility to the surface. However, computation load is always a concern. In this chapter, an algorithm is proposed to reduce the checking regions in the cutter accessibility checking procedure by using surface decomposition technique.

3.1 Introduction

In the literature of process planning for 5-axis machining, most of the previous effort has been on developing automated methods that generate an interference-free tool-path for a given part surface (Choi *et al.*, 1993; Li and Jerard, 1994; Pi *et al.*, 1998; Xu *et al*., 2002; Chiou and Lee, 2002a; Ding *et al.*, 2003; Chiou, 2004). However, most of these reported methods assume that a cutter is already selected such

that the cutter can finish the given surface without interference, thus only focusing on generating a set of CC points that satisfy the accuracy requirement and generating the corresponding interference-free cutter postures at the CC points. This assumption may lead to potential problems in tool-path generation and the quality of the resultant tool-path. Firstly, a user may find it difficult to select a suitable cutter that is able to finish the surface. He/she may have to pick up a cutter based on experience and then apply the tool-path generation program to check the feasibility of the cutter. This trial-and-error approach can be very time-consuming. Alternatively, the user may choose the smallest cutter to play safe, which will certainly compromise the machining efficiency. So far, there is limited reported work on cutter selection in 5-axis machining and there is no effective reported method that can determine whether a cutter can finish a given surface before tool-path generation.

The cutter selection problem can be defined as "given a part surface, a 5-axis machine, and a set of cutters, find the best cutter that can traverse the entire surface without interference". Cutter selection can be considered as a two-phase decision-making process. The first task is to determine those accessible cutters, from the cutter set, that can finish the entire surface. The second task is to choose the best cutter from the accessible cutters according to some optimization criteria. This chapter focuses on solving the problems in the first task, which is essential to check whether a given cutter can finish the entire surface. The second task is left to the next chapter. To be more specific, if the cutter is accessible to any point on the surface, the cutter is said to be able to finish the entire surface. In this work, a two-phase approach is proposed to check the accessibility of the cutter. In phase-I, the part surface is firstly sampled to obtain a set of points that represent the surface. These points are then divided into two groups, the interference-prone and interference-free, by analyzing the geometric

properties of the points and the cutter. In phase-II, a searching algorithm is developed to check the accessibility of a cutter to each interference-prone point based on the cutter accessibility (CA) algorithm described in Chapter 2.

3.2 Related Works

Research focusing on automatic cutter selection in 5-axis finishing is limited. 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, the radius of curvature is larger than the effective cutting radius. Jensen *et al.* (2002) developed a cutter selection algorithm for fillet-end cutters based on curvature matching machining, in which local-gouging, rear-gouging, and global-collision are considered. The algorithm is trial-and-error in nature. It starts with the largest cutter in a tool database. Beginning from 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 approach.

In machining process planning, cutter selection is generally carried out before tool-path generation. At the stage of cutter selection, the tool-path pattern is preferably not fixed, and a more general sampling method (rather than CC-point generation) is needed. Moreover, since it is impossible to check every point on the

given surface, an intelligent approach is needed to decompose the entire surface into *interference-prone* regions where gouging and collision is likely to happen and *interference-free* regions where it is not. The accessibility checking will be carried out only within the interference-prone regions to shorten the computation time.

3.3 Surface Decomposition for Cutter Accessibility Analysis

NURBS (Non-Uniform Rational B-Spline) representation is widely used for sculptured surfaces in industry (Piegl and Richard, 1995). In this work, a machined surface is described by a set of NURBS patches $\mathbf{S}_i(u,v)$ with C^2 continuity. A fillet-end cutter is described by its major radius (R), minor radius (r_f), and length (L). Since a sculptured surface can be represented as a set of surface patches that are trimmed by one or more curves, the following discussion will focus on a single NURBS patch without losing generality.

Surface decomposition has been employed to reduce computation time in multi-axis milling planning. Elber (1994, 1995) showed an interesting approach for subdividing a surface into 3-axis finish regions and 5-axis finish regions to improve the productivity based on the NURBS surface curvature evaluation. Ding *et al*. (2001) presented a similar method to decompose a surface into interference and non-interference regions for the ball-end tool accessibility evaluation in 3-axis finishing. In this work, a novel approach is developed to divide a surface into interference-prone regions and interference-free regions for 5-axis machining based on the corresponding geometry of both cutter and part surface.

3.3.1 Local surface geometric property

Surface properties, such as unit normal vector and curvature, are well defined in the literature (O'Neil, 1966). For a specific point on the surface patch **S**(u,v), the normal curvature is the curvature of an intersection curve between the surface and the plane containing the surface normal vector at the point. There exist the maximum ($\kappa_{\max}$) and minimum ($\kappa_{\min}$) normal curvature values, called the principal curvatures. This kind of curvature property can be represented with two variables: Gaussian curvature (K) and Mean curvatures (H):

$$K = \kappa_{\max} \times \kappa_{\min} = \frac{LN - M^2}{EG - F^2} \tag{3.1}$$

$$H = \frac{1}{2}(\kappa_{\max} + \kappa_{\min}) = \frac{EN + GL - 2FM}{2(EG - F^2)} \tag{3.2}$$

Where E, F, G and L, M, N are the coefficients of the first and second fundamental forms at the specific point, respectively. Their values can be calculated as (Farin, 1996):

$$\begin{array}{lll} E = \boldsymbol{S}_u \bullet \boldsymbol{S}_u & F = \boldsymbol{S}_u \bullet \boldsymbol{S}_v & G = \boldsymbol{S}_v \bullet \boldsymbol{S}_v \\ L = \boldsymbol{n} \bullet \boldsymbol{S}_{uu} & M = \boldsymbol{n} \bullet \boldsymbol{S}_{uv} & N = \boldsymbol{n} \bullet \boldsymbol{S}_{vv} \end{array} \tag{3.3}$$

Where $\boldsymbol{n}$ is the normal vector, $\boldsymbol{S}_u$ and $\boldsymbol{S}_v$ are the first-order partial derivatives, and $\boldsymbol{S}_{uu}$, $\boldsymbol{S}_{uv}$ and $\boldsymbol{S}_{vv}$ are the second-order partial derivatives at the point, respectively. Based on the values of K and H, the local surface shape around the point can be divided into three categories: convex, concave and saddle (O'Neil, 1966):

(1) $K \geq 0$ and $H \leq 0$: κ_{max} and κ_{min} are smaller than or equal to zero, local surface is convex.

(2) $K \geq 0$ and $H > 0$: κ_{max} and κ_{min} are greater than or equal to zero, local surface is concave.

(3) $K < 0$: κ_{max} and κ_{min} have different signs, local surface is saddle shape.

In theory, a surface $\mathbf{S}(u,v)$ can be decomposed by the curves on which $K(u,v)=0$. However, $K(u,v)$ is a high order expression of u and v even for a low order NURBS surface patch. Therefore, it is very difficult to solve the above equation analytically. For implementation, a numerical method seems more feasible. In this research, the grid-based methodology proposed by Smith and Farouki (2001) is adopted to search for the boundary curves among different shape regions. First, the surface patch is uniformly sampled in u and v directions to obtain a set of grid points. At each grid point, the Gaussian and Mean curvatures are calculated. Next, the points with concave and saddle local property are identified. The neighboring concave and saddle points are grouped together to form concave regions and saddle regions, respectively. The remaining points form convex regions. The boundary points are linked up to form their respective boundaries. The computation complexity of this algorithm is $O(n)$, where n is the number of sampled points on the surface.

It is worth mentioning that in this work, a uniform sampling approach is employed for easy implementation. In practice, adaptive sampling strategy in terms of rate of change in curvature is preferable. Moreover, the approximation error also needs to be considered. It is noted that adaptive sampling methods of NURBS surfaces have received much research efforts and some tessellation techniques have been well established (Piegl and Richard, 1995; Austin *et al.*, 1997; Piegl and Tiller,

1998), which could be adopted here to reduce the computation time while providing a far superior characterization to the surface.

Up to this point, the surface subdivision based on local surface shape is completed. The points within the concave regions and saddle regions are interference-prone. For points within the convex regions, local-gouging can be effectively ruled out. However, rear-gouging and global-collision may still occur, especially at the points that are close to the boundary. An interesting observation is that, starting from the centre of a convex region, the closer a point is to the boundary, the more interference-prone (in particular rear-gouging) the point is. It is therefore possible to identify a portion of the convex region that is free of rear-gouging and global-collision, thus further reducing the checking area.

3.3.2 Identifying the interference-free area from a convex region

As shown in Figure 3.1a, when a fillet-end cutter is positioned at a point on a surface, a point on the filleted portion (cutting edge) is in contact with the point. This point is called the cutter-contact (CC) point. Rear-gouging occurs if a point on the cutter bottom surface is underneath the part surface. Global-collision occurs if the distance between the cutter axis and the surface is less than R within the range of cutter length (L). Hence, the detection of rear-gouging and global-collision is in fact a distance-evaluation problem. However, a numerical method is the only solution to solve this problem, which is very time consuming. At the same time, since the feeding direction is not fixed, the cutter could approach the point from any direction. If we position the cutter along a fixed direction, say the normal direction of the point on the surface, the CC point is at the edge of the cutter bottom circle (see Figure 3.1a). We can see that, from the top and along the axis, the area covered by the cutter in all

possible positions forms a circle with a radius of $2R\text{-}r_f$ (see Figure 3.1b and 3.1c). The envelope surface of the cutter in all possible positions effectively forms a cylinder with a radius of $2R\text{-}r_f$ and length of L. We call this cylinder a *dummy flat-end cutter* of the fillet-end cutter. It can be seen that the volume occupied by the fillet-end cutter at all possible feeding directions towards the point is inside the volume occupied by its dummy flat-end cutter. Therefore, we can check the accessibility of a fillet-end cutter (R, r_f, L) at a point along the normal direction of the point by checking the accessibility of a flat-end cutter ($2R\text{-}r_f$, L) with its bottom centre at the point and its axis along the normal direction of the point. If the flat-end cutter does not have any interference at the point, the fillet-end cutter does not either. By using the dummy flat-end cutter, we effectively simplify the accessibility checking problem for a fillet-end cutter. It is worth mentioning that this simplification process uses a tighter criterion to check the accessibility of a point to a cutter by positioning the cutter along the normal direction of the point only. Although some interference-free points in the convex region may be treated as interference-prone, it will not have any negative effect at the later stage since we are only interested in identifying the interference-free regions quickly at this stage.

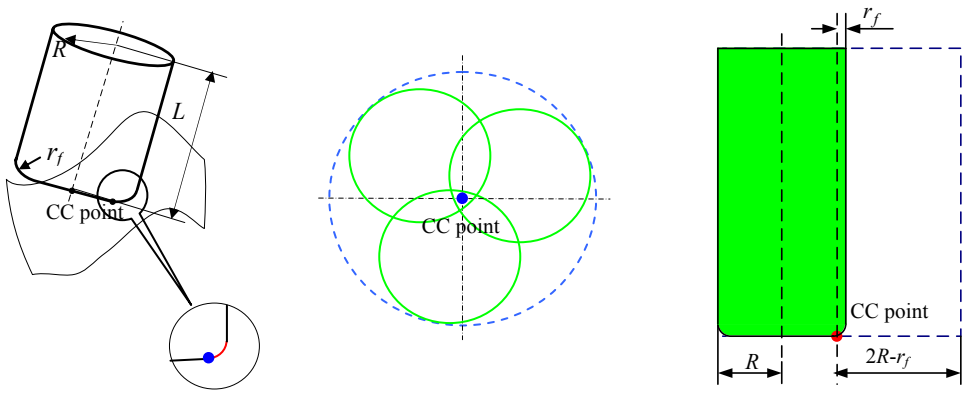


Figure 3.1 A fillet-end cutter and its dummy flat-end cutter

Now, we proceed to identifying the interference-free portion of a convex region by using the dummy flat-end cutter. A general case for this problem is presented in Figure 3.2a, where *r* is a single convex region to be checked on the part surface *S*. *e* represents its boundary and *Xr* all the other regions on *S* except *r*.

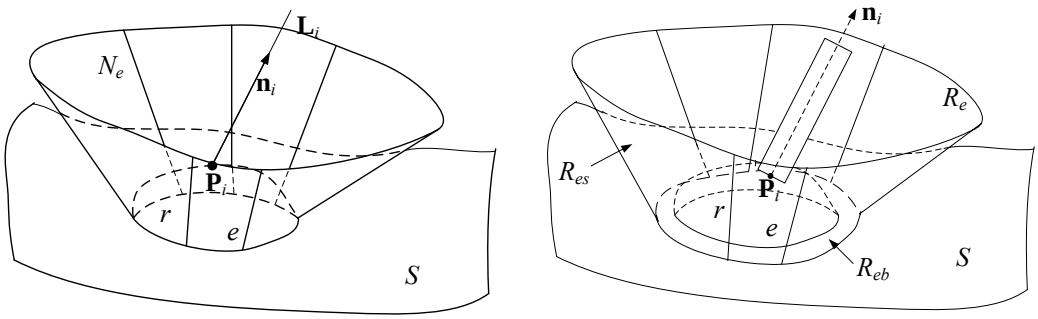


Figure 3.2 A convex region *r* on the part surface *S* and some geometric properties

Lemma 1: Let N_e be the ruled surface formed by moving a ray (in the surface normal direction) along *e*. For any point $\mathbf{P}_i$ on *r* ($\mathbf{P}_i \notin e$), a ray $\mathbf{L}_i$ in the direction of its surface normal vector $\mathbf{n}_i$ has no intersection with N_e.

Proof:

A Gaussian map (G-map) with respect to a surface patch refers to the mapping of normal directions onto a unit sphere, which characterizes the variation range of the surface normal vectors. Smith and Farouki (2001) pointed out that, for a complex surface, boundaries of G-map correspond to the loci of vanishing Gaussian curvature, namely $K = 0$, and the surface boundary. For a single convex region *r* on *S*, the value of Gaussian curvature at any point within *r* is non-negative and any point on *e* has vanishing Gaussian curvature. Thus, the G-map for *r* is a closed range and for any point $\mathbf{P}_i$ on *r* ($\mathbf{P}_i \notin e$), its normal vector $\mathbf{n}_i$ corresponds to a point within the G-map.

Therefore, a ray $\mathbf{L}_i$ (starting from $\mathbf{P}_i$) in the direction of its surface normal vector $\mathbf{n}_i$ has no intersection with N_e. Otherwise, $\mathbf{n}_i$ would have had corresponded to a point outside the G-map. The lemma thus is proved.

Theorem 1: Given any point on *r*, a flat-end cutter is positioned with its bottom centre at the point and its axis along the surface normal direction. If the cutter is interference-free (rear-gouging and global-collision) with *S* at any point on *e*, the cutter is interference-free with *S* at any point on *r*.

Proof:

Referring to Figure 3.2b, R_e is a surface defined as $R_e = r \cup R_{es} \cup R_{eb}$, where R_{es} is the envelope surface formed by the cutter's shaft undergoing the sweeping movement (the cutter axis along the normal direction of the point and the bottom centre at the point) along *e* and R_{eb} is formed by the cutter bottom surface undergoing the sweeping movement along *e* and then trimmed by *e*. Since the cutter is interference-free with *S* at any point on *e*, R_e will have no intersection with any of the remaining surface patches, namely *Xr*. When the cutter is placed at any point within *r*, according to Lemma 1, the cutter will not have any intersection with R_e. Therefore, the cutter will not have any intersection with any of the remaining surface patches, namely *Xr*. Furthermore, since the cutter is positioned with its axis along the normal direction of the point on *r*, and *r* is convex, the cutter will not have any interference with *r*. Therefore, at any point on *r*, the cutter is interference-free with *S*.

The above theorem can simplify the checking procedure for flat-end cutters by converting the surface checking problem into a curve checking problem. The task of identifying the interference-free portion from a convex region is converted to

identifying the largest boundary (from the convex region) on which the dummy flat-end cutter is interference-free. The detailed algorithm is given as follows:

Algorithm: ***Identifying the interference-free regions on a surface***

Input: (a) *A part surface represented by a set of trimmed NURBS patches S_i with C^2 continuity.*

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

Output: *A set of interference-free points $\{S_{i\text{-}free}\}$ and a set of interference-prone points $\{S_{i\text{-}prone}\}$*

Begin

(1) Sample the trimmed surfaces to a set of grid points.

(2) Identify the convex/concave/saddle regions, each with a set of points and its boundary. Put the concave and saddle points into $\{S_{i\text{-}prone}\}$.

(3) Pick a convex region r and place its boundary points into a point set $\{\mathbf{P}_{ei}\}$ and other points in $\{\mathbf{P}_i\}$. Xr represents all the other surface regions except r. Create an empty boundary point set $\{\mathbf{P}_{ei\text{-}free}\}$ for its interference-free portion. Pick one point from the point set $\{\mathbf{P}_{ei}\}$ as the *current point*.

(4) Place the dummy flat-end cutter ($2R$-r_f, L) with its bottom centre at the *current point* and its axis in the normal direction of the point. (a) check whether the cutter's posture is within the machine axis limits, (b) calculate the distance between the cutter bottom plane and the sampled points on Xr for gouging detection, and (c) calculate the distance between the cutter axis and the sampled points on Xr for collision detection.

(5) If this point is free of interference, place it into $\{\mathbf{P}_{ei\text{-}free}\}$, and find the next

point in $\{\mathbf{P}_{ei}\}$ as the *current point*, go back to (4). Otherwise, place the point into $\{S_{i\text{-}prone}\}$ and find the nearest point of the *current point* in $\{\mathbf{P}_i\}$ and set it as the *current point*, go back to (4). If no more points are left in $\{\mathbf{P}_{ei}\}$, put $\{\mathbf{P}_{ei\text{-}free}\}$ and the remaining points in $\{\mathbf{P}_i\}$ into $\{S_{i\text{-}free}\}$. Go to (6).

(6) If all the convex regions are traced, stop. Otherwise, Go back to (3)

End

As mentioned above, the computation complexity in the step (1) is O(n), where n is the number of checked points. For other steps, the complexity is O(mn_1), where m is the number of sampled points for gouging and collision checking, and n_1 is the number of points in the convex regions. Thus, the computation complexity of this surface decomposition algorithm is O($n + mn_1$).

3.4 The Overall Algorithm for Cutter Accessibility to a Surface

For a sculptured surface, the algorithm introduced in Section 3.3 can be used to find the interference-prone and interference-free regions. Only the points in the interference-prone regions are involved for the analysis of cutter accessibility to the surface, by employing the CA algorithm proposed in Chapter 2. If an accessibility map at each point in the interference-prone regions is available, the cutter is accessible to the surface. The algorithm is described as follows:

Algorithm: ***Analyzing accessibility of a cutter to a surface***

Input: (a) *A part surface represented by a set of trimmed NURBS patches S_i with C^2 continuity*

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

Output: *The accessibility of the cutter to the surface*

Begin

(1) Apply the surface decomposition algorithm to divide the surface into a set of interference-free points $\{S_{i\text{-}free}\}$ and a set of interference-prone points $\{S_{i\text{-}prone}\}$, using the algorithm in Section 3.3.

(2) Pick one point from $\{S_{i\text{-}prone}\}$ and name it the *current point*.

(3) Apply the CA algorithm to find the accessibility map of the cutter at the current point, as that introduced in Section 2.3.4.

(4) If accessibility map is available at the current point, find the next point in $\{S_{i\text{-}prone}\}$ and name it the current point; go back to (3). Otherwise, go to (6). If no more points are left in $\{S_{i\text{-}prone}\}$, go to (5).

(5) Output "The cutter is accessible to the surface". Stop.

(6) Output "This cutter is not accessible to the surface". Stop.

End

The computation complexity from step (2) to (4) is O(kmn_2), in which m is the number of sampled points for interference checking, k is the total number of discrete θs sampled over the rotational angle range $[\theta_{\min}, \theta_{\max}]$, and n_2 is the number of checked point for cutter accessibility in interference-prone regions. Incorporated with the complexity of step (1) as discussed above, the complexity of this algorithm is O($mn_1 + n + kmn_2$) = O(kmn_2). Here, n, n_1 and n_2 are generally on the same level in computation complexity. If the CA algorithm is directly employed in each checked point on the surface, the computation complexity is O(kmn), where n is the total number of checked points. From experiment, n_2 is much smaller than n for many

sculptured surfaces. Thus, the computation load can be greatly alleviated by this surface decomposition technique.

3.5 Summary

This chapter addresses one of the important issues in cutter selection for 5-axis sculptured surface machining (finish cut), the accessibility of a given cutter to the irregular free-form surface. In the last chapter, the CA algorithm has been developed to check the accessibility of a cutter to a surface point. This point-based solution can be extended to cover the whole surface for cutter accessibility to the surface. However, heavy computation load is a concern of this approach since the CA algorithm is numerical in general. In this chapter, an intelligent approach has been proposed to decompose the whole surface into interference-prone regions where gouging and collision is likely to happen and interference-free regions where it is not. The accessibility checking will be carried out only within the interference-prone regions. Integrated with the CA algorithm, a more efficient algorithm has been developed to analyze the accessibility of a cutter to a surface. The complexity analysis shows that the computation time is greatly reduced with this method compared to that of direct checking on all surface points.

CHAPTER 4

CUTTER SELECTION PART 2:

ACCESSIBILITY COMPARISON BETWEEN CUTTERS

In 5-axis milling of sculptured surfaces, a cutter's accessibility refers to a feasible A-map, in terms of tilting and rotational angles, along which the cutter does not cause any interference at a point on the surface. In Chapter 2, a point-based algorithm has been developed that is able to find a cutter's accessibility at a point on the surface. To check whether a cutter can finish a given sculptured surface, it is essential to find the cutter's accessibility at all the sampled points on the surface. To find the optimal cutter, repeated checking on the same set of sampled points is needed for all the available cutters. In this chapter, we will do an accessibility comparison study between cutters of different dimensions by considering cutters' geometric characteristics. Based on the findings, when the accessibility of a larger cutter is available, the accessibility of a smaller cutter may be obtained without running the time-consuming point-based algorithm. Combined with the surface decomposition technique proposed in Chapter 3, an efficient algorithm is subsequently developed for finding the optimal cutter while avoiding the redundant checking at some of the sampled points.

4.1 Introduction

In 5-axis milling, the task of cutter selection in the finish cut is to determine the best cutter from the available ones that can traverse the entire surface without

interference. The optimization criterion usually refers to the maximum material removal rate and the interference refers to local-gouging, rear-gouging and global-collision. At a point on the part surface, if a cutter has a posture that causes no interference, the cutter is said to be accessible to this point. If the cutter is accessible to any point on the surface, the cutter is said to be accessible to the part surface. Therefore, the key issue in cutter selection is to check the accessibility of a cutter over the entire surface.

In Chapter 2, a point-based method is developed to check the cutter's accessibility to a point for fillet-end cutter without the consideration of feeding direction. By considering the geometric properties of the cutter and the underlying surface, three algorithms have been developed to find the accessible posture range (APR) along which the cutter is free of local-gouging, rear-gouging, and global-collision, respectively at a point. The intersection of the three respective APRs and the machine axis limits at the point represents the A-map of the cutter at that point. Applying this method to the whole surface, i.e., all the sampled points of the surface, one can find whether a cutter is accessible to the surface. Therefore, the cutter selection task can be effectively performed by repeating this procedure to all the cutters.

In automated process planning for 5-axis milling, computational efficiency is often a major concern since the point-based algorithm is numerical in general. As we mentioned in Chapter 2, for the point-based CA algorithm, the computation complexity is O(km) for a cutter, in which m is the number of sampled points and k is the total number of discrete θs sampled over the rotational angle range [$\theta_{\min}$, $\theta_{\max}$]. To check whether a cutter is accessible to a surface, the computation complexity is O(kmn), in which n denotes the number of points to be checked on the surface.

Therefore, repeating this search procedure to find the accessibility of all the available cutters will lead to very heavy computation load, especially for large-size and/or complex surfaces. One way to reduce the computation load is to minimize *n* for a specific cutter (*m* is directly related to the sampling resolution). In Chapter 3, certain success in this direction has been achieved by dividing the part surface into interference-prone and interference-free regions. The other way is to reduce the redundancy when the search procedure is applied from one cutter to the other. Intuitively, a cutter with a smaller major radius tends to have greater accessibility at a point than a cutter with a larger major radius. This rule of thumb is true when both cutters are of flat-end. Under this circumstance, accessibility search for the larger cutter first produces a set of accessible points and a set of non-accessible points. When a smaller cutter is considered, it is accessible to the accessible points, i.e., search at these accessible points can be skipped. Essentially, the computational redundancy is avoided based on accessibility comparison between cutters. In the case of flat-end cutters, accessibility comparison can be conducted in a straightforward manner. However, the same cannot be said for fillet-end cutters with different dimensions.

In this chapter, the accessibility of two fillet-end cutters of different dimensions is compared based on their geometric characteristics. The problem can be defined as "given two cutters A and B (A is larger than B), at an accessible point to cutter-A, determine the A-map of cutter-B from the A-map of cutter-A". Based on the study, a detection-and-correction approach is proposed to solve this problem. In the first phase, interference of cutter-B with the part surface, if any, is detected within the A-map of cutter-A. In the second phase, the posture of cutter-B is corrected to find its A-map, if any. In this way, the A-map of cutter-B at some of the accessible points of

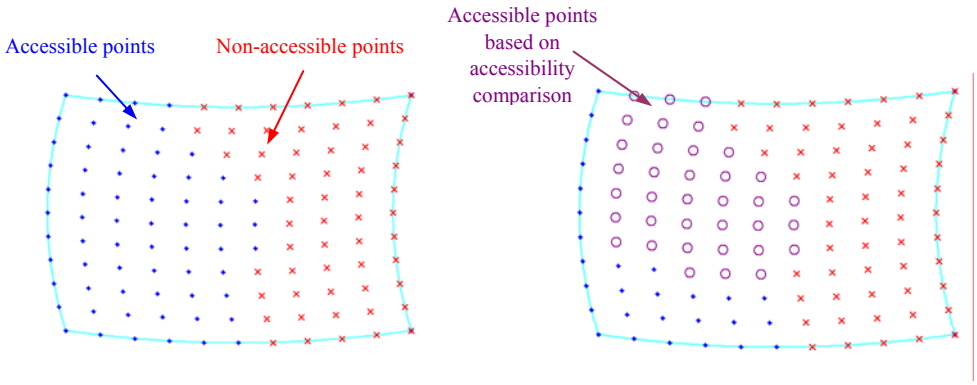
cutter-A can be found by some quick estimation methods. Based on this accessibility comparison method, a non-redundant search algorithm is developed to find the optimal cutter for machining a given surface. The general procedure is briefly described here. The cutters are firstly ranked from large to small according to their dimensions. The algorithm starts by picking the largest cutter, T_{top}, the A-map at every sampled point is obtained and the sampled points are categorized into accessible and non-accessible. Subsequently, the next cutter, T_{next} is picked for checking. At the accessible points to T_{top}, the accessibility comparison algorithm is run to find the A-map for T_{next}. At the non-accessible points to T_{top}, the CA algorithm is run against T_{next} to find it's A-map. This process is repeated to the next cutter until a cutter is found to be accessible to all the sampled points.

4.2 Accessibility Comparison between Cutters

In the machining of sculptured surfaces, a cylindrical cutter with larger R is generally preferable to one with smaller R if both are accessible to the surface. This is because intuitively, a larger cutter will have a higher material removal rate and better surface finish. Therefore, the task of cutter selection is to find the largest cutter, from the available ones, that is accessible to the surface. In this work, the available cutters are firstly arranged in a descending order of R. Cutters with the same R are arranged in an ascending order of r_f. Based on the observation that a flat-end cutter is more efficient than a ball-end cutter with the same R, it is considered for two cutters of the same R, the one with smaller r_f is more efficient. An example list of ranked cutters is shown in Table 4.1.

Table 4.1: A list of fillet-end cutters in large-to-small order

Cutters	Major radius R (mm)	Minor radius r_f (mm)					
T_1 to T_6	12	0.5	1	1.5	2	2.5	3
T_7 to T_{12}	10	0.5	1	1.5	2	2.5	3
T_{13} to T_{18}	8	0.5	1	1.5	2	2.5	3
T_{19} to T_{22}	6	0.5	1	1.5	2		



(a) Accessible points and non-accessible points for T_{large}

(b) Some accessible points for T_{small} found by the quick estimation algorithm based on accessibility comparison

Figure 4.1 Accessible points of a larger cutter and a smaller one

The optimal cutter selection process can follow a simple trial-and-error approach by picking a cutter top-down from the cutter list and applying the CA algorithm at a point to all the sampled points in the interference-prone regions until an accessible cutter is found. That means that the CA algorithm has to be applied to all the sampled points against all the non-accessible cutters in the list. Some redundancy has been observed in this simple trial-and-error approach. For example, after the accessibility of largest cutter (T_{large}) in the list is checked, the sampled points are divided into two groups: the accessible ones $\{\mathbf{P}_a \mid T_{large}\}$ and non-accessible ones $\{\mathbf{P}_{na} \mid T_{large}\}$ as shown in Figure 4.1a. For the next cutter (T_{small}) in the list, the CA algorithm must be run at $\{\mathbf{P}_{na} \mid T_{large}\}$. At $\{\mathbf{P}_a \mid T_{large}\}$, it is, however, desirable to

obtain the accessibility of T_{small} without running the CA algorithm (see Figure 4.1b). To be more general, this is an accessibility comparison problem between two cutters, which is addressed in the following subsections.

4.2.1 Problem definition for accessibility comparison

The accessibility comparison problem is defined as "given two cutters T^L (R^L, r_f^L) and T^S (R^S, r_f^S) where $R^L \geq R^S$, T^L has an APR of (θ,(λ_{min}, λ_{max})) to a point $\mathbf{P}_c$ on a sculptured surface with C^2 continuity, find the APR of T^S within (θ,(λ_{min}, λ_{max})) to $\mathbf{P}_c$". We have developed some quick estimation algorithms to solve this problem for the following four scenarios:

(1) $R^S = R^L$ and $r_f^S > r_f^L$

(2) $R^S < R^L$ and $r_f^S = r_f^L$

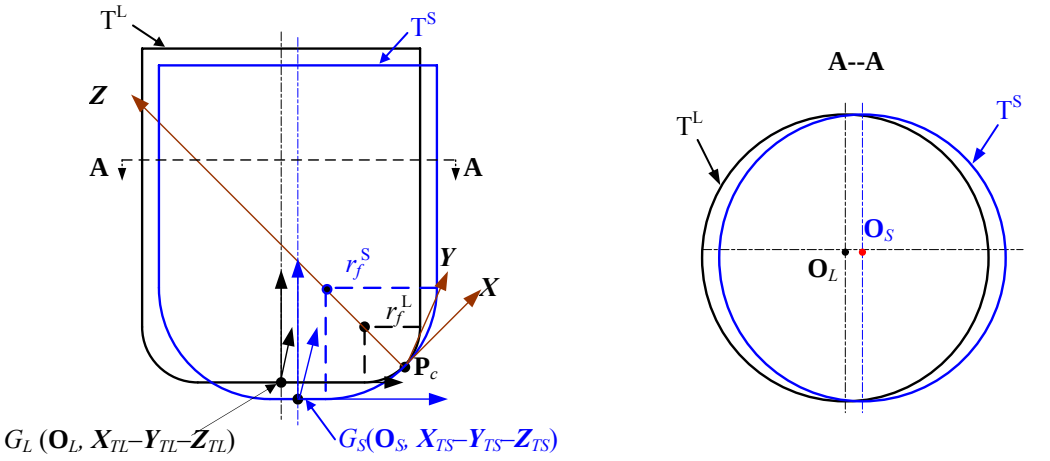
(3) $R^S < R^L$ and $r_f^S > r_f^L$

(4) $R^S < R^L$ and $r_f^S < r_f^L$

Among the four interference attributes considered, the posture range for machine limits (ML) is related to the machine configuration and the surface geometry at $\mathbf{P}_c$ but not the cutter geometry. Hence, the ML posture range of T^S is the same as that of T^L. On the other hand, the local-gouging (LG) posture range for T^S can be found by a relatively simple algorithm with a complexity of O(k) as mentioned earlier (Section 2.3.1). Therefore, in the following discussion, accessibility comparison between T^L and T^S is limited to rear-gouging (RG) and global-collision (GC) only.

4.2.2 $R^S = R^L$ and $r_f^S > r_f^L$

Given a θ within the machine axis limits, T^L is accessible to $\mathbf{P}_c$ with a range of (θ, (λ_{min}, λ_{max})). This means that for any λ ($\lambda_{min} \le \lambda \le \lambda_{max}$), T^L is interference-free. However, it is found that there is no guarantee that T^S is interference-free at (θ, λ). This is shown as followings.



(a) T^L and T^S with the same posture (b) Cross-section view of T^L and T^S

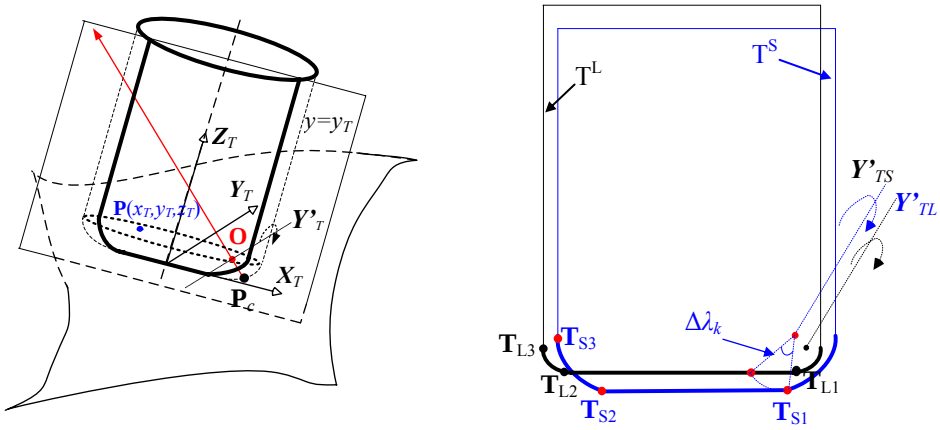
Figure 4.2 T^L and T^S ($R^S = R^L$, $r_f^S > r_f^L$) with the same posture

Figure 4.2a shows T^L and T^S ($R^S = R^L$, $r_f^S > r_f^L$) positioned with the same posture (θ, λ) at $\mathbf{P}_c$. The local frame and tool frame for both cutters are defined as those in Section 2.3. The local frame H ($\mathbf{P}_c$, $\boldsymbol{X}$–$\boldsymbol{Y}$–$\boldsymbol{Z}$) is the same for T^L and T^S, while the tool frame for T^L and T^S are G_L ($\mathbf{O}_L$, $\boldsymbol{X}_{TL}$–$\boldsymbol{Y}_{TL}$–$\boldsymbol{Z}_{TL}$) and G_S ($\mathbf{O}_S$, $\boldsymbol{X}_{TS}$–$\boldsymbol{Y}_{TS}$–$\boldsymbol{Z}_{TS}$) respectively. G_L and G_S share identical axis directions but different origins. Vector $\mathbf{O}_L\mathbf{O}_S$ can be represented in frame G_L as:

$$\mathbf{V}_{\mathbf{O}_L\mathbf{O}_S} = \begin{pmatrix} dx \\ dy \\ dz \\ 1 \end{pmatrix} = {}^{G_L}\mathbf{P}_{\mathbf{O}_S} = {}^{G_L}\mathbf{T}_H \bullet {}^{H}\mathbf{T}_{G_S} \bullet \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}_{G_S} = \begin{pmatrix} (R^{\mathrm{L}} - R^{\mathrm{S}}) + (r_f{}^{\mathrm{S}} - r_f{}^{\mathrm{L}})(1-\sin\lambda) \\ 0 \\ (r_f{}^{\mathrm{L}} - r_f{}^{\mathrm{S}})(1-\cos\lambda) \\ 1 \end{pmatrix} \quad (4.1)$$

Where ${}^{G_L}\mathbf{T}_H$ is the homogenous transformation matrix from H to G_L, and ${}^{H}\mathbf{T}_{G_S}$ from G_S to H. $[0, 0, 0, 1]^T$ is the homogeneous coordinates of point $\mathbf{O}_S$ represented in G_S. It is clear that $dx \geq 0$, which indicates that $\mathbf{O}_S$ locates at the right of $\mathbf{O}_L$ as shown in Figure 4.2b. Therefore, the front cylindrical surface of T^S is beyond that of T^L, which may result in global-collision between T^S and the part surface. Similarly, $dz \leq 0$, which indicates that $\mathbf{O}_S$ locates below $\mathbf{O}_L$. The bottom plane of T^S may result in rear-gouging. In conclusion, there is no guarantee that T^S is interference-free at (θ, λ). Hence, it is needed to find the RG and GC maps for T^S.

Figure 4.3a shows the geometry of a cutter (R, r_f) that is positioned at $\mathbf{P}_c$ on the part surface. With a different λ, the cutter will have a different contact point on the filleted portion and the pivot point $\mathbf{O}$ is along the normal vector of $\mathbf{P}_c$ and with a distance r_f from $\mathbf{P}_c$. When only λ is changed, the cutter rotates about $\boldsymbol{Y'}_T$-axis (parallel to $\boldsymbol{Y}_T$-axis and passes through the pivot point $\mathbf{O}$), which means that y-coordinate of any point on the cutter surface is constant in the tool frame during inclination. Based on this geometric property, the cutter can be sliced into many sections normal to the $\boldsymbol{Y}_T$-axis (see Figure 4.3a). If we can find the accessible range of λ for every section, the intersection of the A-maps of λ for all the sections can be used to approximate the A-map of λ for the cutter. In this way, the 3D problem is converted to a 2D problem. The following discussion will focus only on a section of the cutter at $y = y_{Tk}$ ($-R \leq y_{Tk} \leq R$).



(a) The geometries of a cutter at $\mathbf{P}_c$ (b) Sections at $y=y_{Tk}$ for T^L and T^S ($\lambda = \lambda_{min}$)

Figure 4.3 Finding the RG A-map for T^S using a 2D method

4.2.2.1 RG posture range for T^S

Figure 4.3b shows two sections of T^L and T^S ($\lambda = \lambda_{min}$) at $y = y_{Tk}$ ($-R^L \le y_{Tk} \le R^L$), respectively. The cutter bottom curves $\mathbf{T}_{L1}$-$\mathbf{T}_{L2}$-T_{L3} (for T^L) and $\mathbf{T}_{S1}$-$\mathbf{T}_{S2}$-T_{S3} (for T^S) are considered for rear-gouging (the front filleted portions are excluded since local-gouging free is assumed). When $\lambda = \lambda_{min}$, it is known that line $\mathbf{T}_{S1}\mathbf{T}_{S2}$ is below $\mathbf{T}_{L1}\mathbf{T}_{L2}$, and it can be further proved that $\mathbf{T}_{S3}$ is above curve $\mathbf{T}_{L1}$-$\mathbf{T}_{L2}$-$\mathbf{T}_{L3}$. Although $\mathbf{T}_{L1}$-$\mathbf{T}_{L2}$-T_{L3} is free of rear-gouging, we cannot rule out the interference between $\mathbf{T}_{S1}$-$\mathbf{T}_{S2}$-T_{S3} and the part surface. To make sure that $\mathbf{T}_{S1}$-$\mathbf{T}_{S2}$-T_{S3} is free of rear-gouging, we need to rotate T^S section clockwise about $\boldsymbol{Y'}_{TS}$ by $\Delta\lambda_k$ such that $\mathbf{T}_{S1}$-$\mathbf{T}_{S2}$-T_{S3} is on or above $\mathbf{T}_{L1}$-$\mathbf{T}_{L2}$-$\mathbf{T}_{L3}$. The RG posture range for the T^S section is therefore ($\lambda_{min}+\Delta\lambda_k$, λ_{max}). Since $\mathbf{T}_{S1}$ is the closest point to the pivot point, when $\mathbf{T}_{S1}$ falls on $\mathbf{T}_{L1}$-$\mathbf{T}_{L2}$, all the other points on $\mathbf{T}_{S1}$-$\mathbf{T}_{S2}$-T_{S3} are above $\mathbf{T}_{L1}$-$\mathbf{T}_{L2}$-$\mathbf{T}_{L3}$. $\Delta\lambda_k$ is the angle required for rotating $\mathbf{T}_{S1}$ to reach $\mathbf{T}_{L1}$-$\mathbf{T}_{L2}$, which can be calculated as,

$$\Delta\lambda_k = \cos^{-1}\left(\frac{r_f^{\,S} - |\,dz\,|}{d_k}\right) - \cos^{-1}\left(\frac{r_f^{\,S}}{d_k}\right) \tag{4.2}$$

Where d_k denotes the distance from $\mathbf{T}_{S1}$ to $\boldsymbol{Y'}_{TS}$-axis as $d_k = \sqrt{\left[(R^L - r_f^{\,S}) - \sqrt{(R^L - r_f^{\,S})^2 - y_{Tk}^{\,2}}\right]^2 + (r_f^{\,S})^2}$, and dz is from Eq. (4.1). To make sure that whole of T^S is free of rear-gouging, the minimum inclination angle is $\lambda_{rg} = \lambda_{\min} + \max(\Delta\lambda_k)$. The maximum value of $\Delta\lambda_k$ appears when d_k reaches its minimum, i.e., $d_k = r_f^{\,S}$ at $y_{Tk} = 0$. Therefore,

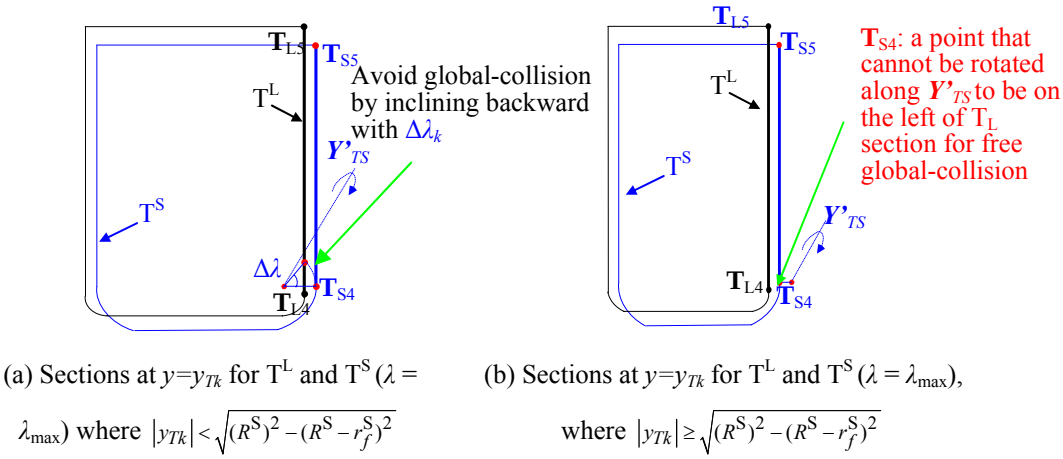
$$\lambda_{rg} = \lambda_{\min} + \cos^{-1}\left(\cos\lambda_{\min} + \frac{r_f^{\,L}}{r_f^{\,S}}(1 - \cos\lambda_{\min})\right) \tag{4.3}$$

The RG posture range at θ for T^S is (θ, (λ_{rg}, $\lambda_{\max}$)). This quick estimation algorithm has a complexity of O(k), where k is the total number of discrete θs. Compared with the CA algorithm for obtaining RG A-map in Section 2.3.2, which has a complexity of O(km), the computation load is significantly reduced.

4.2.2.2 GC posture range for T^S

From Eq. (4.1), it is clear that the front portion of T^S is beyond the boundary of T^L. Therefore, global-collision of T^S cannot be ruled out. By following the 2D approach, we found two scenarios as shown in Figure 4.4a and 4.4b, respectively. When $|y_{Tk}| < \sqrt{(R^S)^2 - (R^S - r_f^S)^2}$ (see Figure 4.4a), collision can be totally avoided by rotating T^S counter-clockwise. When $|y_{Tk}| \geq \sqrt{(R^S)^2 - (R^S - r_f^S)^2}$ (see Figure 4.4b), however, the same cannot be done. In other words, collision cannot be totally eliminated by adjusting λ only. Therefore, direct accessibility comparison for GC

avoidance between T^S and T^L cannot be done. Instead, the CA algorithm for global-collision is applied to T^S for its GC A-map (θ, (λ_{gc1}, λ_{gc2})).



(a) Sections at $y=y_{Tk}$ for T^L and T^S ($\lambda = \lambda_{max}$) where $|y_{Tk}| < \sqrt{(R^S)^2 - (R^S - r_f^S)^2}$

(b) Sections at $y=y_{Tk}$ for T^L and T^S ($\lambda = \lambda_{max}$), where $|y_{Tk}| \geq \sqrt{(R^S)^2 - (R^S - r_f^S)^2}$

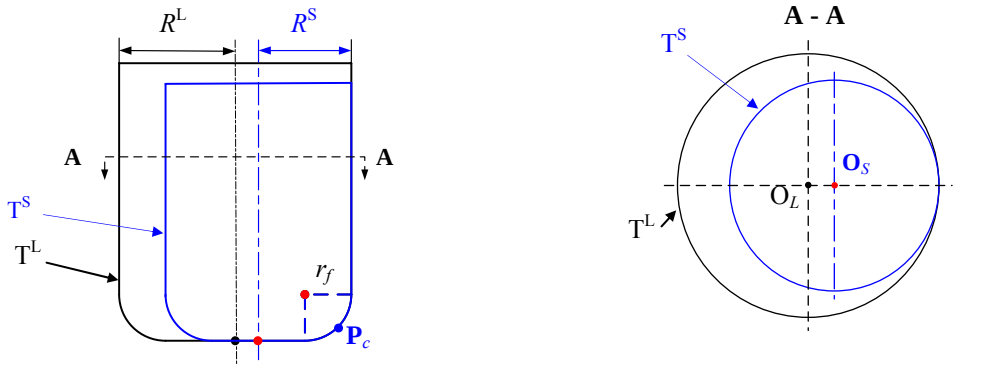
Figure 4.4 Finding the GC A-map for T^S using a 2D method

The posture range of T^S for both RG and GC at θ, is the intersection of (λ_{rg}, λ_{max}) and (λ_{gc1}, λ_{gc2}), which is a quick conservative estimate. The overall RG and GC posture range can be obtained by applying this method to all the discrete θs. In case when no posture range of RG and GC is found, the CA algorithm needs to be applied to T^S at this point.

4.2.3 $R^S < R^L$ and $r_f^S = r_f^L$

Given a θ within the machine axis limits, T^L is accessible to $\mathbf{P}_c$ with a range of (θ, (λ_{min}, λ_{max})). We found that T^S is also accessible to $\mathbf{P}_c$ at any (θ, λ) in (θ, (λ_{min}, λ_{max})). Figure 4.5 shows T^L and T^S ($R^S < R^L$, $r_f^S = r_f^L$) with the same posture (θ, λ) at $\mathbf{P}_c$. It can be easily proved that regardless of the height, the outer surface of T^S is totally contained in that of T^L, which can be clearly seen from Figure 4.5a and 4.5b. This means that T^S is also interference-free at (θ, λ). Therefore, at point $\mathbf{P}_c$, the A-map

of T^S can be taken as the same as that of T^L, and the accessibility search for T^S can be totally skipped.

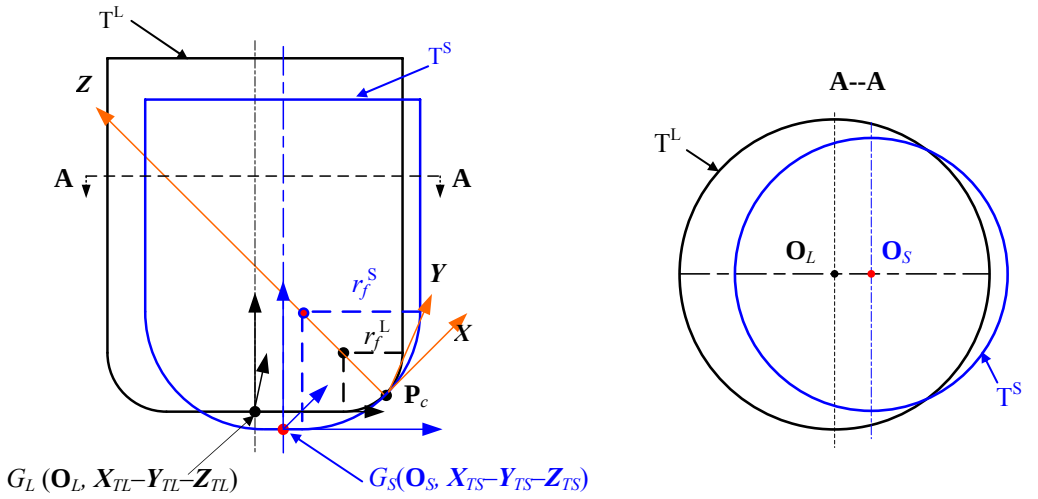


(a) Front view of T^L and T^S (b) Cross-section view of T^L and T^S

Figure 4.5 T^L and T^S ($R^S < R^L$, $r_f^S = r_f^L$) with the same posture

4.2.4 $R^S < R^L$ and $r_f^S > r_f^L$

Given a θ within the machine limits, T^L is accessible to $\mathbf{P}_c$ with a range of (θ, (λ_{min}, λ_{max})). Figure 4.6a shows T^L and T^S sharing the same posture (θ, λ) at $\mathbf{P}_c$ ($\lambda_{min} \leq \lambda \leq \lambda_{max}$). In frame G_L, the z-coordinate of $\mathbf{O}_S$ can be obtained from Eq. (4.1) as $z(\mathbf{O}_S) = -(r_f^S - r_f^L)(1 - \cos\lambda)$. Therefore, $z(\mathbf{O}_S) \leq 0$, which indicates that $\mathbf{O}_S$ locates below $\mathbf{O}_L$. The bottom plane of T^S may result in rear-gouging. On the other hand, it can be clearly seen from Figure 4.6b that the front shaft side of T^S is beyond the boundary of T^L. This means that global-collision between T^S and the part surface cannot be ruled out either. Hence, we need to find the RG and GC ranges for T^S. To obtain the RG posture range for T^S, we follow the same analytical procedure described in section 4.2.2. The final RG posture range of T^S at θ is (θ, (λ_{rg}, λ_{max})), where λ_{rg} can be calculated using Eq. (4.3).



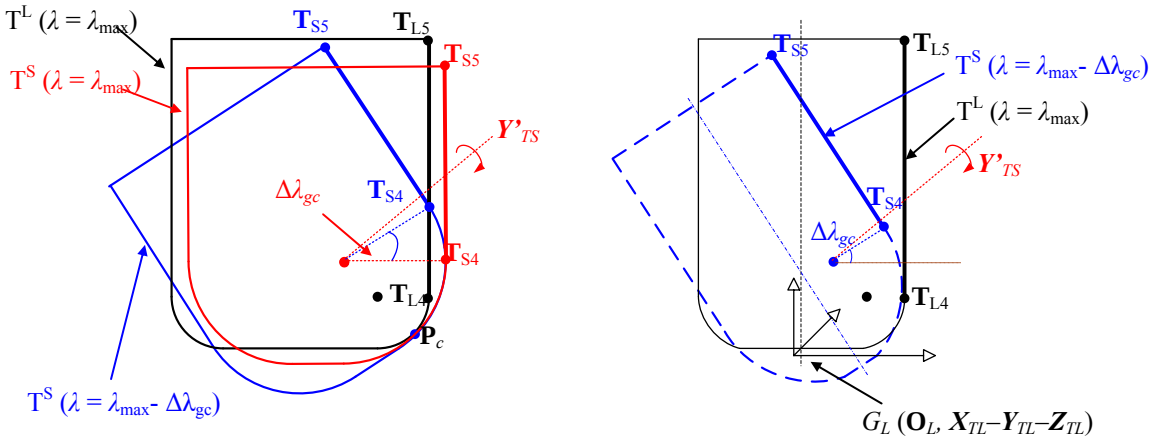
(a) T^L and T^S with the same posture (b) Cross-section view of T^L and T^S

Figure 4.6 T^L and T^S ($R^S < R^L$, $r_f^S > r_f^L$) with the same posture

To obtain the GC posture range for T^S, we only consider the front shaft surface of T^S (see Figure 4.6b). Fig 4.7a shows the sections ($y = 0$) of T^L and T^S at (θ, λ_{max}). Lines $\mathbf{T}_{L4}\mathbf{T}_{L5}$ and $\mathbf{T}_{S4}\mathbf{T}_{S5}$ represent the front shaft boundaries of T^L and T^S, respectively. Meanwhile, Figure 4.7a also shows a section of T^S after backward inclination by angle $\Delta\lambda_{gc}$ such that $\mathbf{T}_{S4}$ is on $\mathbf{T}_{L4}\mathbf{T}_{L5}$. We found that that the entire front shaft surface of T^S at (θ, $\lambda_{max} - \Delta\lambda_{gc}$) is within that of T^L at (θ, λ_{max}) if $R^L\cos(\Delta\lambda_{gc}) \geq R^S$ is satisfied. This is shown as followings.

Figure 4.7b shows sections of T^S ($\lambda = \lambda_{max} - \Delta\lambda_{gc}$) and T^L ($\lambda = \lambda_{max}$) at $y = y_{Tk}$ ($-R^S \leq y_{Tk} \leq R^S$), respectively. Lines $\mathbf{T}_{L4}\mathbf{T}_{L5}$ and $\mathbf{T}_{S4}\mathbf{T}_{S5}$ represent the front shaft boundaries of T^L and T^S, respectively. It can be seen that if $\mathbf{T}_{S4}$ is on or at the left of $\mathbf{T}_{L4}\mathbf{T}_{L5}$, $\mathbf{T}_{S4}\mathbf{T}_{S5}$ is free of collision with the part surface. The distance from $\mathbf{T}_{S4}$ to $\mathbf{T}_{L4}\mathbf{T}_{L5}$ is:

$$\varepsilon_k = \sqrt{(R^L)^2 - y_{Tk}^2} - \left[R^L - \left(R^S - \sqrt{(R^S)^2 - y_{Tk}^2} \right) \cos(\Delta\lambda_{gc}) \right] \tag{4.4}$$



(a) Sections at $y = 0$ for T^L and T^S ($\lambda = \lambda_{max}$), and T^S ($\lambda = \lambda_{max} - \Delta\lambda_{gc}$)

(b) Sections at $y = y_{Tk}$ for T^L ($\lambda = \lambda_{max}$) and T^S ($\lambda = \lambda_{max} - \Delta\lambda_{gc}$)

Figure 4.7 Finding the GC range for T^S using a 2D method

Since $\varepsilon_k \geq 0$ indicates that $\mathbf{T}_{S4}$ is on or at the left of $\mathbf{T}_{L4}\mathbf{T}_{L5}$, we need to find the condition that ensures $\varepsilon_k \geq 0$. From Eq. (4.4), we know that $\varepsilon_k = 0$ and $d\varepsilon_k / dy_{Tk} = 0$ at $y_{Tk} = 0$. Hence, to ensure $\varepsilon_k \geq 0$ for all y_{Tk}, one necessary condition is $d^2\varepsilon_k / d^2 y_{Tk} \geq 0$ at $y_{Tk} = 0$. This leads to $R^L \cos(\Delta\lambda_{gc}) \geq R^S$. At the same time, when $R^L \cos(\Delta\lambda_{gc}) \geq R^S$, it can be proved from Eq. (4.4) that $d\varepsilon_k / dy_{Tk} \geq 0$ for any $y_{Tk} > 0$ and hence $\varepsilon_k \geq 0$ for any y_{Tk} due to symmetric property. The value of $\Delta\lambda_{gc}$ is given as,

$$\Delta\lambda_{gc} = \cos^{-1}\left(\sin\lambda_{\max} + \frac{r_f^{\,L}}{r_f^{\,S}}\left(1 - \sin\lambda_{\max}\right)\right) \tag{4.5}$$

The GC posture range at θ for T^S is $(\theta, (\lambda_{min}, \lambda_{max} - \Delta\lambda_{gc}))$ when $R^L \cos(\Delta\lambda_{gc}) \geq R^S$. This quick estimation algorithm has a complexity of $O(k)$, where k is the total number of discrete θs. It is also noted that when the difference of R^L and R^S is getting larger, the condition of $R^L \cos(\Delta\lambda_{gc}) \geq R^S$ is easily satisfied, which means that this

algorithm is applicable in most of the cases. When $R^{L}\cos(\Delta\lambda_{gc}) < R^{S}$, however, the CA algorithm has to be applied for obtaining the GC posture range of T^{S}.

The posture range of T^{S} for both RG and GC at θ is the intersection of (λ_{rg}, λ_{max}) and (λ_{min}, λ_{gc}), i.e., (($\lambda_{min} + \Delta\lambda_{rg}$), ($\lambda_{max} - \Delta\lambda_{gc}$)). The overall RG and GC posture range can be obtained by applying this method to all the discrete θs. In case when no posture range of RG and GC is found, the CA algorithm needs to be applied to T^{S} at this point.

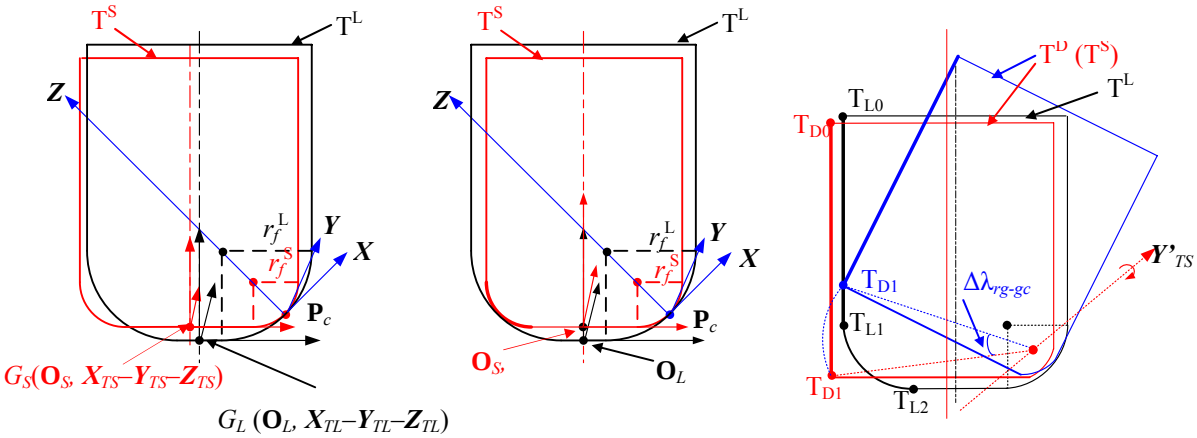
4.2.5 $R^{S} < R^{L}$ and $r_f^{S} < r_f^{L}$

Given a θ within the machine limits, T^{L} is accessible to $\mathbf{P}_c$ with a range of (θ, (λ_{min}, λ_{max})). Figure 4.8a shows T^{L} and T^{S} with the same posture (θ, λ) at $\mathbf{P}_c$ ($\lambda_{min} \le \lambda \le \lambda_{max}$). From Eq. (4.1), vector $\mathbf{O}_L\mathbf{O}_S$ can be represented in G_L as $(dx, dy, dz)^T = ((R^{L} - R^{S}) + (r_f^{S} - r_f^{L})(1 - \sin\lambda), 0, (r_f^{L} - r_f^{S})(1 - \cos\lambda))^T$. Firstly, let us look at a special scenario that along a posture (θ, λ), the filleted portion of T^{S} is in full contact with that of T^{L} (see Figure 4.8b). It can be clearly seen that the whole outer surface of T^{S} is inside that of T^{L}. If $\lambda_{min} \le \lambda \le \lambda_{max}$, T^{S} is free of rear-gouging and global-collision. This situation can be mathematically defined as $dx = 0$ or $R^{S} = R^{L} - (r_f^{L} - r_f^{S})(1 - \sin\lambda)$. When R^{S} becomes smaller, i.e., $R^{S} \le R^{L} - (r_f^{L} - r_f^{S})(1 - \sin\lambda)$, T^{S} is also interference-free at (θ, λ). Therefore, the lower-bound of λ for the posture range of T^{S} is,

$$\lambda_{\text{low-bound}} = \sin^{-1}\left(\frac{(R^{S} - r_f^{S}) - (R^{L} - r_f^{L})}{(r_f^{L} - r_f^{S})}\right) \tag{4.6}$$

The posture range of T^{S} is ($\max(\lambda_{min}, \lambda_{\text{low-bound}})$, λ_{max}). Therefore, if $R^{S} \le R^{L} - (r_f^{L} - r_f^{S})(1 - \sin\lambda)$ can be satisfied when $\lambda_{min} \le \lambda \le \lambda_{max}$, Eq. (4.6) can be used as a quick estimation method. Otherwise, there is no guarantee that T^{S} is free of

interference at (θ, λ). For this case, a different method, introduced in the following section, is used to find the posture range for T^S.



(a) T^L and T^S (general situation) (b) T^L and T^S (special situation) (c) T^L and T^D (T^S) at $\lambda = \lambda_{min}$

Figure 4.8 T^L and T^S ($R^S < R^L$, $r_f^S < r_f^L$) with the same posture

Again, we follow the slicing approach introduced in section 4.2.2. Figure 4.8c shows the sections of T^L and T^S along λ_{min} at $y = y_{Tk}$ ($-R^S \le y_{Tk} \le R^S$). It can be easily proved that the front section curve of the cylindrical portion of T^S is inside the section curve of T^L. On the other hand, the bottom plane of T^S is always above that of T^L. Therefore, the front-half of T^S section is interference-free. Part of the back-half of T^S section, however, may be out of the boundary of T^L section. Therefore, the first task is to identify this out-of-boundary portion of T^S section. The second task is to rotate T^S section clockwise until the out-of-boundary portion falls onto or inside T^L section. This angle increment determines the lower-bound of λ for the posture range of T^S section.

These two tasks, however, cannot be accomplished analytically. Instead, a numerical problem is faced. To simplify this problem, a two-phase simplification process is applied. Firstly, a cutter (R^L, r_f^S) is employed to replace T^S and it is seen

from section 4.2.3 that T^S will be interference-free if cutter (R^L, r_f^S) is at a (θ, λ). Secondly, the left-bottom corner of the section of cutter (R^L, r_f^S) is changed to a sharp one with a right angle. This created section is used as a dummy section for T^S section, called the T^D section (see Figure 4.8c). It is clear that if T^D section is interference-free ($-R^S \le y_{Tk} \le R^S$), T^S section is interference-free. Therefore, the interference-free range of λ for T^D section ($-R^S \le y_{Tk} \le R^S$) can be used as a conservative estimate for the interference-free range of λ for T^S section.

Now, it is assumed that $\lambda = \lambda_{min}$. Referring to Figure 4.8c, if T_{D1} is rotated about $\boldsymbol{Y'}_{TS}$ by $\Delta\lambda_k$ such that $\mathbf{T}_{D1}$ falls on $\mathbf{T}_{L0}\mathbf{T}_{L1}$, T^D section is interference-free. A conservative estimated posture range for T^S section is therefore ($\lambda_{min}+ \Delta\lambda_k$, λ_{max}). $\Delta\lambda_k$ can be calculated by,

$$\Delta\lambda_k = \sin^{-1}\left(\frac{r_f^S}{d_k}\right) + \cos^{-1}\left(\frac{R^L - r_f^S + \sqrt{(R^L)^2 - (y_{Tk})^2} - |dx_0|}{d_k}\right) \tag{4.7}$$

Where d_k denotes the distance from $\mathbf{T}_{D1}$ to $\boldsymbol{Y'}_{TS}$-axis as $d_k = \sqrt{\left[R^L - r_f^S + \sqrt{(R^L)^2 - y_{Tk}^2}\right]^2 + (r_f^S)^2}$, and $|dx_0|$ denotes the distance of $\mathbf{T}_{L0}\mathbf{T}_{L1}$ to $T_{D0}T_{D1}$ as $|dx_0| = (r_f^S - r_f^L)(1 - \sin\lambda_{min})$. The maximum value of $\Delta\lambda_k$ appears when $|y_{Tk}|$ reaches its maximum, i.e., $y_{Tk} = -R^S$ or $y_{Tk} = R^S$.

$$\begin{aligned}\Delta\lambda_{max} = &\sin^{-1}\left(\frac{r_f^S}{\sqrt{\left[R^L - r_f^S + \sqrt{(R^L)^2 - (R^S)^2}\right]^2 + (r_f^S)^2}}\right) \\ &+ \cos^{-1}\left(\frac{R^L - r_f^S + \sqrt{(R^L)^2 - (R^S)^2} - |dx_0|}{\sqrt{\left[R^L - r_f^S + \sqrt{(R^L)^2 - (R^S)^2}\right]^2 + (r_f^S)^2}}\right)\end{aligned} \tag{4.8}$$

The RG and GC posture range at θ for T^S is therefore, (θ, (λ_{min}+ $\Delta\lambda_{max}$,, λ_{max})). The overall interference-free posture range can be obtained by applying this method to all the discrete θs. This quick estimation algorithm also has a complexity of O(k), where k is the total number of discrete θs. In case that (λ_{min}+ $\Delta\lambda_{max}$) > λ_{max} for all the θs, it means that no feasible map can be found using this quick estimation method. In this case, the CA algorithm has to be applied to T^S at this point.

4.2.6 Discussion

Based on the accessibility comparison between T^S and T^L, the RG and GC posture ranges of T^S can be obtained from that of T^L, except that (1) the CA algorithm has to be used for obtaining the GC posture range for scenario 1 and (2) the CA algorithm may be used for obtaining the GC posture range in some case for scenario 3. The developed quick estimation methods are analytical in nature and each has a complexity of O(k). Compared to the CA algorithm with a complexity of O(km), the computation load is significantly lower. The algorithm, called *Express APR Finder*, for accessibility analysis of the current cutter is described as follows:

Algorithm: *Express APR Finder*

Input: (1) *The whole set of sampled points* $\{\boldsymbol{P}_i\}$, $i = 1, \ldots, N$.

(2) *Cutters* $\{T_k (R, r_f), k = 1, 2, \ldots, M-1\}$ *that have been checked with their corresponding accessible points* $\{\boldsymbol{P}_a|T_k\}$ *and non-accessible ones* $\{\boldsymbol{P}_{na}|T_k\}$.

(3) *The current cutter* $T_M (R^M, r_f^M)$

Output: *The posture range of* T_M *at every sampled point*

Begin

(1) Set $i = 0$.

(2) Find the ML and LG posture ranges of T_M at $\mathbf{P}_i$ using the method introduced in Chapter 2.

(3) If $\mathbf{P}_i$ cannot be found in any of the $\{\mathbf{P}_a \mid T_k\}$(k = 1, 2, …, M-1), apply the CA algorithm to find the RG and GC posture ranges of T_M at $\mathbf{P}_i$. Go to (6).

(4) Find all the cutters from $\{T_k, k = 1, 2, \ldots, M\text{-}1\}$whose $\{\mathbf{P}_a \mid T_k\}$contain $\mathbf{P}_i$ and form a cutter set $\{T_j\}$

(a) IF there is a cutter in $\{T_j\}$ that has the same r_f as r_f^M, T_M has the same RG and GC posture ranges as this cutter. Go to (5).

(b) Take the largest cutter (or the first cutter) from $\{T_j\}$ as T^L, and T_M as T^S, apply one of the quick algorithms introduced in sections 4.2.2, 4.2.4, and 4.2.5 (based on the relationships between their major and minor radii) to obtain the RG and GC posture ranges of T^S at $\mathbf{P}_i$. Go to (5).

(5) Find the APR as the intersection of ML, LG, RG and GC. IF no feasible posture can be found in APR, apply the CA algorithm to find the RG and GC posture ranges of T_M at $\mathbf{P}_i$.

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

(7) IF $i \leq$ N, go to (2); otherwise, output the posture range of T_M at every sampled point, stop.

End

Based on *Express APR Finder* algorithm, we proceed to introduce a non-redundant search algorithm to find the optimal cutter for machining a given sculptured surface.

4.3 A Non-redundant Algorithm for Optimal Cutter Selection

In 5-axis machining, a trial-and-error approach can be adopted in which a cutter is taken from the list (from large to small) and the CA algorithm is run at all sampled points on the surface, until a feasible cutter is found. However, extensive computation is involved. In this section, a non-redundant algorithm is proposed by combining the CA algorithm, the quick estimation algorithms based on accessibility comparison between cutters, and the surface decomposition presented in Chapter 3. The algorithm begins with the largest cutter, T_{top} from the list. The sculptured surface is decomposed as interference-free and interference-prone regions, and the accessible posture range at every sampled point on the interference-prone regions is obtained by applying the CA algorithm. If T_{top} is accessible to every sampled point, it is the optimal cutter. Otherwise, the sampled points are categorized into *accessible* and *non-accessible*. Subsequently, the next cutter in the list is taken and *Express APR Finder* is applied to the cutter at every sampled point. This process continues until a feasible cutter is found. This algorithm is described as follows:

Algorithm: ***Non-Redundant Cutter Selection***

Input: (1) *The whole set of sampled points* $\{\boldsymbol{P}_i\}$, $i = 1, \ldots, Ns$, *which represents the sculptured surface*

(2) *A cutter list* $\{T_j (R, r_f, L), j = 1, 2, \ldots, l\}$

Output: *The optimal cutter for machining the surface*

Begin

(1) Set $j = 1$ and T_j as the current cutter.

(2) Decompose the surface into interference-prone $\{S_{j\text{-}free}\}$ and interference-free $\{S_{j\text{-}prone}\}$, with the algorithm introduced in Chapter 3.

(3) The algorithm of *Express AM Finder*, introduced in Section 4.2.6 is employed to check cutter accessibility to the points in $\{S_{j\text{-}prone}\}$.

(4) If at a point, the posture range is not available, $j = j+1$. Set T_j as the current cutter and go to (2). If no more cutters are left in the cutter list, go to (6).

(5) Output "the optimal cutter is T_j". Stop.

(6) Output "no cutter is feasible to finish the surface". Stop.

End

4.4 Summary

In 5-axis milling of sculptured surfaces, the cutter's accessibility to a point on the surface plays a key role in process planning tasks such as optimal cutter selection for finish cut. When the previously developed cutter accessibility algorithm is applied to cutters with different dimensions independently, heavy computation load is often encountered. In this chapter, the accessibility between a larger cutter and a smaller cutter has been studied based on their geometric characteristics. Several quick estimation algorithms for obtaining the accessibility of the smaller cutter based on that of the larger cutter have been developed, which leads to significant savings in computation. By combining the quick estimation algorithm, the accessibility algorithm in Chapter 2, and the surface decomposition algorithm in Chapter 3, a non-redundant algorithm has also been proposed to find the optimal cutter for sculptured surface machining.

CHAPTER 5

TOOL-PATH GENERATION PART 1:

DETERMINATION OF PATH DIRECTION

In process planning, the task of tool-path generation is to select a tool-path pattern and path direction, generate the cutter-contact (CC) points that satisfy the accuracy requirement, and determine the cutter's posture (orientation) at every CC point without causing any interference. During the whole process, determination of the cutter posture is one of the critical issues in process planning for 5-axis machining. However, the determination of cutter posture is complicated and time-costing due to the two rotational axes in a 5-axis machine. On the other hand, an algorithm has already been developed in the last chapter for selecting the optimal cutter to finish the given surface, together with the A-map at each sampled point. Since the density of the sampled points is much higher than that of CC points, this checking result can be utilized in the process of tool-path generation through an approximation approach. In this way, the process planning problems can be solved in an integrated and efficient manner. This possibility serves as a worthwhile research topic in order to reduce the consumption time in tool-path generation. Several issues in tool-path generation have been studied in this work, including path direction for a path pattern and CL data generation. This chapter addresses the former issue by using the A-map at each sampled point.

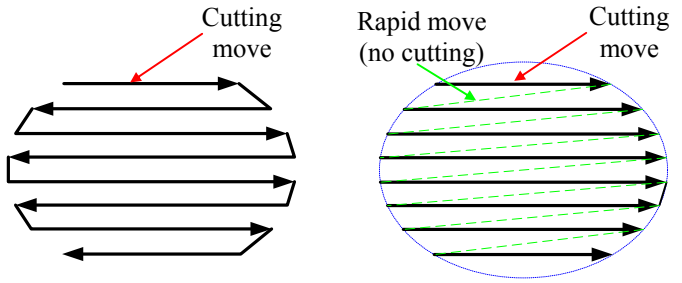
5.1 Introduction

Iso-planar (Cartesian) tool-paths are generated by intersecting parallel planes with the given part surface. Over the years, the iso-planar path has received extensive attention owing to its robustness in almost every scenario, e.g., the machining of compound and composite surface (Choi *et al*., 1988), trimmed surface (Yang *et al*., 2003) and pocketing (Jamil, 1994). In theory, many newly developed tool-path generation techniques are effective in sculptured surface machining, such as space-filling path (Marshall and Griffiths, 1993), adaptive iso-parametric tool-path (Elber and Cohen, 1994), iso-curvature path (Jensen and Anderson, 1992), isophoto-based path (Ding *et al*., 2003), guide surface path (Kim and Choi, 2000), non-iso-parametric path (Lee, 1998), boundary-conformed path (Yang *et al.*, 2003), constant scallop path (Li and Feng, 2004) and so on. In practice, however, the iso-planar approach has not been replaced and is still used in commercial CAM software owing to its robustness.

In iso-planar path generation, path orientation is one critical issue to affect machining efficiency and accuracy (Sarma, 1999). There must exist an optimum path for every shape at a specific orientation (Lakkaraju and Raman, 1990). In the literature of iso-planar tool-path generation for sculptured surfaces (Li and Jerard, 1994; Choi *et al*., 1988; Xu *et al*., 2002; Ding *et al*., 2003), most of the previous effort has been on developing automated methods that produce a surface with high efficiency and accuracy. The common assumption is that a path direction is already given or set by the user. However, in sculptured surface machining, a user may find it difficult to set a suitable path orientation to finish the surface owing to complex surface geometry and complicated cutter motion in a 5-axis machine. The user may select several path directions, apply the tool-path generation, and conduct the simulation and verification to find the optimum path orientation. This approach can be

time-consuming and costly. It is therefore needed to develop the technique to automatically determine the path direction for finishing a given surface before tool-path generation in 5-axis milling.

Two common types of iso-planar path are zigzag tool-path (Figure 5.1a) and one-way tool-path (Figure 5.1b). Although zigzag-milling provides high productivity in surface machining, one-way milling results in high surface quality and longer tool life in high speed machining (Schulz, 1996). The scope of this work is limited to one-way iso-planar path in finish cut.



(a) zigzag tool-path pattern (b) one-way tool-path pattern

Figure 5.1 Two types of direction-parallel tool-path

The task of path direction determination can be defined as "given a part surface, a suitable cutter, a 5-axis machine tool, find the suitable path direction to finish the given surface with high surface quality and machining efficiency". Here, this task is implemented based on the A-map at each sampled surface point, the checking result from cutter selection which must be conducted before tool-path generation. It is foreseen that the characteristic of cutter orientation at each surface point reflects cutting property of the entire surface, such as machining efficiency and tool dynamics. Regarding to heuristic machining strategies, algorithms are developed in this chapter to evaluate the characteristic of cutter motion at each surface point and

at the entire surface, and then to search for the optimal path direction for finishing the given surface.

5.2 Related Works

Many recent research results focus on the optimal direction of iso-planar paths for cutting a planar area. Most of them are on the zigzag pocket problem. In the literatures, different aspects are taken into account to improve the efficiency and accuracy of a tool-path. The early attempt to optimize zigzag tool-path focuses on the influence of the path length on the cutting time (Wang *et al*., 1987; Wang *et al*., 1988; Prabhu *et al*., 1990). However, Sarma (1999) argued that it is the number of switch backs that is the important contributor to cutting time, especially in high-speed machining. He then introduced the concept of a crossing function of a two-dimensional contour to measure the times the raster-lines intersect with the contour, and then applied this concept into the selection of optimal path direction. Other work concentrates on the influence of tool retraction on the cutting quality and cutting time. Held (1994) selected a path direction that minimizes the number of local extrema (scan-reflex vertex), under the assumption of a significantly smaller tool radius in relation to the machining area. Park and Choi (2000) extended this approach to pick a path direction that minimizes the number of scan-reflex vertices after removing local (small) features, the reflex profiles with length smaller than the too-path interval. These algorithms are limited to two-dimensional area manufacturing with fixed tool axis direction. Only boundary geometry is considered in the optimization procedure. Though comprehensive and effective, the algorithms cannot be directly extended to 5-axis sculptured surface machining.

In a 5-axis machine, cutter axis direction is changed simultaneously as the cutter moves along tool-paths to match the complex surface geometry for efficient machining. The dynamic change in cutter motion and orientation puts significant impact on the machining efficiency and accuracy, especially in high speed machining. First, at high speed rates, a significant portion of the tool motion is dedicated to accelerate/decelerate near the points with sharp changes in path direction and cutter orientation (Monreal and Rodriguez, 2004). Second, since surface finishing is a very delicate operation, sharp turns in cutter motion push the cutter off the course, leaving "tool-marks" on the machined surface and resulting in out-of-tolerance areas (Choi and Jerard, 1998). By taking into account of the characteristics of cutter motion in a 5-axis machine, this chapter proposes an optimization approach for path direction selection in sculptured surface machining from both global and local point of view. To the entire surface, the global change of cutter motion is examined to ensure the smooth tool dynamics in machining a sculptured surface. At a single point, the cutter posture is identified to ensure the maximized cutting efficiency with respect to a certain path direction. The machining with the resultant path direction would lead to high machining efficiency and machining surface quality.

The A-map at each sampled point is one of the checking results from the cutter selection procedure. Any point within the A-map represents a cutter posture with which the cutter can access the surface point without interference. To determine the optimal path direction, cutter posture should be identified from the A-map for the evaluation of cutter motion, which involves some machining strategies. Thus, the commonly used machining strategies are firstly presented before the discussion on optimal path direction.

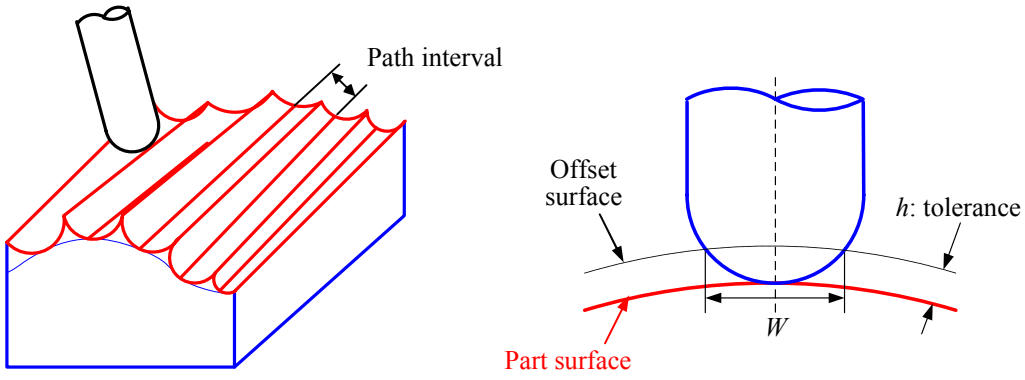
5.3 Machining Strategies in 5-axis Finish Cut

To successfully finish a surface, the cutter posture at each CC point should be identified such that interference does not occur and high machining efficiency is achieved. Generally, there exists a cutter posture in the A-map with which the cutter geometry closely matches the part surface geometry at a point for highly efficient machining. Given the part surface and cutter's A-map at a surface point, a machining strategy will help to determine the optimal cutter posture to cut the local surface. Some of the commonly used machining strategies are listed as follows (Balasubramaniam *et al.*, 2003):

- *Smooth cutter dynamics*: since the cutter moves with high feed rates and speeds in high speed machining, such consideration as machine tool dynamics, and the saturation of the actuators in the machine, makes significant impact on the tool life and surface finish quality. Finish cut is a delicate operation with tight accuracy requirement. It is necessary to keep the change rate of cutter posture as small as possible. One possible approach is to select the orientations between neighboring points with small change (Jun *et al.*, 2003) and another is to select a path direction with change as small as possible in cutting process.
- *Machining strip width*: Path interval between two sequential tool-paths is one of the important factors for cutting efficiency, as shown in Figure 5.2a. Generally, path interval at a CC point is determined by the machining strip width (see Figure 5.2b), a combination function related to cutter posture, cutter geometry, surface geometry, feeding direction and allowable surface profile tolerance (Lee, 1998; Yoon *et al.*, 2003). To maintain efficient cutting, it is preferable to have the machining strip width as large as possible. It is worth mentioning that maximum strip width at each discrete point does not imply

that the cutting-load will abruptly change on a tool-path. Besides the machining strip width, cutting load is also related to other geometric and non-geometric factors, such as surface geometry and cutting parameters (Monreal and Rodriguez, 2004). This work mainly concerns the machining strip width. Cutting load leveling and smoothing can be left to tool-path planning (Choi and Jerard, 1998).

Which of these criteria is more important depends on the application, and several criteria are possible to be combined and adopted for the decision of cutter posture and path direction. In this chapter, we use the combination of these criteria. From a global point of view, the optimal path direction should be determined such that the change rate of cutter motion along a tool-path as small as possible to keep smooth tool dynamics. From a local point of view, large fluctuation of cutter posture between neighboring points should be prevented while keeping the maximized machining strip width. Based on these heuristic criteria, the optimal path direction for a sculptured surface will be determined by the algorithm to be introduced in the next section.



(a) path interval (b) machining strip width (W)

Figure 5.2 Path interval and machining strip width

5.4 Determination of Path Direction

As mentioned above, there are generally two optimization objectives in the process to select a cutting direction for the iso-planar paths. The first one is that the selected direction should produce high machining efficiency or short overall tool-path length. The second is to maintain smooth tool dynamics, which is particularly crucial in high speed machining. In this section, both objectives are taken into account to determine the optimal path direction for finishing a given surface. The general procedure is briefly described here. To consider the machining strategies mentioned above simultaneously without having the actual tool-paths and the corresponding postures, an evaluation factor is introduced at every sampled point on the part surface: the posture change rate (PCR). Given a sampled point **P** and one of its neighboring sampled points, $\mathbf{P}_{\text{next}}$, the PCR along the direction of $\mathbf{PP}_{\text{next}}$ is the difference between the two corresponding postures then normalized by the distance. If we can obtain the PCRs of all the sampled points $\{\mathbf{P}_i, i = 1, 2, \ldots, n\}$ along any given cutting direction, the direction that possesses the minimum $\sum_{i=1}^{n} \text{PCR}_i$ can be chosen as the one with the smoothest tool-path.

The remaining of this section will discuss the issues in the general procedure for the optimal path direction. To obtain the PCR at a sampled point **P**, the corresponding postures at **P** and $\mathbf{P}_{\text{next}}$ should be identified from the A-maps at both points. In Section 5.4.1, the cutter posture at a point of interest is first selected within the A-map by applying specific machining strategies. Based on the specified cutter postures, in Section 5.4.2, the method to obtain the PCR at a sampled point is proposed along each cutting direction defined in the global frame. Section 5.4.3

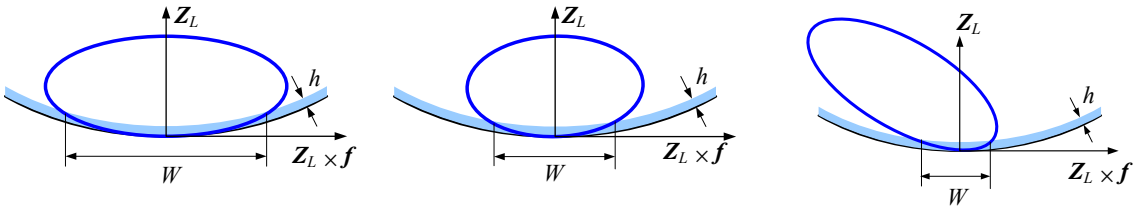
presents the detailed algorithm for selecting the path direction before generating actual tool-paths.

5.4.1 The cutter posture along a path direction at a surface point

Within the A-map, an extensive number of cutter postures can be used to machine the local surface without interference to the surface. However, there exist postures with which the cutter geometry can closely match the part surface for better cutting. Thus, it is necessary to rate the postures within the A-map according to some machining strategies. Both the machining stripe width and smooth cutter dynamics are considered here. As shown in Figure 5.2b, machined strip width W is defined as the distance between two neighboring cusp height curves when a cutter is moving along a tool-path. The cusp height curves are formed by the intersection of cutter swept volume and the part offset surface, one with a distance as the allowable profile tolerance from the part surface. Since machining strip width is crucial in determination of path interval and closely related to cutting efficiency, it is preferable to have a cutter posture with the largest machining strip width when other requirements are fulfilled. On the other hand, sharp change of cutter posture results in high acceleration/deceleration on the motor speed, leaving "tool marks" on the machined surface and producing out-of-tolerance areas. It is thus preferable to have a cutter posture at a point with a small change from that at the neighboring point. Both aspects are addressed in the following discussion.

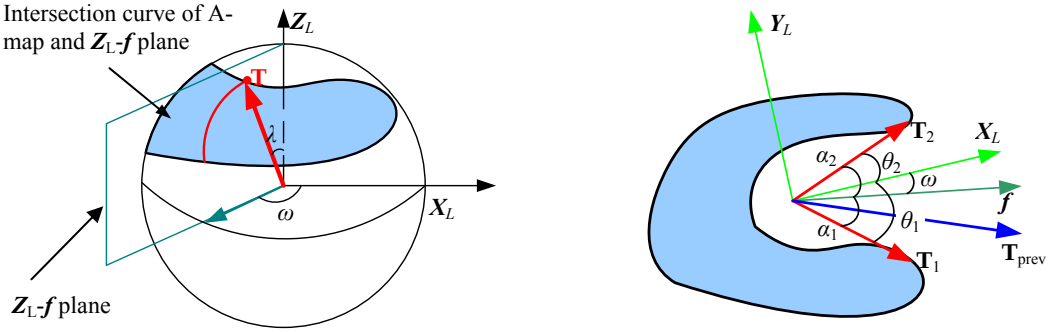
Figure 5.3a shows the elliptical cross-section of the cutter on the normal plane perpendicular to the feeding direction $\boldsymbol{f}$. The strip of the material that can be effectively removed by the cutter is determined from the elliptical cross-section and the surface with offset h from the part surface (a more detailed discussion is provided

in Section 6.3.2.1). The machining strip width decreases with the increment of the tilt angle λ since the major radius of the ellipse is decreasing (Lee, 1998), as shown in Figure 5.3b. Furthermore, the machining strip width also decreases with the increment of value $|\alpha|$ since the cutter curvature increases at the point touching $\mathbf{P}_c$ (see Figure 5.3c), where α is the angle between the cutter axis and feeding direction on $\boldsymbol{X}_L$-$\boldsymbol{Y}_L$ plane (see Figure 5.4b). For better cutting efficiency, a heuristic is adopted here to select the cutter posture such that maximum machining strip width is achieved. From the A-map, we find θ, which is the closest to the feeding rotational angle ω in the local frame, and take the minimum feasible inclination angle λ to form the posture (θ, λ) at $\mathbf{P}_c$, as shown in Figure 5.4a and b.



(a) cutter posture (α_0, λ_0) (b) cutter posture (α_0, λ_1), $\lambda_1 > \lambda_0$ (c) cutter posture (α_1, λ_0), $|\alpha_1| > |\alpha_0|$

Figure 5.3 Machining strip width (Lee, 1998)



(a) A-map (b) the projection of A-map on $\boldsymbol{X}_L$-$\boldsymbol{Y}_L$ plane

Figure 5.4 Selection of cutter posture from A-map

Another consideration can be added to smooth the tool dynamics by reducing the variation in postures along a pair of neighboring cutter points. The reason is that the jump in cutter orientation causes deterioration in dynamic performance of the machine tool. In machine frame, an orientation is represented by cutter axis unit vector as:

$$\mathbf{T} = (\sin\lambda\cos\theta)\boldsymbol{X}_L + (\sin\lambda\sin\theta)\boldsymbol{Y}_L + (\cos\lambda)\boldsymbol{Z}_L \tag{5.1}$$

Where $\boldsymbol{X}_L$, $\boldsymbol{Y}_L$ and $\boldsymbol{Z}_L$ are the unit local frame axis vectors represented in global frame. A selection heuristic can be formed that selects a (θ, λ) with minimum distance of $\mathbf{T}$ from that of the posture $(\theta, \lambda)_{\text{prev}}$ at a previous cutter contact point, if any.

It is observed that the postures from these 2 heuristics are not necessarily in line. At this stage, the first heuristic is taken as the major consideration to select the cutter posture with the maximum machining strip width. The feeding direction in global frame is firstly converted to the angle ω in local frame. From the A-map of $\mathbf{P}_c$, if λ exists at $\theta = \omega$, the minimum feasible λ is taken to form the posture (θ, λ) at $\mathbf{P}_c$. Otherwise, the θ closest to ω is selected, and then the minimum feasible λ at θ to form posture (θ, λ).

As shown in Figure 5.4b, according to first heuristic, two postures with same $|\alpha|$ value (i.e., α_1 and α_2) in A-map may exist from clockwise and counter-clockwise directions, respectively. In this situation, the second heuristic concerning cutter posture change is applied. At θ_1 and θ_2, the minimum feasible λ_1 and λ_2 are taken to form the postures, with corresponding cutter orientation as $\mathbf{T}_1$ and $\mathbf{T}_2$ in global frame. The one (i.e., $\mathbf{T}_1$ in Figure 5.4b) with the smaller distance in global frame from the cutter orientation $\mathbf{T}_{\text{prev}}$ at previous point of interest is then selected as the cutter posture at $\mathbf{P}_c$.

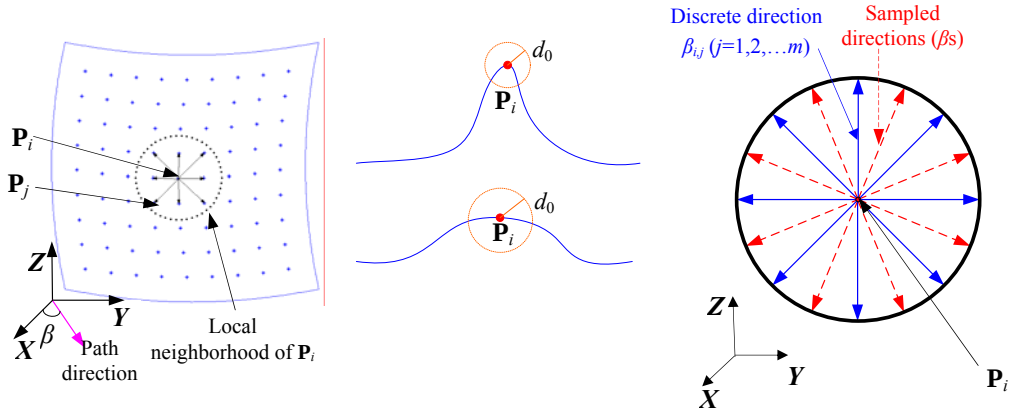
The computation complexity is O(k) to select the cutter posture from the A-map at a point, where k is the total number of discrete θs sampled over the range ($\theta_{\min}$, $\theta_{\max}$). It is noted that, users can also implement other selection criteria to suit their special requirements. For example, forward leaning can be a necessary requirement to prevent the cutter from burnishing the surface (Balasubramaniam *et al.*, 2003). This constraint, say $-30^{\circ} \le \theta - \omega \le 30^{\circ}$, can be incorporated fairly trivially by trimming the A-map.

It can be seen that more concern is on the first heuristic in this stage to select the cutter posture at a surface point, while less on the second heuristic. In the next stage, more consideration will put on the second heuristic to obtain the optimal path direction with minimum posture change of a cutter when machining the whole surface.

5.4.2 PCR at a point

In this subsection, we show how to obtain the PCR at a sampled point, $\mathbf{P}_i$, along a given path direction defined in the global frame on the ***X*-*Y*** plane (see Figure 5.5a). A two-step approach is developed to achieve this. In the first step, the discrete PCRs of $\mathbf{P}_i$ is evaluated in respect of the neighboring points around $\mathbf{P}_i$. As shown in Figure 5.5a, a local neighborhood of $\mathbf{P}_i$ is firstly defined that contains m sampled points $\{\mathbf{P}_j, j = 1, 2, \ldots, m\}$. In this work, the neighborhood is formed based of distance of $\mathbf{P}_i\mathbf{P}_j$, i.e., $|\mathbf{P}_i\mathbf{P}_j| \le d_0$, where d_0 is a predefined value according to the curvature at $\mathbf{P}_i$, e.g., $d_0 = \min\{1.0/\max(|\kappa_{\max}|, |\kappa_{\min}|),\ 10\}$ mm. As shown in Figure 5.5b, a smaller d_0 is needed at the point with larger curvature, while a larger d_0 at the point with smaller curvature. For each neighboring point $\mathbf{P}_j$, the discrete PCR of $\mathbf{P}_i$ along the direction of $\mathbf{P}_i\mathbf{P}_j$ is evaluated. In the second step, the PCR of $\mathbf{P}_i$ is obtained along any cutting direction β in the global frame, PCR_β, by making use of the discrete PCR_{ij}. Firstly, at

$\mathbf{P}_i$, we sample the range of (0, 2π) uniformly into a number of discrete angle values βs, each corresponding to a path direction (see Figure 5.5c). Take any direction along β, we find the two closest β_{ij} to β in the global frame. PCR_β can then be obtained by applying linear interpolation of the two corresponding PCR_{ij}. Since the density of the sampled points is generally high, the linear interpolation should produce close results. It can be seen that, the main challenge in this approach is to calculate the discrete PCR of $\mathbf{P}_i$ along a direction of $\mathbf{P}_i\mathbf{P}_j$, which is discussed in the next discussion.



(a) Calculating $PCR_{i,j}$ in the local neighborhood of $\mathbf{P}_i$

(b) Local neighborhood of $\mathbf{P}_i$

(c) Calculating PCR of $\mathbf{P}_i$ along the sampled directions

Figure 5.5 Obtain the PCR of $\mathbf{P}_i$ along all the cutting direction

PCR_{ij} is the difference between the two corresponding postures at $\mathbf{P}_i$ and $\mathbf{P}_j$ then normalized by the distance. Several representations can be introduced to measure the difference between two corresponding postures including (1) the difference of cutter posture angle values in respective local frames (Balasubramaniam *et al.*, 2003); (2) the angle between two unit vectors along cutter axis in global frame; and (3) the end distance between axis vectors mapped on a unit sphere in global frame. The main merit of the first representation is its simplicity to calculate, since no transformation

of cutter posture between local frame and global frame is needed. However, local frames at the two points might not be same due to irregular surface geometry. The representation therefore cannot reflect the real difference between the two postures. The second representation with the angle between two vectors is straightforward since both vectors are in the same frame. The third representation inherits the advantage of the second but is simpler to be evaluated, which does not involve trigonometric function. In this work, the third representation is utilized for calculating PCR_{ij} as:

$$PCR_{ij} = \frac{\left|\mathbf{T}_i\mathbf{T}_j\right|}{\left|\mathbf{P}_i\mathbf{P}_j\right|} \tag{5.2}$$

Where $\mathbf{T}_i$ and $\mathbf{T}_j$ are the unit vectors in global frame along the cutter axis directions at point $\mathbf{P}_i$ and $\mathbf{P}_j$, respectively (see Figure 5.4a). It is noted that $\mathbf{T}_i$ and $\mathbf{T}_j$ should be represented in the global frame instead of the local frame, since the local frames at two points varies from varied surface local geometry.

To calculate PCR_{ij} using Eq.(5.2), the cutter posture corresponding to $\mathbf{T}_i$ and $\mathbf{T}_j$ need to be obtained first. As discussed in Section 5.4.1, the cutter posture at a point can be determined from the A-map with regard to machining strategies when a path direction in local frame is given, i.e., ω in Figure 5.4b. In this study, it is assumed that cutting planes are perpendicular to ***XY***-plane of global frame. In global frame, the path direction passing through $\mathbf{P}_i$ (x_i, y_i, z_i) and $\mathbf{P}_j$ (x_j, y_j, z_j), represented by an angle of the cutting plane from the ***XZ***-plane in counter clockwise direction, is given as:

$$\beta_{ij} = \tan^{-1}\left(\frac{y_j - y_i}{x_j - x_i}\right) \tag{5.3}$$

To use the method introduced in Section 5.4.1 for determining cutter postures at point $\mathbf{P}_i$ and $\mathbf{P}_j$, β_{ij} has to be converted to anlge ω_i and ω_j in respective local frame as:

$$\begin{aligned} \cos\omega_k &= \frac{([-\sin\beta_{ij}, \cos\beta_{ij}, 0]^T \times \boldsymbol{Z}_{Lk})}{\left|[-\sin\beta_{ij}, \cos\beta_{ij}, 0]^T \times \boldsymbol{Z}_{Lk}\right|} \bullet \boldsymbol{X}_{Lk} \\ \sin\omega_k &= \frac{([-\sin\beta_{ij}, \cos\beta_{ij}, 0]^T \times \boldsymbol{Z}_{Lk})}{\left|[-\sin\beta_{ij}, \cos\beta_{ij}, 0]^T \times \boldsymbol{Z}_{Lk}\right|} \bullet \boldsymbol{Y}_{Lk} \end{aligned} \quad (k = i, j) \tag{5.4}$$

Where $[-\sin\beta_{ij}, \cos\beta_{ij}, 0]^T$ is the normal vector of the cutting plane in global frame. $\boldsymbol{X}_{Lk}$, $\boldsymbol{Y}_{Lk}$ and $\boldsymbol{Z}_{Lk}$ denote the unit axis vectors of the local frame at point $\mathbf{P}_k$ (k = i, j) represented in global frame, respectively (See Figure 2.1a). Using Eq. (5.4), path direction β_{ij} is converted to angle ω_i in the local frame at point $\mathbf{P}_i$ and angle ω_j at point $\mathbf{P}_j$. It is worth noting that the value of ω_i and ω_j might be different, because of varied local frame configuration ($\boldsymbol{X}_L$-$\boldsymbol{Y}_L$-$\boldsymbol{Z}_L$) at different surface points. Regarding to angles ω_i and ω_j, both the cutter orientation (λ_i, θ_i) at $\mathbf{P}_i$ and (λ_j, θ_j) at $\mathbf{P}_j$ can be determined from their A-maps, using the algorithm introduced in Section 5.4.1.

In Eq. (5.2), $\mathbf{T}_i$ and $\mathbf{T}_j$ are represented in global frame instead of local frame. It is easy to transform (λ_i, θ_i) and (λ_j, θ_j) to $\mathbf{T}_i$ and $\mathbf{T}_j$ using Eq. (5.1). Up to now, the discrete PCR_{ij} (j = 1, 2 …, m) can be obtained by using the Eqs. (5.1)–(5.4) and integrating the approach of cutter posture specification in Section 5.4.1. Based on the discrete PCR_{ij}s, we can find the PCR_β at $\mathbf{P}_i$ along every direction using a linear interpolation approach as (see Figure 5.5c):

$$\text{PCR}_\beta = \text{PCR}_{ij1} + \frac{\text{PCR}_{ij2} - \text{PCR}_{ij1}}{\beta_{ij2} - \beta_{ij1}}(\beta - \beta_{ij1}) \tag{5.5}$$

Where β_{ij1} and β_{ij2} are the two closest to β among β_{ij} (j = 1, 2 …, m), and PCR_{ij1} and PCR_{ij2} are the corresponding discrete PCR respectively.

5.4.3 Overall searching algorithm for the optimal path direction

In the process for determining the path direction, the discrete directions are firstly obtained by uniformly sampling the path direction among the angle range of [0, 2π]. After obtaining the PCRs of all the sampled points $\{\mathbf{P}_i, i = 1, 2, \ldots, n\}$ along any given cutting direction, the direction that possesses the minimum $\sum_{i=1}^{n} \mathrm{PCR}_i$ can be chosen as the one with the smoothest tool-path. The overall algorithm is described as follows.

Algorithm: ***Searching for the path direction to finish a sculptured surface***

Input: (a) *A set of points $\{\boldsymbol{P}_i\}$ (i = 1,..., n) representing a sculptured surface S(u,v)*

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

(c) *Path direction angle range ($\beta_{\min}, \beta_{\max}$)*

Output: *Optimal path direction β_s for finishing the given surface*

Begin

(1) Set $i = 1$.

(2) Calculate the PCR at a surface point $\mathbf{P}_i$ from $\{\mathbf{P}_i\}$ along the direction passing through its neighboring sampled points:

(a) Find the sampled point set $\{\mathbf{Q}_j\}, j = 1,\ldots, m$, which is neighboring to $\mathbf{P}_i$. Set $j = 1$.

(b) Calculate the path direction angle β_{ij} along the path $\mathbf{P}_i\mathbf{Q}_j$, using Eq. (5.3).

(c) Transfer β_{ij} from global frame to local frame as ω_i at $\mathbf{P}_i$ and as ω_j at $\mathbf{Q}_j$, using Eq. (5.4).

(d) Specify cutter postures corresponding to ω_i at point $\mathbf{P}_i$ and ω_j at $\mathbf{Q}_j$ from their A-maps, using the approach introduced in Section 5.4.1.

(d) Calculate PCR_{ij} and Record it. Set $j = j + 1$. If $j \leq m$, go back to (b).

(3) Set $i = i + 1$. If $i \leq n$, go to (2).

(4) Uniformly sample $(\beta_{min}, \beta_{max})$ into $(N+1)$ angles, set $k = 0$.

(5) IF $k \leq N$, $\beta_k = \beta_{min} + (\beta_{max} - \beta_{min})(k/N)$; otherwise, go to (8).

(6) Obtain the PCR_{ik}s of all the sampled points $\{\mathbf{P}_i, i = 1, 2, \ldots, n\}$ along cutting direction β_k, using Eq. (5.5) based on PCR_{ij} at $\mathbf{P}_i$, and calculate the overall change rate $PCR_{\beta k} = \sum_{i=1}^{n} PCR_{ik}$ along β_k

(7) Set $k = k + 1$, go to (5).

(8) Take the direction that possesses the minimum $PCR_{\beta k}$ as the optimal one β_s for finishing the surface, e.g., $\beta_s = \{\beta_s \mid PCR_{\beta s} = \min\{PCR_{\beta k}\}\}$. Stop.

End

It is noted that, except the minimum PCR_{β}, the maximum value $PCR_{k\max} = \max\{PCR_{ik}\ i = 1,\ldots, n\}$, can also be taken into account for determining the optimal path direction. Firstly, to avoid the sharp large fluctuation of cutter posture between neighboring points, a threshold value of $PCR_{k\max}$ can be specified by the user. If the $PCR_{k\max}$ along a direction is greater than the threshold, this direction will be ruled out for selection. Secondly, when the values of $PCR_{\beta k}$ are close to each other for several path directions (close to minimum PCR_{β}), $PCR_{k\max}$ can then be employed as another evaluation factor. In this case, the direction that possesses the minimum $PCR_{k\max}$ among these directions can be chosen as the optimal one for finishing the surface. In this way, the largest posture jump between neighboring points is also minimized for smooth dynamic performance. In addition, it is worth mentioning that this work only

considers geometric characteristics in machining. Some technical concerns, such as the preferable tilting and orientation ranges, can be easily incorporated into this algorithm by limiting the A-map before the search starts.

5.5 Summary

This chapter presents a new method for determining the optimal direction of iso-planar tool-paths for 5-axis sculptured surface finish cut. The task is conducted after the process of cutter selection by making full use of the checking result from cuter selection process, i.e., cutter's A-map at every sampled surface point. Incorporated with the commonly used machining strategies, a searching algorithm has been proposed for finding the optimal path direction by considering the influence of cutter posture to cutting efficiency and accuracy. To consider these criteria simultaneously without having the actual tool-paths and the corresponding postures, cutter posture change rate (PCR) has been introduced as an evaluation factor. That is, the direction that possesses the minimum $\sum_{i=1}^{n} \mathrm{PCR}_i$ among the discrete sampled path directions is chosen as the optimal one for finishing the surface. The maximum machining efficiency and smooth tool dynamics could be ensured when cutting along the determined path direction. Further, the developed algorithm is much efficient since it excludes the numerical algorithm to detect and correct the interference in 5-axis machining.

CHAPTER 6

TOOL-PATH GENERATION PART 2:

CL DATA GENERATION

In traditional tool-path generation, considerable time is consumed to determine the optimal interference-free cutter postures from its accessible range since the algorithm for accessible range is numerical in general. This chapter presents a novel algorithm to determine the cutter posture in 5-axis tool-path generation based on the cutter's A-maps in an integrated and efficient manner. By utilizing A-maps at the sampled points, a nearly optimal posture at a CC point is obtained with an interpolation means using the A-maps at the neighboring sampled points. Integrated with this algorithm, a time-cost-saving approach is also proposed to produce the cutter location (CL) data, including the CC points and corresponding cutter postures, on iso-planar paths for finishing a sculptured surface.

6.1 Introduction

Since 1990's, enormous research work has been reported for the optimization of tool-paths generated for 5-axis sculptured surface machining. Choi *et al.* (1993) proposed a scheme to minimize the cusp heights left by searching the feasible posture regions for optimal CL data. Morishige and Takeuchi (1999) applied a 3-axis configuration space (*C*-space) to generate CL data on tool-paths reflecting the machining strategy such as smooth change in tool posture and the state of machining without considering the gouging. Chiou and Lee (2002a) presented an approach to

generate tool-paths by applying a machining potential field concept, which is constructed based on the largest feasible machining strip width and then the optimal cutting direction. It can be seen that optimization is achieved by searching an optimal interference-free cutter posture at every CC point. In fact, cutter posture is one of the critical issues addressed in 5-axis tool-path generation and is closely related to cutting accuracy and efficiency.

It is not an easy work to search for an interference-free cutter posture in 5-axis cutting because of complex machined surface shape and complicated cutter motion. Some researchers presented their works for finding interference-free postures based on a trial-and-error process (Li and Jerard, 1994; Jensen *et al.*, 2002), where the provisional determination of cutter posture is repeated till interference does not occur. However, the final cutter orientation is found by a non-deterministic approach and other potential CLs are not considered.

On the other hand, some researchers (Choi *et al.*, 1993; Lee, 1997; Xu *et al.*, 2002) tried to search for an optimal interference-free cutter posture based on the feasible cutter posture range. In Chapter 2, a point-based cutter accessibility (CA) algorithm has been proposed to evaluate the A-map of a cutter to a point on the part surface. It can be directly applied for the specification of cutter posture at a CC point in tool-path generation process by searching the A-map for an optimal cutter posture with respect to user-defined machining strategies. However, the CA algorithm is numerical in nature and computation time is a concern. As discussed in Chapter 2, the computation complexity of CA algorithm is O(km), in which m is the number of sampled points and k is the total number of discrete θs sampled over the rotational angle range ($\theta_{\min}$, $\theta_{\max}$). Meanwhile, the CA algorithm needs to be conducted at a large number of CC points and possibly at each potential point when iteratively

searching for the maximum path-interval value in tool-path generation. Thus, considerable time is required to determine the optimal interference-free cutter postures from its accessible range in tool-path generation. Experiments showed that more than 90% computation time is spent for the specification of cutter posture when applying CA algorithm in 5-axis tool-path generation.

Instead of the cumbersome CA algorithm, this chapter presents an efficient algorithm that uses the A-maps at all sampled points, to estimate the optimal interference-free cutter posture at a surface point in tool-path generation. Significant time-saving can be achieved when applying this algorithm into optimal tool-path generation processing.

6.2 A Quick Approach to Obtain Cutter Posture at a Point

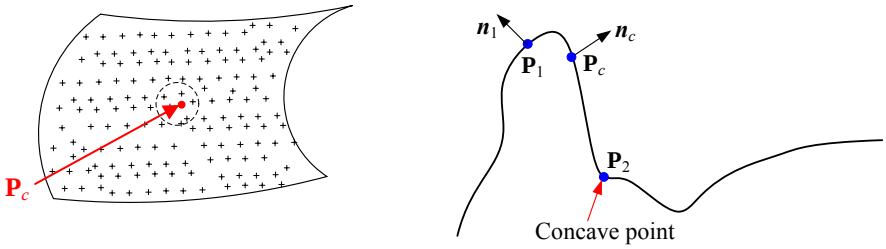
In this section, we propose an efficient approach to obtain the cutter posture at a surface point $\mathbf{P}_c$ along a feeding direction based on the A-maps at the sampled points. At $\mathbf{P}_c$, the cutter posture is obtained by interpolating the corresponding cutter postures at its neighboring sampled points, instead of using the cumbersome CA algorithm. This can be done by finding a small neighborhood of $\mathbf{P}_c$ from the sampled points, and identifying the cutter posture from the A-map at each neighboring point with respect to the feeding direction as discussed in Section 5.4.1. The following discussion will focus on finding neighboring sampled points, and determining the cutter posture at $\mathbf{P}_c$ by an interpolation approach, respectively.

6.2.1 Searching for neighboring sampled points of a surface point

In this subsection, a number of candidate neighboring points around the point of interest $\mathbf{P}_c$ will be firstly selected from the sampled surface database $\{\mathbf{P}_i, i = 1, \ldots,$

$n\}$, where n is the number of sampled points for cutter selection. First, the neighboring candidate points are found by searching for the p-nearest (e.g., p = 10) neighboring points $\mathbf{P}_j$, j = 1, …, p, around $\mathbf{P}_c$ in (u, v) parameter domain, as shown in Figure 6.1a. The sampled point data are arranged in a multi-layer (u, v) look-up table to save computing time for calculating the distance $d_j = \sqrt{(u_c - u_j)^2 + (v_c - v_j)^2}$. On the other hand, it is observed that not all neighboring candidate points are appropriate to participate in the interpolation, e.g., the points with a great jump of the surface geometric characteristics from $\mathbf{P}_c$. Some simple heuristic criteria have been introduced to rule out these points. For example, the candidate points with great change in normal direction will be ruled out, i.e., the point with $\boldsymbol{n}_j \bullet \boldsymbol{n}_c \le \cos(\alpha_n)$, where α_n is the desirable deviation angle predefined based on the experiment, such as 5º. Another geometric consideration is the sudden change of local surface shape (convex and concave), i.e., the candidate point locates in a shape region different from $\mathbf{P}_c$. This can be detected by observing the normal curvatures on the normal planes through both $\mathbf{P}_c$ and $\mathbf{P}_i$, i.e., $\kappa_j * \kappa_c < 0$ if in different shape regions. As shown in Figure 6.1b, point $\mathbf{P}_1$ is determined as having sudden normal direction change from $\mathbf{P}_c$ and labeled as non-neighboring point of $\mathbf{P}_c$. Similarly, point $\mathbf{P}_2$ is determined as sudden shape change from $\mathbf{P}_c$ and also labeled as non-neighboring point of $\mathbf{P}_c$.

At each neighboring candidate point $\{\mathbf{P}_j, j = 1, \ldots, p\}$, the cutter optimal interference-free postures (θ_j, λ_j) can be determined with respect to the feeding direction by using the approach introduced in Section 5.4.1. It is noted that, to ensure the avoidance of interference during machining, a pre-defined separation angle η_s, say 1º, is added to the tilting angle value, i.e., $\lambda_j = \lambda_j + \eta_s$, to separate the cutter outer surface and the critical part surface (Lee, 1997).



(a) the neighborhood of point $\mathbf{P}_c$ (b) non-neighboring points for interpolation

Figure 6.1 Neighboring candidate points of point $\mathbf{P}_c$

6.2.2 Determining the cutter posture at the point of interest

Now we have found the neighboring points $\{\mathbf{P}_j, j = 1, \ldots, p\}$ of $\mathbf{P}_c$ and the cutter postures (θ_j, λ_j) at these points corresponding to the feeding direction. In this subsection, we will propose an interpolation algorithm to determine the cutter posture at $\mathbf{P}_c$. Since the cutter posture (θ_j, λ_j) is represented in the local frame and the local frame varies for different points, it is necessary to transform these postures to a common frame to conduct the interpolation. In this work, the machine frame was chosen as this common frame and the cutter axis direction $\mathbf{T}_c$ is obtained as:

$$\mathbf{T}_c = \sum_{j=1}^{p} \frac{1}{\left|\mathbf{P}_c\mathbf{P}_j\right|^2}\mathbf{T}_j \Bigg/ \left|\sum_{j=1}^{p} \frac{1}{\left|\mathbf{P}_c\mathbf{P}_j\right|^2}\mathbf{T}_j\right| \tag{6.1}$$

Where $\mathbf{T}_j$ $(j = 1, \ldots, p)$ is the unit vector of cutter axis at point $\mathbf{P}_j$, and can be calculated from Eq. (5.1). $|\mathbf{P}_c\mathbf{P}_i|$ is the Euclidean distance between $\mathbf{P}_c$ and $\mathbf{P}_j$. The utilization of the distance square as a weight instead of the distance here is due to the interpolation result from our experiments, which compared the estimate value with the exact value from CA algorithm at $\mathbf{P}_c$.

The quick algorithm for determination of cutter posture at $\mathbf{P}_c$ has a complexity of O(kp), where p is the number of neighboring candidate points. Generally, p is very

small (such as 10). Compared with the approach to firstly evaluating the feasible range using the CA algorithm and then identifying the optimal cutter posture, with a complexity of O($km + k$) or simply O(km), the computation load is significantly reduced. It is noted that the computation analysis for the quick algorithm does not account the time for finding neighboring points, which has less computational complexity than the interpolation algorithm.

6.3 Optimal Tool-path Generation

With the cutting direction selected, the planar tool-paths can be generated by computing the intersection curves between cutting planes and the part surface. Thus, in iso-planar tool-path generation, the location of each cutting plane needs to be determined. For every single path, the CC points also need to be determined with their corresponding cutter postures as well. Figure 6.2 shows a systematic approach to guide the information process in tool-path generation. The first tool-path is set as the current tool-path and the location of its cutting plane is set to just off the surface edge with a small distance Δy_0 (e.g., $\Delta y_0 = 0.1R$, where R is the cutter major radius, as shown in Figure 6.3a) in the normal direction of cutting planes. Then the CC points are generated on the current path to satisfy the pre-defined profile tolerance and their corresponding postures are determined. Based on the CC points and the corresponding postures, the maximum allowable path interval between the next tool-path and the current one is calculated to locate the next cutting plane. This process is repeated to generate all paths for finishing the surface. After determining all CC point, the corresponding CL data in global frame can be easily generated according to the determined posture for a cutter (Choi and Jerard, 1998). For simple description, the CL data here refers to the location of a CC point and the corresponding cutter

orientation (θ, λ) in local frame. In the following discussion, we will address two critical issues in this process: (1) determine CC points on a single path; and (2) calculate the maximum allowable path interval between neighboring paths.

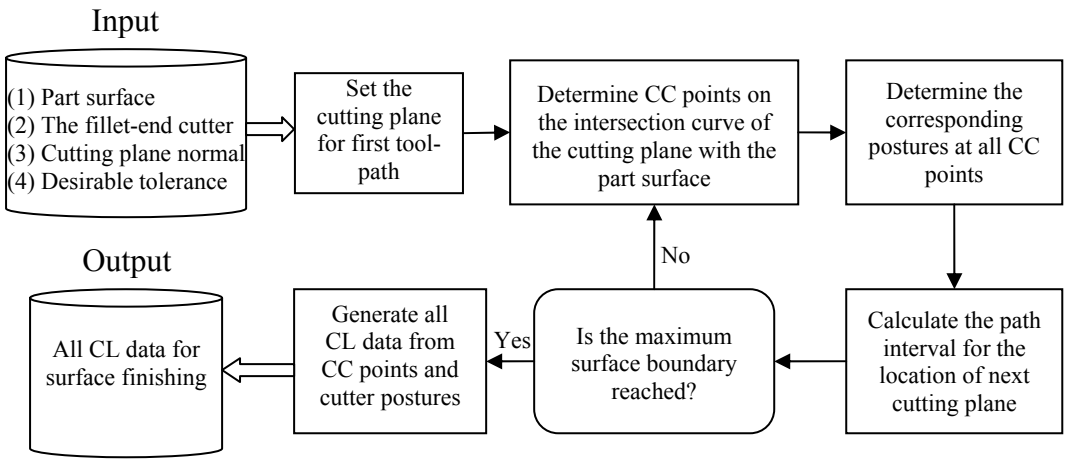


Figure 6.2 Flowchart of tool-path generation

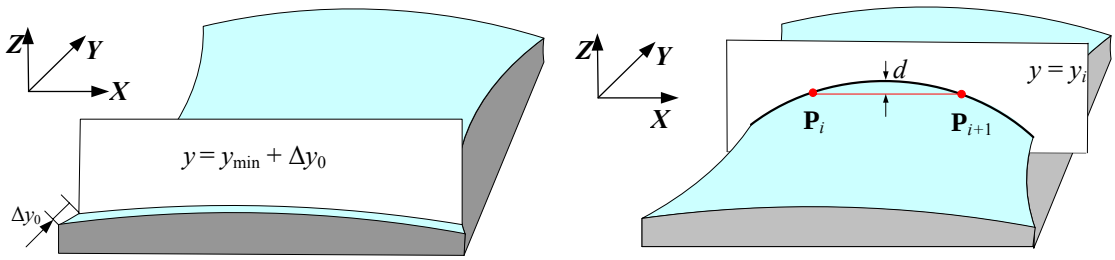
6.3.1 CC point generation on a single tool-path

A single path is produced by intersecting the surface with the current cutting plane. In this section, we will focus on how to obtain the discrete CC points on this path to satisfy the pre-defined profile tolerance. The first CC point can be obtained by intersecting the cutting plane and the surface boundary. Beginning from the first CC point, the process can be interpreted as "finding the next CC point on the same path such that the deviation of the tool motion trajectory is within the desirable tolerance τ from the part surface". Several interpolation techniques have been developed for the cutter motion in a NC milling machine, such as linear, circular or NURBs. The linear interpolation is employed here since it is the most commonly adopted in NC machines.

As shown in Figure 6.3b, at a CC point $\mathbf{P}_i(u_i, v_i)$, the next point on the 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 τ as:

$$\begin{aligned} &S_y(u_{i+1}, v_{i+1}) = y_i \\ &d(\mathbf{P}_i\mathbf{P}_{i+1}, S(u,v)) \le \tau \end{aligned} \tag{6.2}$$

Here we assume that the cutting plane is normal to ***Y*** axis as $y = y_i$ (if not, a trivial frame transformation can be used).



(a) First tool-path $y = y_{\min} + \Delta y_0$ (b) Next CC point on a single tool-path

Figure 6.3 The single planar tool-path

In theory, $\mathbf{P}_{i+1}(u_{i+1}, v_{i+1})$ can be determined by solving the equations in (6.2) using an analytical approach. In practice, however, the solving procedure is very complicated and even not applicable since Eqs (6.2) is a set of non-linear equations from NURBS surface representation. Thus, a numerical solution is adopted as follows:

(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 tool-path with the distance as the step-forward length from $\mathbf{P}_i$;

(3) Check whether the deviation between the tool trajectory and the tool-path is within the neighboring range ($(1-\delta)\ \tau \le d \le \tau$), where δ is the pre-defined value, such as 0.05.

If the condition is not satisfied, the step-forward length is accordingly changed, and steps (2) and (3) are repeated till the suitable $\mathbf{P}_{i+1}$ is obtained. The remaining of this section will address these three steps separately.

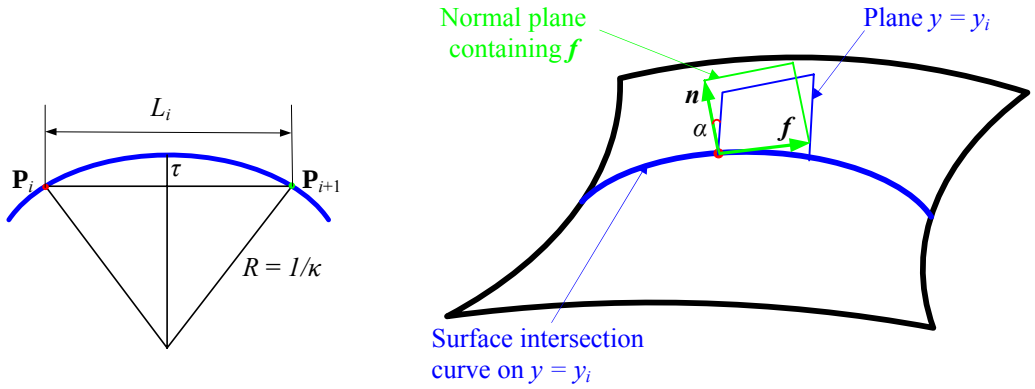
6.3.1.1 Setting the initial step-forward length

Apparently, the location of point $\mathbf{P}_{i+1}(u_{i+1}, v_{i+1})$ reflects the curve geometry feature at the current CC point $\mathbf{P}_i$ (u_i, v_i), such as the surface curve curvature. In general, interpolation points with high density should be resulted for the curve with higher curvature, while sparse interpolation points with lower curvature. Thus, the step forward length L_i for $\mathbf{P}_{i+1}$can be estimated based on the local surface geometry and the desirable profile tolerance τ, as shown in Figure 6.4a. Here, the geometry of surface curve on the cutting plane in the vicinity of $\mathbf{P}_i$ is approximated by a circular curve, and the curvature of the circle is that of the curve at $\mathbf{P}_i$. L_i is then given as:

$$L_i = \sqrt{8R\tau - 4\tau^2} = \sqrt{8\tau/\kappa - 4\tau^2} = 2\sqrt{\frac{\tau(2-\kappa\tau)}{\kappa}} \tag{6.3}$$

Where τ is the predefined profile tolerance and κ is the curvature of the path curve at $\mathbf{P}_i$. According to Meusnier theory (Guggenheimer, 1977), the value of κ can be calculated as:

$$\kappa = \kappa_n / \cos\alpha \tag{6.4}$$



(a) Step-forward length (b) surface curve curvature k_n at $\mathbf{P}_i$

Figure 6.4 Initial estimate of the step-forward length at $\mathbf{P}_i$

Where κ_n is the normal curvature of the surface curve on the normal plane containing $\boldsymbol{f}$, $\boldsymbol{f}$ is the positive tangent direction of the path curve at $\mathbf{P}_i$ on plane $y = y_i$, and α is the angle between the normal plane and the plane $y = y_i$, as shown in Figure 6.4b. From Eqs. (6.3) and (6.4), it can be seen that the calculation of normal curvature κ_n is needed. According to Faux and Pratt (1981), the first fundamental matrix G and the second fundamental matrix D of the part surface are given as:

$$G = \begin{pmatrix} \boldsymbol{S}_u \bullet \boldsymbol{S}_u & \boldsymbol{S}_u \bullet \boldsymbol{S}_v \\ \boldsymbol{S}_u \bullet \boldsymbol{S}_v & \boldsymbol{S}_v \bullet \boldsymbol{S}_v \end{pmatrix} \quad \text{and} \quad D = \begin{pmatrix} \boldsymbol{n} \bullet \boldsymbol{S}_{uu} & \boldsymbol{n} \bullet \boldsymbol{S}_{uv} \\ \boldsymbol{n} \bullet \boldsymbol{S}_{uv} & \boldsymbol{n} \bullet \boldsymbol{S}_{vv} \end{pmatrix} \tag{6.5}$$

Where $\boldsymbol{n}$ is the normal vector, $\boldsymbol{S}_u$ and $\boldsymbol{S}_v$ are the first-order partial derivatives, and $\boldsymbol{S}_{uu}$, $\boldsymbol{S}_{uv}$ and $\boldsymbol{S}_{vv}$ are the second-order partial derivatives at the point, respectively. The definition of the normal curvature is:

$$\kappa_n = \frac{F^T D F}{F^T G F} \tag{6.6}$$

Where $F = [\boldsymbol{S}_u \cdot \boldsymbol{f}, \boldsymbol{S}_v \cdot \boldsymbol{f}]^T$. Defining $\sigma = \boldsymbol{S}_u \cdot \boldsymbol{f} / \boldsymbol{S}_v \cdot \boldsymbol{f}$ and expanding Eq. (6.6), the curvature κ_n is calculated as:

$$\kappa_n = \frac{d_{11}\sigma^2 + 2d_{12}\sigma + d_{22}}{g_{11}\sigma^2 + 2g_{12}\sigma + g_{22}} \tag{6.7}$$

With this definition, the curvature κ_n is positive when the curve is turning toward the positive direction of the surface normal. Otherwise, κ_n is negative. Incorporating Eqs. (6.4) and (6.7) into Eq. (6.3), the initial estimate of step-forward length is obtained.

6.3.1.2 Determining the next estimated CC point

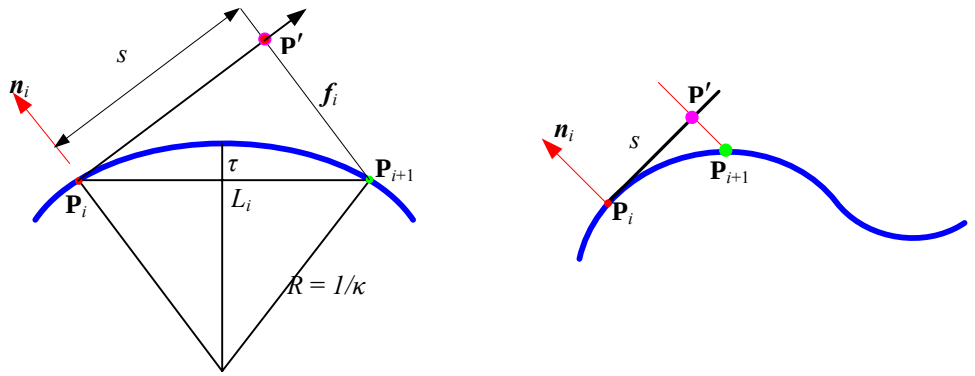
Based on the estimated value of L_i, we proceed to search for the next CC point P_{i+1} on the path with the distance L_i from $\mathbf{P}_i$ along the cutting direction. As discussed above, direct analytical solution is very complicated and might not be applicable. The method used by Hwang (1992) is used here to get the estimated next point P_{i+1}.

Firstly, as shown in Figure 6.5a, the tangent distance s for the next step is calculated based on approximated circular curve as:

$$s = (R - \tau)L_i / R \tag{6.8}$$

And the point $\mathbf{P}'$ with distance s along the tangent direction $\boldsymbol{f}_i$ is:

$$\mathbf{P}' = \mathbf{P}_i + s\boldsymbol{f}_i \tag{6.9}$$



(a) Tangent distance for next CC point (b) Next CC point

Figure 6.5 Determination of next estimated point $\mathbf{P}_{i+1}$

From point $\mathbf{P}'$, $\mathbf{P}_{i+1}(u_{i+1}, v_{i+1})$ is obtained to be the intersection point of the part surface with a line through $\mathbf{P}'$ and along $\boldsymbol{n}_i$ direction (see Figure 6.5b), i.e.,

$$\mathbf{P}_{i+1} = \mathbf{P}' + l\boldsymbol{n}_i \tag{6.10}$$

Where l is the distance of $\mathbf{P}_{i+1}$ from $\mathbf{P}'$, and the value of l is calculated using the method by Scherrer and Hillberry (1978). It is noted that the sign of l can be positive or negative in Eq. (6.10).

6.3.1.3 Calculating deviation between line segment and path curve

To evaluate the maximum deviation, between line segment $\mathbf{P}_i\mathbf{P}_{i+1}$ and the path curve, we need to find the corresponding point $\mathbf{P}(u, v)$ at which the deviation reaches its maximum, as shown in Figure 6.6. Some conditions should be satisfied for this particular point $\mathbf{P}$. Firstly, $\mathbf{P}$ locates on the path curve and thus cutting plane equation $y = y_i$ should be satisfied. Secondly, from geometry analysis, one tangent line at point $\mathbf{P}$ should be parallel to $\mathbf{P}_i\mathbf{P}_{i+1}$, i.e., the normal $\boldsymbol{n}(u, v)$ at $\mathbf{P}$ is perpendicular to $\mathbf{P}_i\mathbf{P}_{i+1}$. These two conditions can be expressed by:

$$\begin{aligned} &\boldsymbol{S}_y(u,v) = y_i \\ &\boldsymbol{n}(u,v) \bullet \mathbf{P}_i\mathbf{P}_{i+1} = 0 \end{aligned} \tag{6.11}$$

Here, the secant numerical approach (Press *et al*., 2002) is employed to solve this non-linear equation group (6.11) to get (u, v) value at $\mathbf{P}$**.** After obtaining point $\mathbf{P}$, the deviation distance d between $\mathbf{P}_i\mathbf{P}_{i+1}$ and path curve is represented as:

$$d = \left| (\mathbf{P} - \mathbf{P}_i) - \left[(\mathbf{P} - \mathbf{P}_i) \bullet \frac{\mathbf{P}_{i+1} - \mathbf{P}_i}{\left|\mathbf{P}_{i+1} - \mathbf{P}_i\right|} \right] \frac{\mathbf{P}_{i+1} - \mathbf{P}_i}{\left|\mathbf{P}_{i+1} - \mathbf{P}_i\right|} \right| \tag{6.12}$$

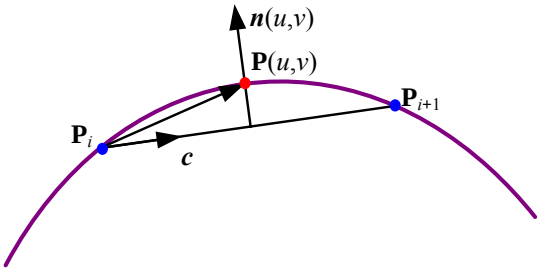


Figure 6.6 The deviation of the chord length from the path curve on plane $y = y_i$

6.3.1.4 The overall algorithm

The algorithm to search for the CC points on a single tool-path is described as follows:

Algorithm: ***Searching for the CC points on a single tool-path***

Input:

(1) *A NURBS surface with a set of trimmed patches* $\mathbf{S}_k(u,v)$, $k = 1, \ldots, l$

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

(3) *The desirable profile tolerance* τ

Output: *A set of CC points on the path*

Begin

(1) Search for the boundary points of the surface curve segments C_j ($j = 1,\ldots, q$) on plane $y = y_i$. Set $k = 0$.

(2) Pick curve section C_k and set one boundary point as the current CC point $\mathbf{P}_i$.

(3) Estimate the step-forward length L_i, using the algorithm introduced in Section 6.3.1.1.

(4) Search for the next CC point $\mathbf{P}_{i+1}$ according to L_i, using the algorithm introduced in Section 6.3.1.2.

(5) Calculate the deviation d of $\mathbf{P}_i\mathbf{P}_{i+1}$ from the path curve, using the algorithm introduced in Section 6.3.1.3.

(6) IF $(1-\delta)\ \tau \le d \le \tau$, where δ is the pre-defined tolerance (such as 0.05), accept the resultant CC point $\mathbf{P}_{i+1}$. Otherwise, decrease L_i if $d > \tau$ or increase L_i if $d < (1-\delta)\ \tau$, and go to (4).

(7) Save the point $\mathbf{P}_{i+1}$. If $\mathbf{P}_{i+1}$ does not reach the curve segment boundary, set it as the current one and go back to (3).

(8) Set $k = k + 1$. IF $k \le q$, go to (2).

(9) Output all CC points on the path. Stop.

End

It is noted that there might exist separated curve segments broken in between on the current cutting plane when generating iso-planar paths. At each CC point on the current path, the cutter posture can be determined with regard to the feeding direction using the method introduced in Section 6.2. After obtaining all CL data on a single cutting plane, the next step is to calculate the largest allowable path interval between the next path and current path.

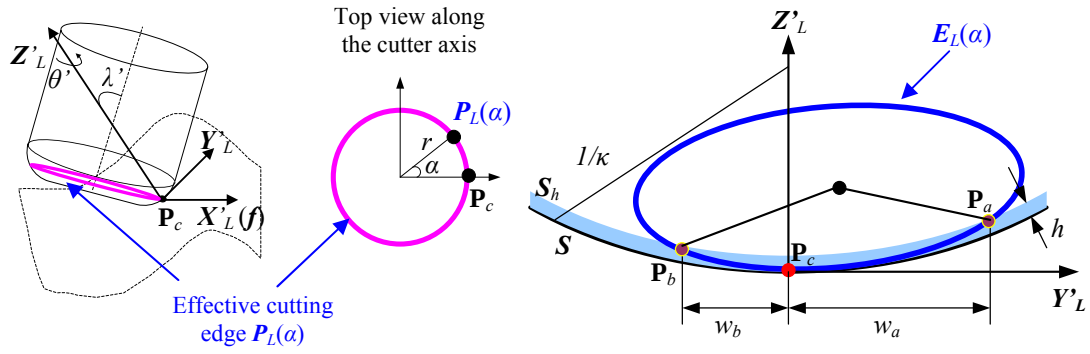
6.3.2 Evaluation of the path interval between adjacent paths

Based on the CL data (including position and posture) on the current path, this subsection presents the method to calculate the maximum allowable path interval Δy for setting the next cutting plane. This can be achieved by calculating path interval Δy_j at each CC point on the current path with the next path and taking the minimum one as Δy, i.e., $\Delta y = \min\{\Delta y_j \mid j = 1, \ldots, n_i\}$, where n_i is the number of CC points on the current cutting path. At a CC point, the calculation of Δy_j is conducted based on the

concept of machining strip width. In the following discussion, the computation of machining strip width is firstly presented before the detailed algorithm for calculating the path interval.

6.3.2.1 Calculation of machining strip width

Lee (1998) presented an algorithm to calculate the machining strip width by investigating the effective cutting edge during machining with a flat-end cutter. Here, we extend this algorithm to evaluate the machining strip of a fillet-end cutter. To simplify the presentation, a new local frame $\boldsymbol{H}'$ ($\mathbf{P}$, $\boldsymbol{X}'_L$, $\boldsymbol{Y}'_L$, $\boldsymbol{Z}'_L$) is defined based on the feeding direction $\boldsymbol{f}$ instead of maximum principle direction (see Figure 6.7a). $\boldsymbol{H}'$ is similar as frame $\boldsymbol{H}$ ($\mathbf{P}$, $\boldsymbol{X}_L$, $\boldsymbol{Y}_L$, $\boldsymbol{Z}_L$) in Figure 2.1. It can be seen that a cutter posture (λ, θ) in $\boldsymbol{H}$ is represented in $\boldsymbol{H}'$ as ($\lambda' = \lambda$, $\theta' = \theta + \Delta\theta$), in which $\Delta\theta$ is the angle between the maximum principle direction and $\boldsymbol{f}$. Here, we also denote the orientation angles as (λ, θ) in $\boldsymbol{H}'$ for simple description.



(a) Local frame according to feeding direction $\boldsymbol{f}$ (b) Machining strip width at a CC point

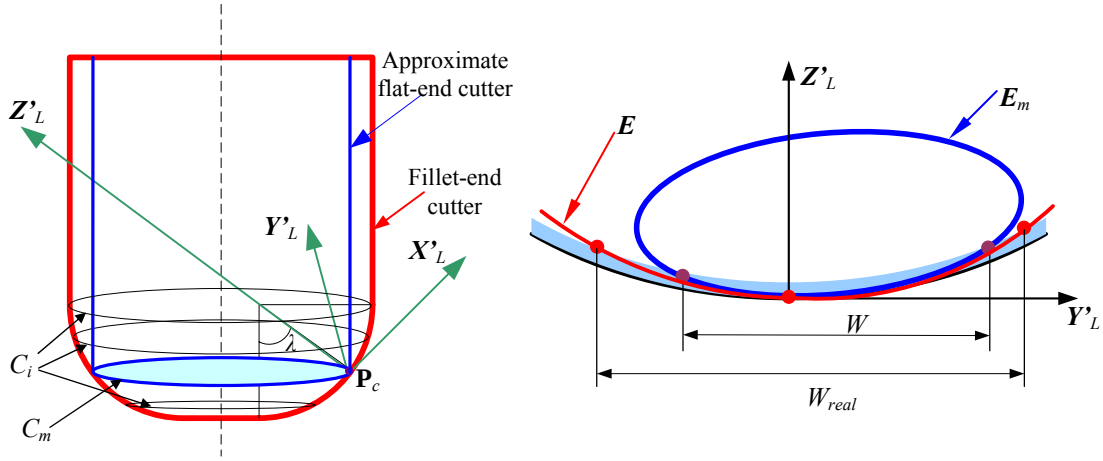
Figure 6.7 Evaluation of machining-strip width at point $\mathbf{P}_c$

When the cutter is moving through a point $\mathbf{P}_c$ along a path, the machined surface must be bounded within the offset surface, a surface with distance as the scallop height tolerance h from the part surface. In this work, machining strip is

defined as the cutter swept volume curve on $\boldsymbol{Y}'_L$ - $\boldsymbol{Z}'_L$ plane and limited by the offset surface, i.e., elliptical cross section $\mathbf{P}_b\mathbf{P}_c\mathbf{P}_a$ as shown in Figure 6.7b. Several techniques have been developed on how to obtain the machining strip width, which can be classified as: (1) analytical swept volume approach; and (2) instantaneous approach. The former considers the tool motion in a continuous mode and calculates the machining strip width based on the swept volume by tool motion. Although this approach is ideal in theory, it is often very complex to represent the swept volume in an analytical way (Chiou, 2004). Furthermore, the calculation of machining strip width requires calculating the distance between the swept volume and the part surface. A set of non-linear equations is needed, which involves iterative numerical solution and may face possible convergence problem. The latter approximates the swept volume by the instantaneous cutter surface along the cutter motion path. It is a reasonable approximation since the swept volume is composed of the cutter surface at infinite number of points along tool-path, and it is much easier to be implemented. In this work, the instantaneous approach is adopted for the evaluation of machining strip width.

Sheltami *et al.* (1998) pointed out that the swept volume of a fillet-end cutter at a CC point $\mathbf{P}_c$ can be estimated by the cutter's effective cutting edge, a circular curve through $\mathbf{P}_c$ on the cutter filleted portion and normal to the cutter axis (see Figure 6.7a). When the cutter is posed with (λ, θ) in local frame, the circle radius of the cutting edge can be given as $r = (R - r_f) + r_f \times \sin\lambda$, where R is the major radius of the cutter and r_f the minor radius. Here we approximate the fillet-end cutter motion as that of a flat-end cutter with radius r. In addition, this approximation is on the safe side when applying for the evaluation of machining strip width in tool-path generation. It can be proved as follows. As shown in Figure 6.8a, the cutting portion of a fillet-

end cutter can be represented as a set of parallel cutting circles $\{C_i, i = 1,\ldots,m,\ldots,\infty\}$, where $R(C_m) = r$. The cutting shape of the fillet-end cutter then can be represented as $\boldsymbol{E} = B\left(\bigcup \boldsymbol{E}_i, i = 1,...,m,...,\infty\right)$, where $\boldsymbol{E}_i$ is the cutting shape of the flat-end cutter with radius as $R(C_i)$, and B is the outer boundary operator. Therefore, $\boldsymbol{E}_m$ from the approximate flat-end cutter must be totally contained inside $\boldsymbol{E}$ from the fillet-end cutter, as illustrated in Figure 6.8b on plane $\boldsymbol{Y'}_L$-$\boldsymbol{Z'}_L$. When applying this flat-end cutter into the evaluation of machining strip width, the resultant result W must be smaller than the actual one W_{real} (see Figure 6.8b). This will result in a little tighter machining accuracy than desired, and will not have any negative effect on maintaining the accuracy in tool-path generation.



(a) Fillet-end and approximate flat-end cutter (b) comparison of machining strip width

Figure 6.8 A fillet-end cutter ant its approximate flat-end cutter

For flat-end cutter milling, a similar algorithm developed by Lee (1998) is employed to evaluate the machining strip width based on the instantaneous technique. In the local frame, the effective cutting edge of a cutter (see Figure 6.7a) with orientation (λ, θ) is expressed as (Sheltami *et al.*, 1998):

$$\boldsymbol{P}_L(\alpha) = \begin{pmatrix} r\cos\lambda\cos\theta\cos\alpha - r\sin\theta\sin\alpha - r\cos\lambda\cos\theta \\ r\cos\lambda\sin\theta\cos\alpha + r\cos\theta\sin\alpha - r\cos\lambda\sin\theta \\ -r\sin\lambda\cos\alpha + r\sin\lambda \end{pmatrix} \tag{6.13}$$

Where $\alpha \in [0°, 360°]$ is the angle parameter of the point on the cutting edge, as shown in Figure 6.7a. The effective cutting shape on $\boldsymbol{Y'}_L$–$\boldsymbol{Z'}_L$ plane (See Figure 6.7b) is found from Eq. (6.13) as:

$$\boldsymbol{E}_L(\alpha) = \begin{pmatrix} 0 \\ r\cos\lambda\sin\theta\cos\alpha + r\cos\theta\sin\alpha - r\cos\lambda\sin\theta \\ -r\sin\lambda\cos\alpha + r\sin\lambda \end{pmatrix} \tag{6.14}$$

As shown in Figure 6.7b, the machining strip width is referred as the distance of $\mathbf{P}_a\mathbf{P}_b$ along the axis $\boldsymbol{Y'}_L$, where $\mathbf{P}_a$ and $\mathbf{P}_b$ are the intersection points of the effective cutting shape and the offset part surface $\boldsymbol{S}_h$ on $\boldsymbol{Y'}_L$–$\boldsymbol{Z'}_L$ plane. Using the second order Tailor expansion, the surface curve on $\boldsymbol{Y'}_L$–$\boldsymbol{Z'}_L$ plane can be approximated as $z_L = ½\ \kappa_n y_L^2$ (DoCarmo, 1976), where κ_n is the curvature of surface curve on the normal plane, which can be calculated by Eqs (6.5) ~ (6.7) after specifying the cutting direction. On the other hand, the point on the exact offset surface $\boldsymbol{S}_h$ is $\mathbf{P}_h = \mathbf{P}_s + h\boldsymbol{n}(\mathbf{P}_s)$. The definition of $\boldsymbol{S}_h$, though strict, is inconvenient to calculate the intersection point $\mathbf{P}_a$ and $\mathbf{P}_b$ because of complicated representation of $\boldsymbol{n}(\mathbf{P}_s)$. The problem may be substantially simplified if one uses a fixed direction, defined by the unit normal vector $\boldsymbol{n}(\mathbf{P}_c)$ at $\mathbf{P}_c$ instead of dynamically changed normal vector $\boldsymbol{n}(\mathbf{P}_s)$, to measure the distance between the surface, i.e., the point on the approximate offset surface $\boldsymbol{S}'_h$ as $\mathbf{P}_h = \mathbf{P}_s + h\boldsymbol{n}(\mathbf{P}_c)$. Yoon *et al*. (2003) suggested that it is on the safe side working with this approximate version. On plane $\boldsymbol{Y'}_L$–$\boldsymbol{Z'}_L$, the $\boldsymbol{S}'_h$ point is then given as:

$$z_L = ½\ \kappa_n\, y_L^2 + h \tag{6.15}$$

Combining Eqs. (6.14) and (6.15), we can obtain the intersection point $\mathbf{P}_a$ and $\mathbf{P}_b$, and then machining strip width w_a and w_b along the axis $\boldsymbol{Y}'_L$ direction at the CC point $\mathbf{P}_c$.

6.3.2.2 Evaluation of the path interval

As shown in Figure 6.9, path interval Δy_j at each CC point $\mathbf{P}_j$ is firstly calculated, and the minimum one is then taken as Δy, i.e., $\Delta y = \min\{\Delta y_j \,|\, j = 1, \ldots, n_i\}$, where n_i is the number of CC points on the current cutting path. The following discussion will focus on the evaluation of Δy_j at a $\mathbf{P}_j$.

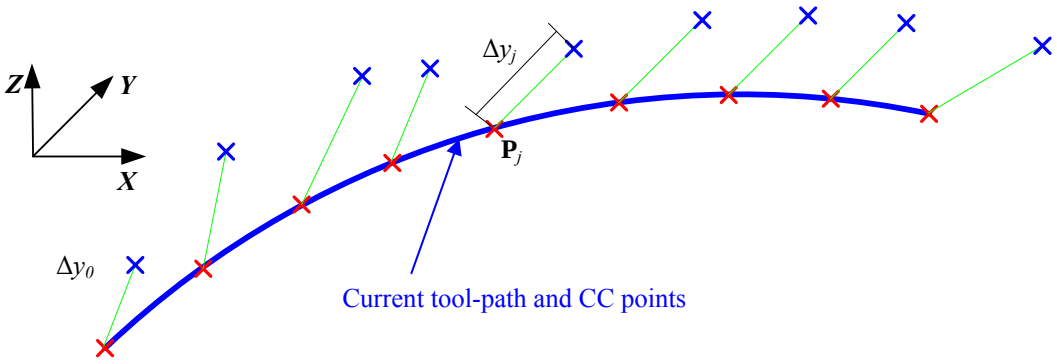


Figure 6.9 Evaluation of the path interval

To achieve the required surface accuracy, the machining strips of the two adjacent paths have to overlap each other to satisfy the scallop height tolerance h, as shown in Figure 6.10a. 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 $\boldsymbol{Y}$-axis as follows.

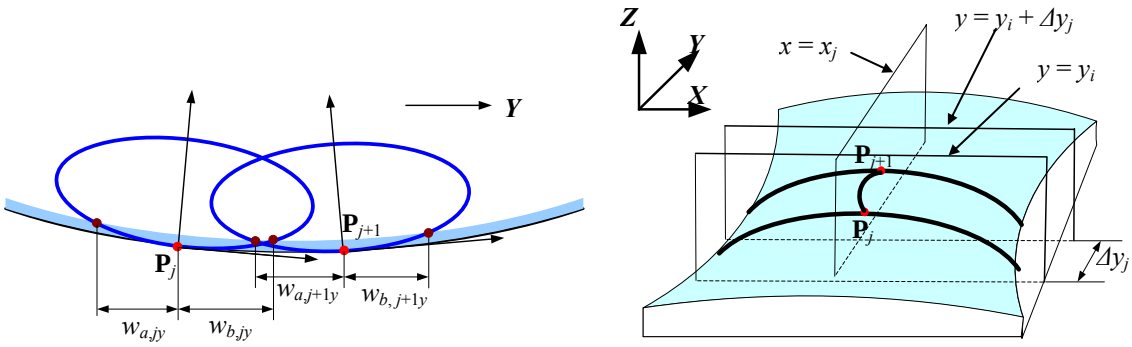
$$\Delta y_j \le \omega_{b,jy} + \omega_{a,j+1y} \le (1+\varepsilon)\Delta y_j \tag{6.16}$$

Where $\omega_{b,jy}$ 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, $\mathbf{P}_{j+1}$ is the corresponding CC point on the next path, and ε is pre-defined value such as 0.05.

An iterative checking algorithm is designed to find $\mathbf{P}_{j+1}$ and obtain 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 6.10b):

$$\begin{aligned} S_x(u_{j+1}, v_{j+1}) &= x_j \\ S_y(u_{j+1}, v_{j+1}) &= y_i + \Delta y_j \end{aligned} \tag{6.17}$$

Where (x_j, y_i, z_j) is the coordinates of point $\mathbf{P}_j$ in the machine frame.



(a) CC points on adjacent paths

(b) CC point $\mathbf{P}_j$ (x_j, y_i, z_j) and corresponding point $\mathbf{P}_{j+1}$ on the next path

Figure 6.10 Calculation of path interval between two tool-paths at a CC point $\mathbf{P}_j$

Secondly, the cutter posture is specified at point $\mathbf{P}_{j+1}$ with regard to its feeding direction using the algorithm in Section 6.2, and the machining strip width is calculated using the algorithm in Section 6.3.2.1. If the condition (6.16) is not satisfied, The value of Δy_j is accordingly changed, and this process of finding $\mathbf{P}_{j+1}$ from Eq. (6.17) and evaluating the machining strip width at $\mathbf{P}_{j+1}$ is repeated till the maximum allowable Δy_j is obtained. The detailed algorithm is given as follows:

Algorithm: ***Calculating the path-interval at a CC point***

Input:

(1) *A CC point* $\boldsymbol{P}_c$ *and the corresponding posture*

(2) *A-maps at sampled surface points*

(3) *Desirable scallop height tolerance h*

Output: *the path interval at* $\boldsymbol{P}_c$

Begin

(1) Calculate the machining strip width w_a and w_b at $\mathbf{P}_c$, and convert them as w_{ay} and w_{by} along ***Y***-axis, using the algorithm in Section 6.3.2.1. Set $\Delta y = w_{ay} + w_{by}$.

(2) Determine the adjacent estimated point $\mathbf{P}_{c1}$ with Δy, using Eq. (6.17).

(3) Determine the cutter posture at $\mathbf{P}_{c1}$ based on A-maps at sampled surface points, using the algorithm introduced in Section 6.2.

(4) Calculate the machining strip width w_{a1} and w_{b1} at $\mathbf{P}_{c1}$, and convert them as w_{ay1} and w_{by1} along ***Y***-axis.

(5) Adaptively adjust the value of Δy according to the value of $w_{by} + w_{ay1}$ as:

IF $\Delta y \le w_{by} + w_{ay1} \le (1 + \varepsilon)\, \Delta y$, go to (6).

IF $\Delta y > w_{by} + w_{ay1}$, decrease Δy with a small step and go to (2).

IF $w_{by} + w_{ay1} > (1 + \varepsilon)\, \Delta y$, increase Δy with a small step and go to (2).

(6) Output Δy as the path interval at point $\mathbf{P}_c$. Stop.

End

It can be seen that the determination of cutter posture at a surface point needs to be conducted for several times (same as the number of iterations for the estimated points) in this algorithm. If the traditional approach is employed to determine the cutter posture, i.e., evaluate the A-map based on the surface geometry and then search for the optimal posture, the computation load is very heavy. On the other hand, it is

easy and efficient to determine the cutter posture at any surface point with the proposed interpolation technique in Section 6.2 using the A-maps of all sampled points.

6.3.3 The overall algorithm for tool-path generation

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

Algorithm: ***Generating tool-paths to finish the surface***

Input:

(1) *A NURBS surface S (u,v)*

(2) *A fillet-end cutter (R, r_f)*

(3) *The normal vector* ***n*** *for path cutting planes*

(4) *Machining profile tolerance* τ *and h*

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

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

Begin

(1) Set a new workpiece frame with axis ***Y*** aligned with ***n***. Transform the control points of $\boldsymbol{S}(u,v)$ from machine frame to the workpiece frame. And find the minimum $y_{\min}$ and maximum $y_{\max}$ for $\boldsymbol{S}(u,v)$. 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.

(2) Plan the path (one or more segments) on the plane $y = y_i$ and put the CC points

into the set $\{\mathbf{P}_{ij}, j = 1,\ldots, n_i\}$, using the algorithm in Section 6.3.1.

(3) Obtain the cutter posture at each CC point, using the algorithm in Section 6.2.

(4) Calculate the path interval Δy_{ij}, $j = 1, \ldots, n_i$, at each CC point on the current path. Set the path interval $\Delta y_i = \min\{\Delta y_{ij}, j = 1,\ldots, n_i\}$, using the algorithm in Section 6.3.2.

(5) Set $y_i = y_i + \Delta y_i$. If $y_i <= y_{\max}$, go back to (2).

(6) Output the CL data.

End

This adaptive optimization approach for tool-path generation may not be new. However, it involves the computation to determine the cutter posture at a large number of CC points and the estimated points for path interval evaluation. Without the presence of the A-maps of the sampled points, it would be practically impossible to implement it by applying CA algorithm to determine the cutter postures in the procedure.

6.4 Summary

This chapter presents an approach to generate the iso-planar paths for finishing a sculptured surface on a milling machine, based on the A-maps of the sampled surface points from cutter selection task. An efficient algorithm is proposed to search for the nearly optimal interference-free cutter posture at a surface point by interpolating those at neighboring sampled points. This interpolation technique is effective to determine the cutter posture at a point, since the sampled points in cutter selection can be much denser than the CC points. Furthermore, the comparison of computation complexity showed that the proposed algorithm is much more efficient than the traditional approach. Based on this quick algorithm for cutter posture, an

adaptive algorithm has been presented to generate the CL data on the tool-paths such that the maximum step-forward length and path interval are achieved while satisfying the desirable machining tolerance.

CHAPTER 7

RESULTS AND DISCUSSION

This work addresses the critical issues related to process planning for 5-axis sculptured surface finish milling, including cutter selection and tool-path generation. The machining constraints consist of the prevention of machining problems (such as machine limits, gouging and collision) and the requirement of surface criteria (such as the profile tolerance for gouging and scallop height). The concept of A-map, i.e., accessible range to a surface point without machining problems, is presented as the basis of the whole process planning procedure. Firstly, in theory, cutter selection can be conducted by evaluating A-map at each point for each cutter in a cutter list till the one with A-map available at all surface points. However, extensive computational time is involved in this way. To relieve heavy computation load, several algorithms are developed, including surface decomposition to reduce the number of checked points for a cutter and cutter accessibility comparison to avoid the redundant checking among cutters. Secondly, since the density of the sampled points can be much higher than that of CC points (used for subsequent tool-path generation), the checking result from cutter selection has also been used for the processing of tool-path generation, including determination of path direction and generation of tool-paths. In this way, the process planning problems are solved in an integrated and efficient manner.

This chapter introduces the work on the illustrated example and software simulation of process planning in 5-axis NC machining of sculptured surfaces by employing the aforementioned algorithms, including calculation of A-map at a

surface point, surface decomposition for a single cutter, cutter accessibility comparison between cutters, determination of cutting direction and tool-path generation. The software utilized for simulation is VERICUT®, which simulates the CNC machines as they behave on the shop floor and detects errors and potential problems before the program goes out to the shop.

7.1 A-map at a Surface Point

A prototype system of process planning for multi-axis sculptured surface machining has been developed in Visual C++ environment and with OpenGL for display. In this section, the calculation of A-map at a surface point is presented based on machining constraints, including machine limits, local-gouging, rear-gouging and global-collision.

7.1.1 Cutter accessibility algorithm at a surface point

Figure 7.1 illustrates a NURBS surface, including concave, convex and saddle portions. The data for the NURBS surface is given in Appendix A. Table 7.1 shows the relevant data for this NURBS surface. The peaks and valleys are exaggerated to validate the robustness of the proposed approach. On the surface, there is an altitude change of 47.1 mm over a linear distance of just 33.1 mm from the apex of a peak to the nadir of the closest valley.

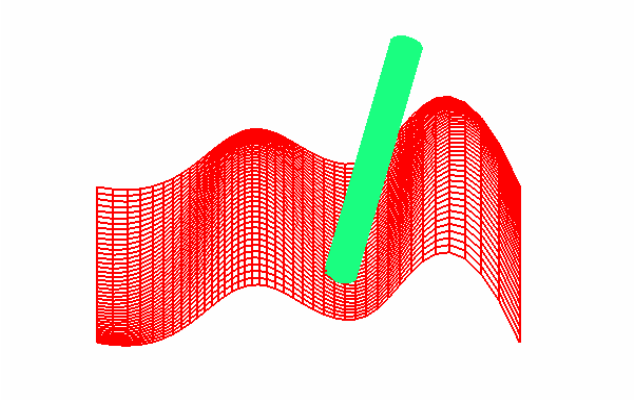


Figure 7.1 A NURBS sculptured surface

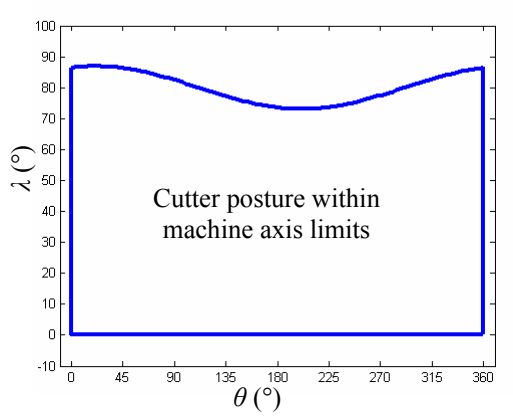
Table 7.1: The surface data

Parameter (u,v)	k_{min} (1/mm)	k_{max} (1/mm)	Local surface geometry
(0.00, 0.00)	$6.2103*10^{-3}$	$1.7957*10^{-2}$	concave
(0.10, 0.10)	$6.3099*10^{-4}$	$4.1308*10^{-2}$	concave
(0.20, 0.10)	$-7.7598*10^{-3}$	$1.6403*10^{-2}$	saddle
(0.40, 0.40)	$-1.1070*10^{-1}$	$-1.0052*10^{-2}$	convex
(0.70, 0.30)	$2.2648*10^{-3}$	$1.2482*10^{-1}$	concave
(0.88, 0.50)	$-5.9461*10^{-2}$	$-6.6877*10^{-3}$	convex

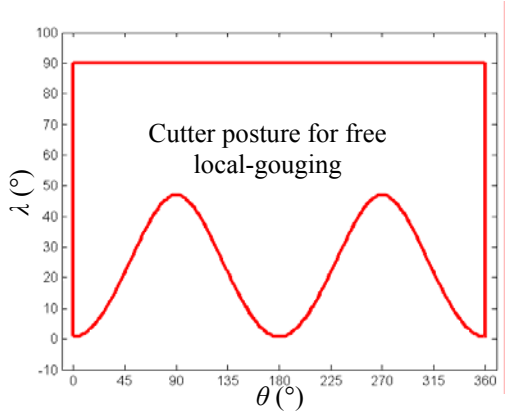
A surface point (172.274, 59.9291, 12.6229) with (u, v) = (0.7, 0.3) was chosen for testing the CA algorithm to find the A-map of a fillet-end cutter (R, r_f, L) = (6mm, 0.5mm, 80mm). The machine was chosen to be of C-A type (Suh and Kang, 1995), and the two revolute ranges are (-90°, 90°; -80°, 80°). The surface was sampled to 201×201 points uniformly along u and v, respectively, for interference checking. Figure 7.2a shows the accessible range satisfying machine limits by implementing the algorithms in (Xu *et al.*, 2002).

At the same point, by implementing algorithms developed in Section 2.3.1, the accessible range of local-gouging is obtained and depicted in Figure 7.2b. It is worth noting that the minimum and maximum values of the tilting angle λ were

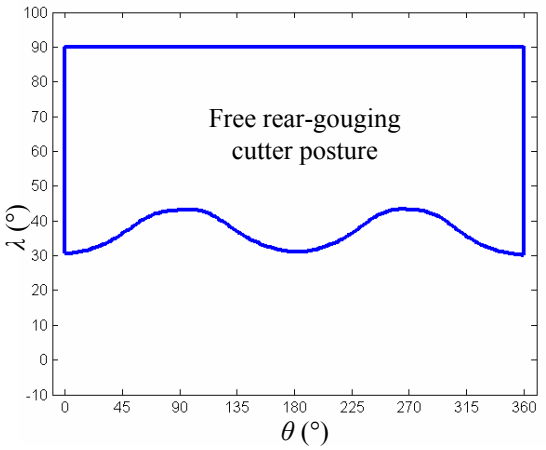
assumed as 0° and 90°, respectively. Every point within the range represents a cutter posture with free local-gouging.



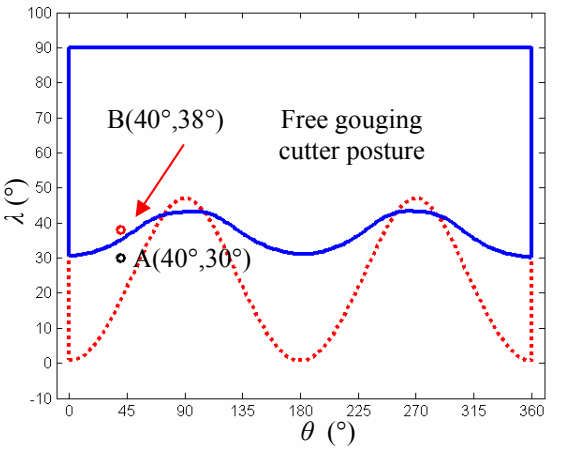
(a) Accessible range for machine axis limits



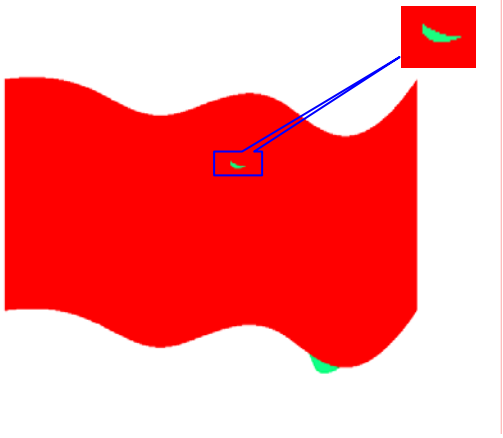
(b) Accessible range for local-gouging avoidance



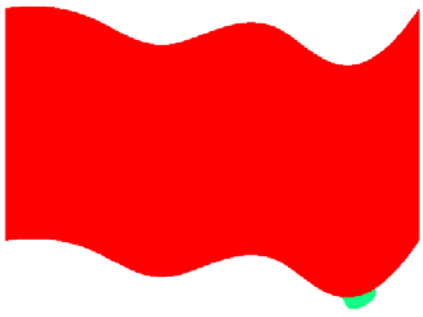
(c) Accessible range for rear-gouging avoidance



(d) Accessible range for gouging avoidance



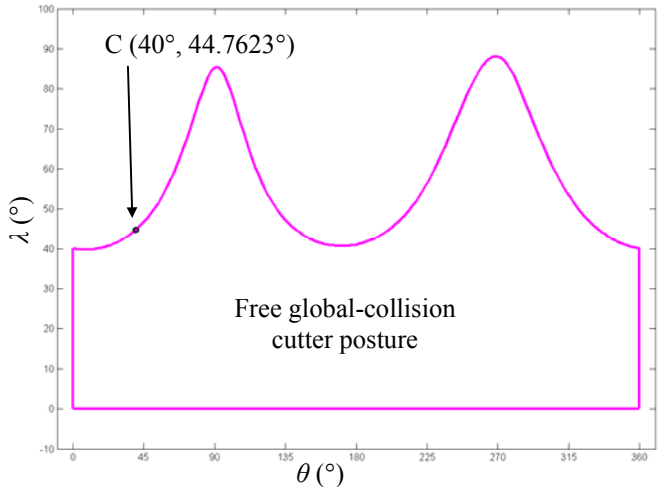
(e) A cutter posture with gouging (angle pair A)



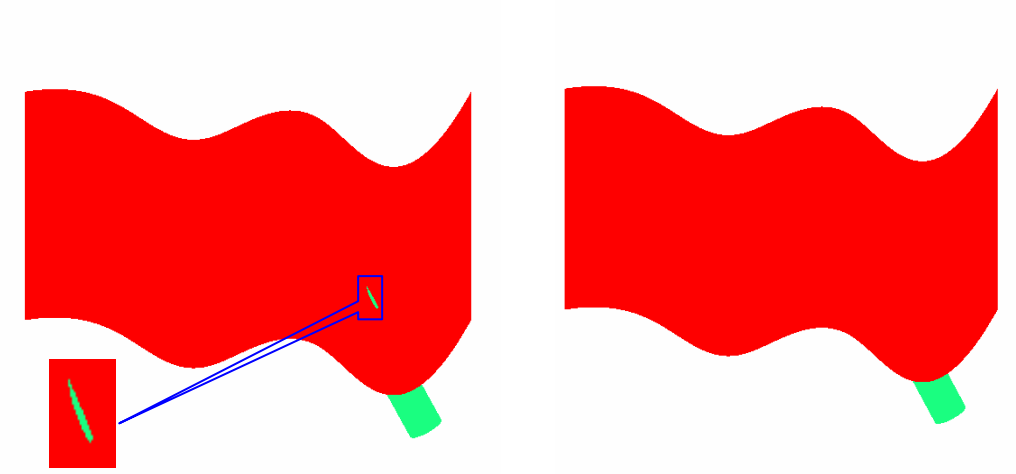
(f) A cutter posture with free gouging (angle pair B)

Figure 7.2 Cutter accessible ranges for machine limits and gouging avoidance

Similarly, at the same CC point, by implementing the algorithms developed in Section 2.3.2, the accessible range for rear-gouging avoidance is evaluated and depicted in Figure 7.2c. To illustrate the gouging (local and global) problem, Figure 7.2b and 7.2c are placed together as shown in 7.2d. Any point beyond the common accessible range represents a cutter posture that leads to gouging. Figure 7.2e shows the gouging between the cutter and the part surface when the cutter is posed a posture ($\theta = 40°, \lambda = 30°$). This is in line with the A-map where the point (40°, 30°) is beyond the A-map for free gouging (see Figure 7.2d). By changing λ to 38°, no gouging exists as shown in Figure 7.2f since (40°, 38°) is within the gouging-free range.



(a) A-map for global-collision avoidance



(b) A cutter posture with collision ($\lambda = 45°$)

(c) A cutter posture with free collision($\lambda = 38°$)

Figure 7.3 Cutter accessible range for global-collision avoidance

Similarly, by implementing the algorithms developed in Section 2.3.4, the accessible range of global collision is obtained and depicted in Figure 7.3a. Every point within the map represents a cutter posture with free global-collision. Figure 7.3b shows the collision effect when the cutter is posed at (θ = 40°, λ = 45°) that is beyond the A-map (at θ = 40°, $\lambda \in$ (0°, 44.7623°)). When keeping θ = 40°, the collision can be avoided by changing λ = 38°. The corresponding effect is shown in Figure 7.3c.

Intersecting the four accessible ranges, the A-map is shown in Figure 7.4. At this surface point, each range has contributed to the overall A-map.

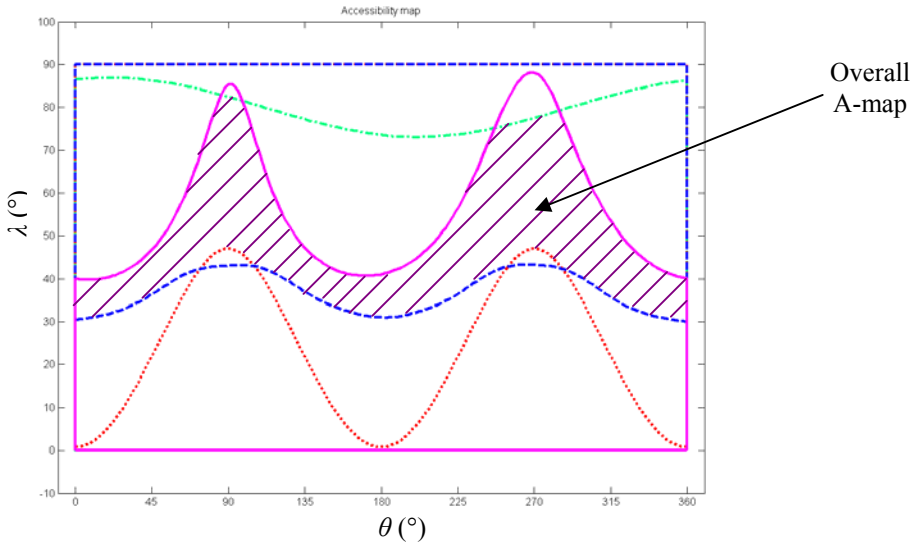
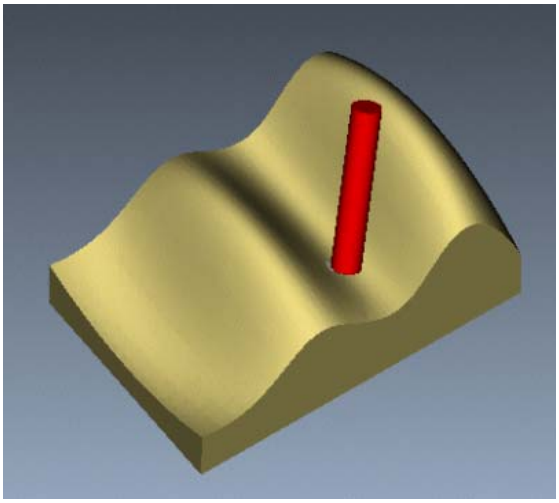


Figure 7.4 A-map at the point

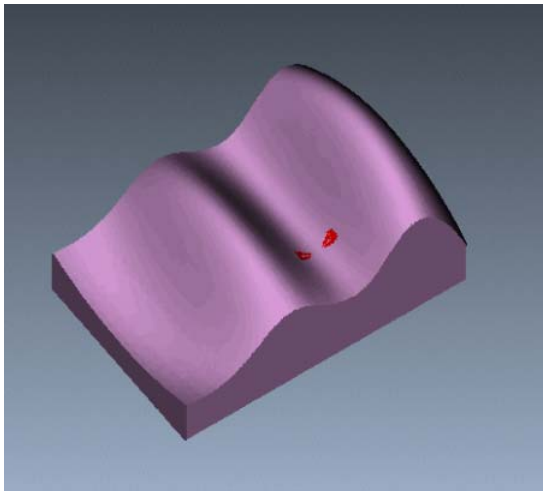
It is noted that in practice, some 5-axis NC machines have a wide range for revolute axis, e.g., Deckel Maho DMU 50V has a revolute range of (0°, 180°) for C-axis and (0°, 360°) for B-axis, respectively. Therefore, for pragmatic sake, the following discussion will be in more details on the relevant issues of gouging and collision and leave issues relevant to machine limits untouched.

7.1.2 Cutting simulation

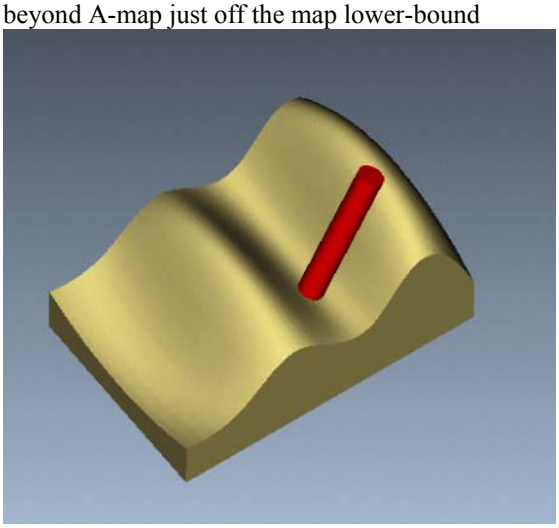
To further verify the effectiveness of our algorithm for A-map, simulation experiments were conducted using the VERICUT® environment. When the cutter is positioned along a posture at each surface point, gouging and collision can be detected using the module AUTO-DIFF.



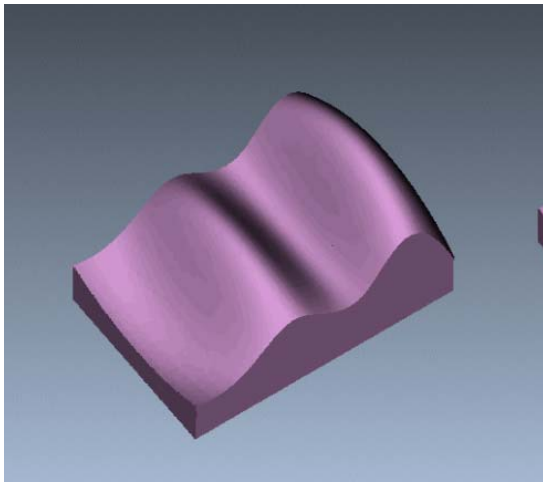
(a) Cutting simulation with the cutter posture beyond A-map just off the map lower-bound



(b) Cutting result after (a)



(c) Cutting simulation with the cutter postures within A-map just off the map lower-bound



(d) Cutting result after (c)

Figure 7.5 Cutting simulations at a point with cutter posture beyond and inside A-map

Assuming that the cutting is only conducted at the point in Section 7.1.1, with the cutter posture along each θ from 181 sampled angle values within posture angle range [0°, 360°]. Figure 7.5a shows the cutting with the cutter postures beyond the A-

map and just off the A-map lower-bound, i.e., $\lambda'_{\theta j} = \lambda_{\theta j}$ - 0.02 (1.1459°). It can be seen that the gouging occurs in the region neighboring the surface point, as shown in Figure 7.5b. To avoid the gouging problem, the cutter is inclined forward by increasing λ while maintaining θ value. In the simulation shown in Figure 7.5c, the cutter postures are set to be within A-map and just off the lower-bound, i.e., $\lambda'_{\theta j} = \lambda_{\theta j}$ + 0.02 (1.1459°). The corresponding effect is shown in Figure 7.5d, which illustrates that gouging is avoided.

7.2 Accessibility of a Single Cutter to a Surface

Intuitively, the cutter selection task can be achieved by applying the CA algorithm for A-map at a point to the whole surface, i.e., all the sampled points of the surface, for all the available cutters until the optimal one is found. However, the computation load would be very heavy because of the numerical nature of CA algorithm. One way to reduce the computation load is to minimize the number of checked points on the surface for a specific cutter. In this section, the result of the algorithm presented in Chapter 3 is shown to reduce the checked point number by dividing the part surface into interference-prone and interference-free regions.

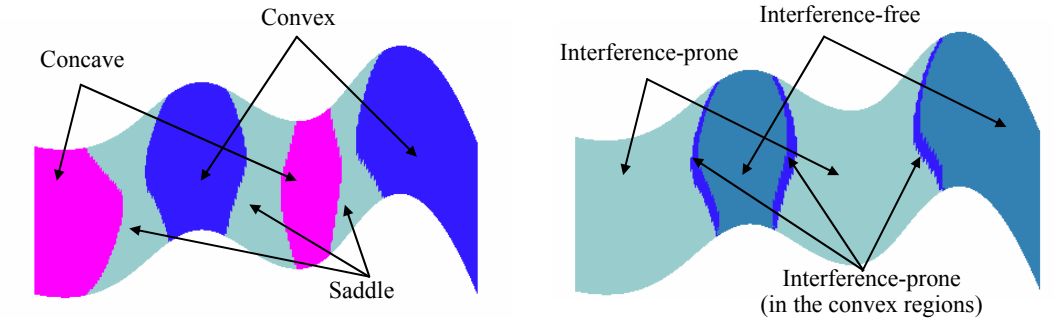
A set of fillet-end cutters is listed in Table 7.2 for optimal cutter selection in terms of maximization of machining efficiency. Cutter major radius (R) ranges from 12mm to 6mm and minor radius (r_f) from 0.5mm to 3mm.

Table 7.2: The cutter library for sculptured surface finishing

Cutters	Major radius R (mm)	Minor radius r_f (mm)						Length L (mm)
T_1 to T_6	12	0.5	1	1.5	2	2.5	3	110
T_7 to T_{12}	10	0.5	1	1.5	2	2.5	3	100
T_{13} to T_{18}	8	0.5	1	1.5	2	2.5	3	90
T_{19} to T_{22}	6	0.5	1	1.5	2			80

For the surface shown in Figure 7.1, 201×201 points were obtained by sampling the surface uniformly along u and v, respectively. First, we applied the CA algorithm to all the sampled points with a specific cutter (R=6.0mm, r_f=0.5mm, L=80mm). It took 11.68 minutes of CPU time on a 2.8MHz PC to finish the search.

Next, by implementing the algorithm in Section 3.2, the surface subdivision algorithm was firstly applied to the sampled points with the same cutter, followed by applying the point-base CA algorithm to the interference-prone points only. The results are shown in Figure 7.6. As illustrated in Figure 7.6a, 43.21% of the points were determined as convex. By using the same cutter, 87.86% of the convex points were further classified as interference-free points, as shown in Figure 7.6b. It can be seen that the concave and saddle points are treated as interference-prone and parts of the two convex regions (near the boundary) are identified as interference-prone regions. In summary, about 37.8877% of the sampled points were identified as interference-free which were excluded from further checking in cutter selection. In terms of computation time, it took about 0.25 minutes to finish surface subdivision and another 7.53 minutes to finish the checking at the interference-prone points. Thus, about 35.55% of computation time was saved.



(a) Surface subdivision based on local surface shape

(b) Surface decomposition results for a fillet-end cutter (R=6.0 mm, r_f=0.5 mm, L=80 mm)

Figure 7.6 The surface decomposed into interference-prone and interference-free regions

The interference-free regions vary for cutters with different dimension. Table 7.3 shows the rate of interference-free regions against the overall surface. The maximum rate value (38.3951%) appears for the cutter with minimum dimension (R, r_f) = (6mm, 2.0mm), while the minimum (33.7393%) for the cutter with maximum dimension (12mm, 0.5mm). This is in line with the algorithm in Section 3.2, in which the algorithm for identification of interference-free regions was proposed based on a dummy flat-end cutter with radius as $2R$-r_f.

Table 7.3: Rate of interference-free regions against the whole surface for cutters

r_f(mm) / R (mm)	0.5	1	1.5	2	2.5	3
12	33.7393%	33.9274%	34.1229%	34.3135%	34.4793%	34.6130%
10	35.1872%	35.3481%	35.4867%	35.6848%	35.8308%	35.9917%
8	36.5164%	36.7095%	36.8951%	37.0362%	37.2095%	37.4149%
6	37.8877%	38.0733%	38.2392%	38.3951%		

7.3 Cutter Accessibility Comparison and Cutter Selection

Surface decomposition can alleviate the computation load to decide whether a cutter is accessible to a given surface. However, if the whole algorithm is repeated for each cutter in cutter list at each interference-prone point till the optimal one is found, the time-cost would still be very high. In Section 4.2, algorithms have been proposed to reduce the redundancy when the search procedure is applied from a larger cutter to a smaller one. When the accessibility of a larger cutter is available at a point, the accessibility of a smaller cutter may be obtained without running the time-consuming CA algorithm.

In this section, a comprehensive case study based on an example is presented to show the validity of the quick estimation method and the improved computational

efficiency of the non-redundant method for optimal cutter selection. The part used is same as that shown in Figure 7.1. All the cutters listed in Table 7.2 are available for finishing the given surface patch. To validate the developed accessibility comparison algorithm, in the first part, four cases are shown involving the four scenarios described in section 4.2, in which the posture ranges of a smaller cutter are obtained based on accessibility comparison and the CA algorithm at a particular point $\mathbf{P}_{c1}$ (113.5, 79.9, 15.6). In the second part, we show a complete case study of optimal cutter selection and compare the performance between the non-redundant algorithm and the one purely based on the CA algorithm.

7.3.1 Case study for the four scenarios

In Section 4.2, accessibility comparison between a larger cutter and a smaller cutter for four scenarios has been studied based on their geometric characteristics. Quick estimation algorithms have been developed to calculate the A-map of the smaller cutter at a point (accessible by the larger cutter) from the A-map of the larger cutter. This section will present examples to show the validity of the quick estimation algorithms by comparing the obtained with the corresponding ones obtained using the CA algorithm for the smaller and larger cutters.

Figure 7.7 illustrates the accessible posture ranges PR_L for T^L (12mm, 1mm) and PR_S for T^S (12mm, 2mm) obtained using the CA algorithm. At some θs, the lower-bound of λ in PR_S is greater than that in PR_L, which indicates that T^S is not ensured to be interference-free at any (θ, λ) in PR_L. This observation is consistent with the discussion in section 4.2.2. Figure 7.7 also shows the RG posture range PR'_S of T^S obtained from accessibility comparison with T^L. The lower-bound of λ in PR'_S is

slightly greater than that in PR_S, which shows that PR'_S is a reasonable conservative estimate of PR_S.

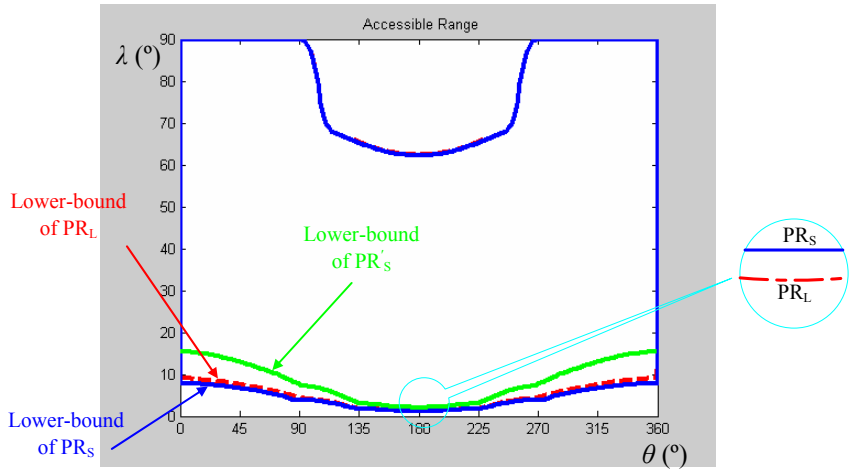


Figure 7.7 Comparison of posture ranges (both RG and GC) for T^S and T^L (scenario 1: $R^S = R^L$ and $r_f^S > r_f^L$)

Figure 7.8 shows PR_L for T^L (12mm, 1mm) and PR_S for T^S (8mm, 1mm) obtained using the CA algorithm. It can be seen that PR_L is entirely inside PR_S. Therefore, by taking PR_L as PR_S, the developed quick estimation algorithm gives a correct answer.

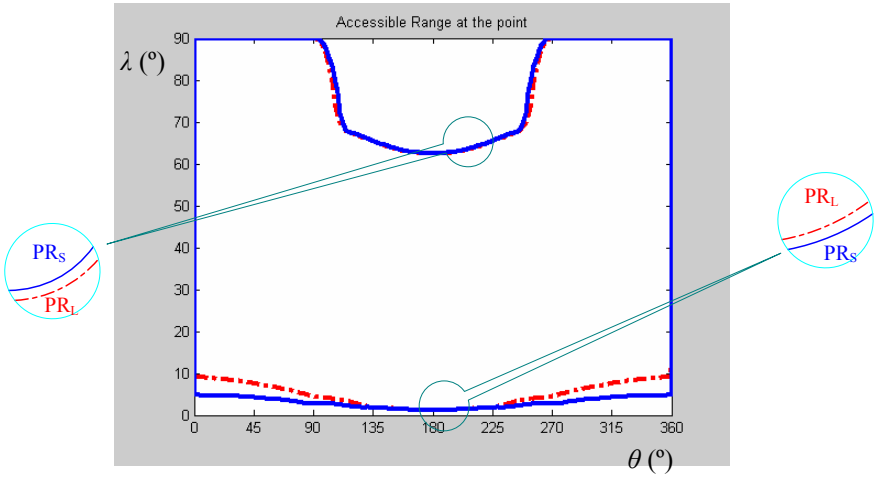


Figure 7.8 Comparison of posture ranges (both RG and GC) for T^S and T^L (scenario 2: $R^S < R^L$ and $r_f^S = r_f^L$)

Figure 7.9 shows PR_L for T^L (12mm, 1mm) and PR_S for T^S (8mm, 2mm) obtained using the CA algorithm. At some θs, the lower-bound of λ in PR_S is greater than that in PR_L. This is evident that T^S is not ensured to be interference-free at any (θ, λ) in PR_L, which is consistent with the discussion in Section 4.2.4. Figure 7.9 also shows PR'_S obtained based on accessibility comparison with T^L, which is a reasonable conservative estimate of PR_S.

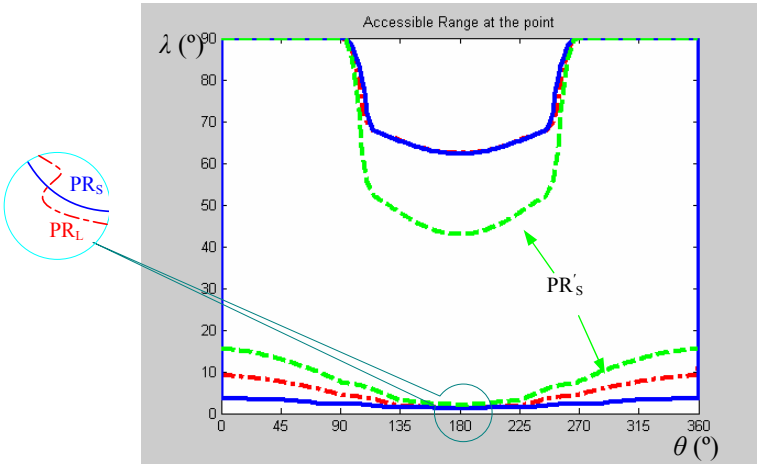
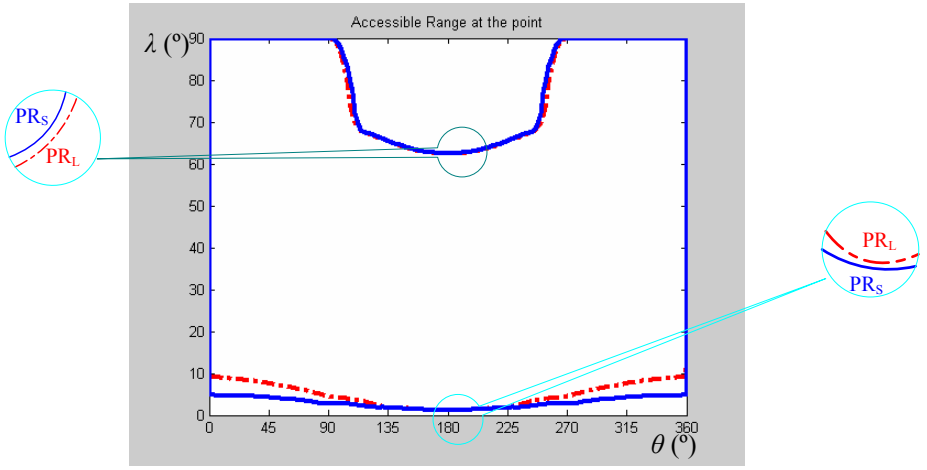


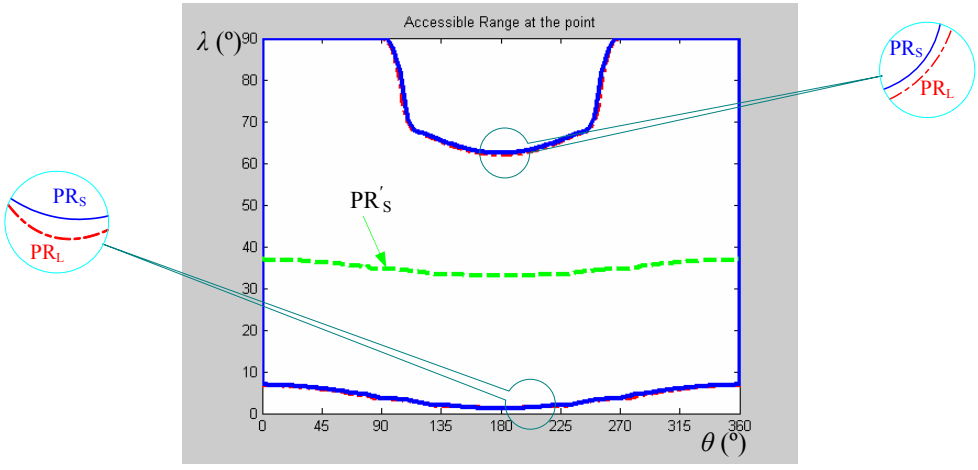
Figure 7.9 Comparison of posture ranges (both RG and GC) for T^S and T^L (scenario 3: $R^S < R^L$ and $r_f^S > r_f^L$)

Figure 7.10a shows PR_L for T^L (12mm, 1mm) and PR_S for T^S (8mm, 0.5mm) obtained using the CA algorithm. This is a case satisfying $R^S < R^L - (r_f^L - r_f^S)$, discussed in the first half of Section 4.2.5. Our quick estimation algorithm suggests that T^S is interference-free within PR_L. This is proved from the results shown in Figure 7.10a in which PR_L is inside PR_S. On the other hand, Figure 7.10b shows PR_L for T^L (12mm, 3mm) and PR_S for T^S (10mm, 0.5mm) obtained using the CA algorithm. This is a case satisfying $R^S > R^L - (r_f^L - r_f^S)$, discussed in the second half of Section 4.2.5. It can be seen that the upper-bound of PR_S is no less than that of PR_L, which indicates that the front shaft surface of T^S is free of global-collision at any (θ, λ)

in PR_L as proved in Section 4.2.5. However, at some θs, the lower-bound of PR_S is greater than that of PR_L, which indicates that the back filleted portion and/or the back shaft surface of T^S may collide with the part surface at these (θ, λ) in PR_L. Using the quick estimation algorithm developed in Section 4.2.5, the lower-bound of PR'_S is obtained as shown in Figure 7.10b. The fact that the lower-bound of PR'_S is greater than that of PR_S indicates that PR'_S is a correct estimate, although with much conservation.



(a) $R^S < R^L - (r_f^L - r_f^S)$



(b) $R^S > R^L - (r_f^L - r_f^S)$

Figure 7.10 Comparison of posture ranges (both RG and GC) for T^S and T^L (scenario 4: $R^S < R^L$ and $r_f^S < r_f^L$)

It is worth mentioning that there exists some difference between range PR'_S obtained from accessibility comparison and the actual range PR_S from the CA algorithm (Figures 7.7 ~ 7.10), though range PR'_S is a reasonable conservative estimate of range PR_S. This mismatch is caused by the tougher criteria used in the process of obtaining PR'_S. In this way, an accessible cutter T^S might be treated as inaccessible at some points. However, it will not have any negative effect on the optimal cutter selection since we are only interested in quickly identifying whether T^S has a posture that causes no interference at this stage. In case when no posture range from quick estimation algorithm is found, the CA algorithm will be applied to T^S for accessibility checking.

7.3.2 Case study for optimal cutter selection

As discussed in Section 7.2, among 201×201 grid points on the surface shown in Figure 7.1, 33.73% was identified as being interference-free and excluded from further analysis when the largest cutter T_1 (12mm, 0.5mm) was considered. The point-based CA algorithm was conducted for T_1 at every interference-prone point. As a result, 18644 points was identified as accessible and 8178 points non-accessible. The next cutter T_2 (12mm, 1mm) in the list was then taken for checking. Using the developed quick estimation algorithm, the conservative posture range was obtained at 17811 points (95.53% out of those 18644) by accessibility comparison with T_1, resulting in significant savings in terms of computation. The re-use rates for the smaller cutters are listed in Table 7.4. The minimum re-use rate is more than 86%. For cutter (10mm, 0.5mm), the re-use rate is 100% since it has the same r_f as T_1. This clearly demonstrates the significant saving achieved by the quick estimation algorithm.

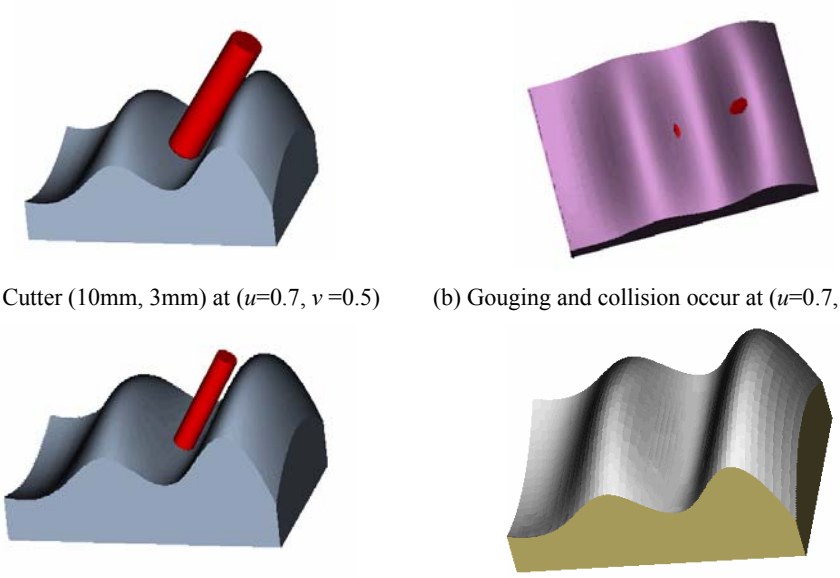
Optimal cutter selection was then conducted by using the non-redundant algorithm proposed in Section 4.3 and the algorithm based on the CA algorithm only independently. In both cases, the cutter with (6mm, 0.5mm) is found to be the largest cutter that could traverse the entire surface without interference. On a PC with a processor of 3G Hz, the total search time was 12.7 minutes for the non-redundant algorithm and 39.2 minutes for the one based on the CA algorithm only. A saving of 67.6% has been achieved.

Table 7.4: Re-use rate of the accessibility range of T_1 for smaller cutters

Cutter	Number of accessible points by (12mm, 0.5mm)	Number of points successfully checked through quick estimation	Re-use rate
(12mm, 1.0mm)	18644	17811	95.53%
(12mm, 1.5mm)	18644	17734	95.12%
(12mm, 2.0mm)	18644	17702	94.95%
(12mm, 2.5mm)	18644	17682	94.85%
(12mm, 3.0mm)	18644	17673	94.79%
(10mm, 0.5mm)	18644	18644	100%
(10mm, 1.0mm)	18644	16337	87.63%
(10mm, 1.5mm)	18644	16167	86.71%
(10mm, 2.0mm)	18644	16098	86.34%
(10mm, 2.5mm)	18644	16066	86.17%
(10mm, 3.0mm)	18644	16035	86.01%

To further verify the effectiveness of our algorithm for cutter selection, simulation experiments were conducted in the VERICUT® environment. At each sampled point, when the cutter is positioned along a posture gouging and collision can be detected using the module AUTO-DIFF. As an example, Figure 7.11a illustrates cutter (10mm, 3mm) at a point (u=0.7, v =0.5). It can be seen that interference occurs both from cutter bottom portion and shaft portion, as shown in Figure 7.11b. In fact, interference cannot be avoided when the cutter is positioned

with any posture at this point. Figure 7.11c illustrates a feasible posture of the optimal cutter (6mm, 0.5mm) at the same point. Both gouging and collision are avoided. In fact, the simulation showed that this cutter is able to access any point on the surface without interference. In Figure 7.11d, the machined surface is shown after the cutter passes through a set of iso-planar tool-paths to finish the given surface, which will be discussed in detail in Section 7.5.



(a) Cutter (10mm, 3mm) at (u=0.7, v =0.5)

(b) Gouging and collision occur at (u=0.7, v =0.5)

(c) Cutter (6mm, 0.5mm) at (u=0.7, v =0.5)

(d) Machined surface with iso-planar tool-paths

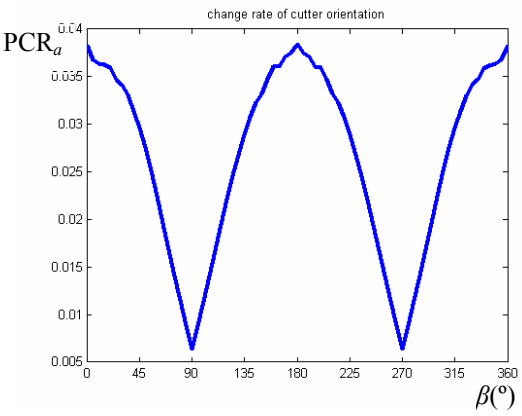
Figure 7.11 Machining simulation using VERICUT®

7.4 Determination of Path Direction

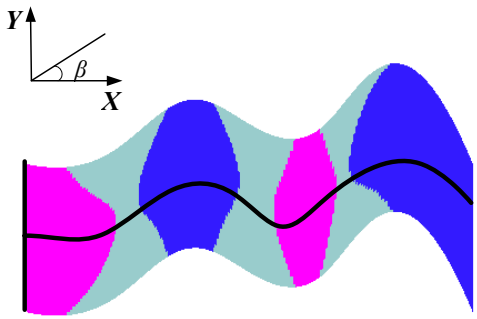
In this section, an example is shown to validate the algorithm proposed in Chapter 5 for determination of the optimal path direction in tool-path generation task by utilizing the A-maps at the sampled points.

To confirm the validity of algorithms proposed, the surface shown in Figure 7.1 is used to evaluate cutter posture variation in a number of cutting directions. The range of [0º, 360º] has been uniformly sampled into 181 discrete angles with an increment of 2º. Figure 7.12a shows the $\sum_{i=1}^{n} \mathrm{PCR}_i$ (n = 201 * 201) along the 181

angles. The direction with angle 90° (or 270° because of the symmetry in surface geometry) from ***X***-axis has the minimum $\sum_{i=1}^{n} \mathrm{PCR}_i$ and thus was taken as the optimal cutting direction. On the other hand, the direction with angle 0° from ***X***-axis has the maximum $\sum_{i=1}^{n} \mathrm{PCR}_i$. This result is consistent with the observation. As shown in Figure 7.12b, the geometric property at a point (including local and global) greatly changes when the cutter moves on a path along ***X***-axis, while it almost maintains constant when the cutter moves on a path along ***Y***-axis (90° from ***X***-axis). For successful cutting, the cutter orientation needs to be adjusted accordingly during machining to match the surface geometry and avoid the gouging and collision with the part surface. Therefore, the cutter has to be oriented with more variation when cutting with the paths along ***X***-axis than that along ***Y***-axis.



(a) Average Cutter posture change rate $\mathrm{PCR}_a = \left[\sum_{i=1}^{n} \mathrm{PCR}_i\right]/n$ along different path direction



(b) varied surface geometry along different path direction

Figure 7.12 Cutter posture change rate for machining a complex surface

To further confirm the validity of algorithms proposed, tool-paths along several other discrete cutting directions in [0°, 90°] were also generated. Table 7.4 shows the resultant path information, including the total path length, overall posture

change at all CC points, and the average PCR. It can be seen that cutting along the direction with 90° from ***X***-axis results in shortest path length (3665.28mm) and minimum PCR (0.0054037), while cutting along the direction with 0° from ***X***-axis results in longest path length (8560.55mm) and maximum average PCR (0.0336675). Therefore, the optimal cutting direction is the one with 90° from ***X***-axis, which is consistent to the above result from the developed algorithm. Furthermore, Table 7.5 also shows that the average PCR decreases with the increment of β, a trend that agrees with that in Figure 7.12a.

Table 7.5: Tool-paths along several different cutting directions

cutting direction β (°)	path length (mm)	Overall posture change	average PCR
0	8560.55	288.212	0.0336675
15	7330.61	243.915	0.0332735
30	5912.67	186.614	0.0315616
45	5406.31	152.192	0.0281508
60	5244.6	116.741	0.0222593
75	4494.08	60.2871	0.0134148
90	3665.28	19.8061	0.0054037

It is noted that, in this case, both the minimum PCR and the shortest path length are achieved along the same direction. However, the shortest path length might not be guaranteed for some surfaces, such as a convex cylindrical surface, when machining along the resultant cutting direction (along the cylindrical axis direction). This may be a limit of the proposed algorithm for cutting direction. On the other hand, although path length might be an effective way to compare cutting efficiency in conventional low speed machining, several other cutting parameters, such as the acceleration/deceleration of feed rates are also crucial aspects for the determination of cutting efficiency, especially in high speed machining. Thus, while path length might

be longer along the cutting direction selected with the proposed approach, the efficiency might not be as low as expected since smooth tool dynamic can reduce acceleration/deceleration procedure and lead to high speed and feed rates in machining.

7.5 Tool-path Generation

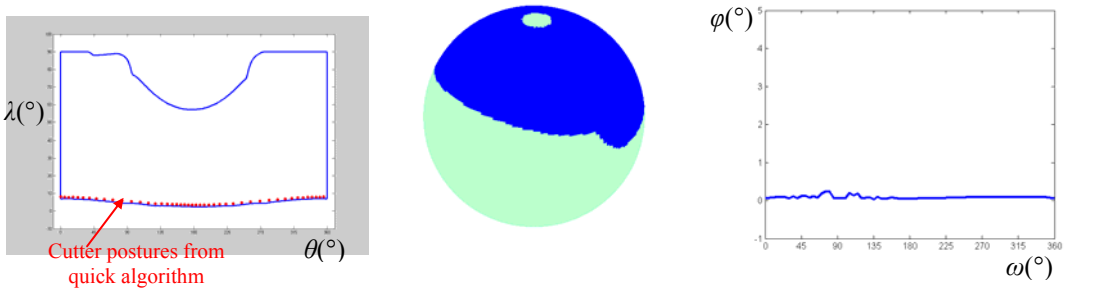
This section shows the effectiveness of the developed algorithm for CL data generation. First, we will show the validity of the quick algorithm for obtaining cutter posture by comparing with the result from CA algorithm. Second, we will show the performance comparison between the quick algorithm and the CA algorithm when applying them in tool-path generation for iso-planar pattern. The cutter used is T(R, r_f, L) = (6, 0.5, 80) mm selected from Section 7.3. The path direction is along ***Y***-axis, the optimal direction resulted from Section 7.4.

7.5.1 Computing accuracy of the quick algorithm for cutter posture

A point $\mathbf{P}_c$ with (u, v) = (0.135727, 0.613037) on the surface was selected to compare the cutter postures from the quick algorithm and the CA algorithm along each feeding direction.

First, the CA algorithm was applied to evaluate the A-map at $\mathbf{P}_c$ in (θ, λ) domain as shown in Figure 7.13a. Figure 7.13b shows the A-map by mapping the range to a unit sphere. The feeding direction range [0º, 360º] has been uniformly sampled as directions with 5º interval. Corresponding to each sampled feeding direction, the cutter posture was found from the A-map by using the algorithm in Section 5.4.1. Second, the quick algorithm introduced in Section 6.2 was applied. Four neighboring points, e.g., (u, v) = (0.135, 0.610), (0.135, 0.615), (0.140, 0.610),

(0.140, 0.615), were selected. Along each feeding direction, the postures at the neighboring points were obtained based on their A-maps. The cutter posture at $\mathbf{P}_c$ was then calculated by interpolating these 4 postures. Comparing the results from both algorithms, Figure 7.13c illustrates the angle φ between two unit cutter axis vectors (from respective cutter postures) along each cutting direction ω. The maximum φ is 0.2545º at ω = 80º, while the minimum is 0.0345º at ω = 135º. It can be seen that the cutter posture obtained using interpolation is very close to the exact one from the CA algorithm. In addition, Figure 7.13a also illustrates the cutter postures along all feeding directions from the quick algorithm in (θ, λ) domain. It can be seen that all orientation angle pairs locate within the range boundary from CA algorithm, which means that no interference occurs when the cutter is posed with these postures. Therefore, the proposed algorithm can effectively find the nearly optimal interference-free cutter posture at $\mathbf{P}_c$.



(a) A-map at $\mathbf{P}_c$ by CA algorithm

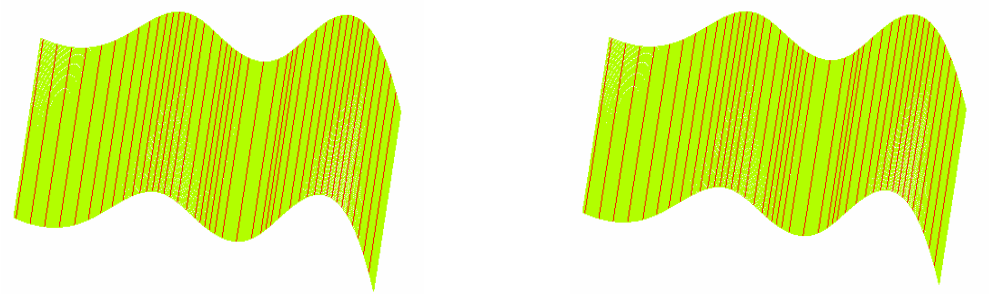
(b) A-map at $\mathbf{P}_c$

(c) angle value between cutter axis directions from quick and CA algorithm

Figure 7.13 Comparison of estimate (quick algorithm) and exact cutter posture (CA algorithm) at $\mathbf{P}_c$

7.5.2 Performance comparison for algorithms of tool-path generation

In this subsection, a performance comparison is conducted between the quick algorithm and the CA algorithm when applied in tool-path generation for iso-planar pattern. Figure 7.14a illustrates the generated iso-planar tool-paths when applying the quick algorithm in tool-path generation. The first path was set on the cutting plane with 1mm from the surface edge. 644 CC points were generated on 44 cutting planes to satisfy the desirable profile tolerance τ = 0.1mm and the scallop height tolerance h = 0.1mm. On the other hand, Figure 7.14b illustrates the generated paths with the similar procedure but using the CA algorithm for cutter posture calculation in tool-path generation. 646 CC point were generated on 44 cutting planes. It can be seen the path distribution with the quick algorithm is very close to that with the CA algorithm. On a PC with a processor of P4-3G Hz, the total computation time was 5.356 seconds for tool-path generation based on the quick approach while 65.828 seconds base on the CA algorithm. A saving of 91.86% has been achieved. Table 7.6 also shows the comparison of resultant tool-paths and the computation time for τ = 0.05mm and h = 0.05mm. It can also be seen that the resultant tool-paths from both algorithm are very close while the computation time from the quick algorithm is much shorter compared that from the CA algorithm.



(a) Tool-paths resulted from the quick approach (b) Tool-paths resulted from the CA algorithm

Figure 7.14 Comparison of iso-planar tool-paths from the quick algorithm and CA algorithm (τ = 0.1mm and h = 0.1mm)

Table 7.6: Performance comparison of the algorithms for tool-path generation

Algorithm	τ (mm)	h (mm)	Number of paths	Number of CC points	Computation time (s)
quick	0.1	0.1	44	646	5.356
CA	0.1	0.1	44	644	65.828
quick	0.05	0.05	62	989	7.925
CA	0.05	0.05	62	986	106.313

To further verify the effectiveness of the algorithm for tool-path generation, simulation experiments were conducted in the VERICUT® environment. Figure 7.15 illustrates the machined surface after the cutter passes through a set of iso-planar paths with τ = 0.1mm and h = 0.1mm to finish the given surface (tool-paths in Figure 7.14a). The figure is same as that in Figure 7.13d, but shown here for better representation. Part of the path G-code (CL data) is given in Appendix B. By using the module AUTO-DIFF, no gouging and scallop height is detected beyond 0.1mm, the desirable tolerance for both profile and scallop height.

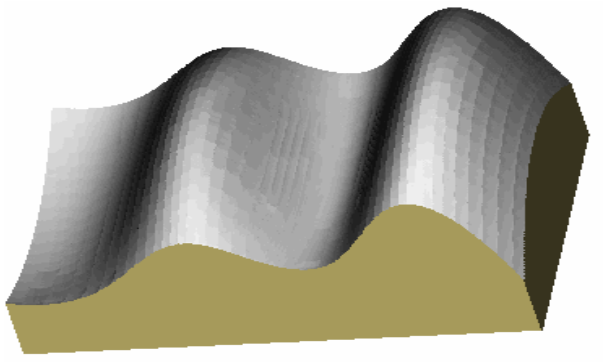


Figure 7.15 The machined surface (simulation) with the generated tool-paths

7.6 Discussion

This research investigates the solutions of the process planning tasks, including cutter selection and tool path generation, for 5-axis finish cut based on the concept of cutter accessibility. Figure 7.16 illustrates the schema of the developed approach for process planning. Given a part surface and a list of available fillet-end cutters, firstly, an intelligent method is applied to efficiently select the optimal cutter from the available ones with respect to cutting efficiency, by checking cutter's accessibility to the sampled points on the given part surface. Besides the optimal cutter, the A-maps at all sampled points are also obtained for the optimal cutter from cutter selection process. Secondly, efficient algorithms are employed for the tasks of tool-path generation, including determining the path direction and generating the CL data. These algorithms are developed based on the A-maps at all sampled points, obtained in cutter selection. In this way, the process planning tasks are solved in an integrated manner. Moreover, this integrated manner contributes to the high efficiency. Firstly, it results in significant reduction of the computation time for process planning by using the checking results from cutter selection into tool-path generation. On the other hand, it also results in high cutting efficiency from the selected optimal cutter, optimal path direction and optimal cutter postures. The experiment results showed that the process planning for 5-axis finish milling can be achieved in an integrated and efficient way by using the methodology introduced in this work.

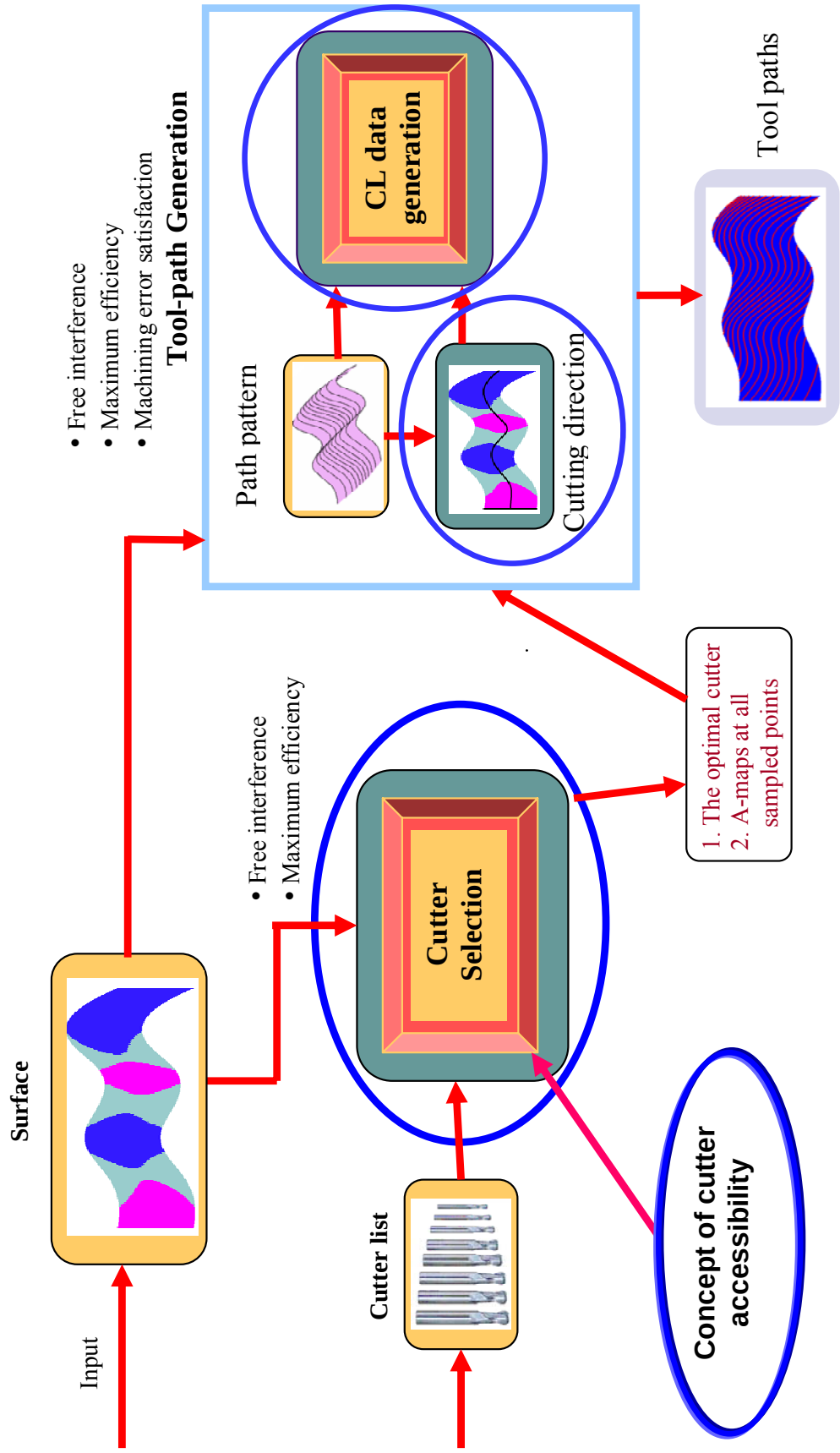


Figure 7.16 Schema of process planning for 5-axis finish cut

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

The main objective of this study was to further investigate techniques for automation of process planning for 5-axis finish sculptured surface cut in an integrated and efficient way. Process planning in 5-axis finish cut includes two main tasks: cutter selection task and tool-path generation task. To be more specific, the work aimed at automating the selection of a fillet-cutter with the maximum efficiency before tool-path generation to finish the whole surface without the interference. Moreover, cutter selection and tool-path generation were not treated as separated tasks. The checking result from the cutter selection was put into use for the processing of tool-path generation, mainly including determination of path direction and generation of CL data. In this chapter, the main work of this research is concluded and the possible directions for future work are presented.

8.1 Conclusions

The main achievements of this study are reflected in the following aspects: (1) evaluation of the A-map at a surface point; (2) accessibility analysis of a specific cutter to a sculptured surface; (3) cutter selection to finish a surface; (4) determination of cutting direction; and (5) tool-path generation.

- The cutter accessibility (CA) algorithm has been proposed to evaluate the A-map at a surface point irrespective to feeding direction. Four geometric attributes have

been considered, including machining axis limits, local-gouging avoidance, rear-gouging avoidance, and global-collision avoidance. The local and global geometry of both cutter and part surfaces are considered for the evaluation of A-map. The computation complexity of CA algorithm is O(km) for a cutter at a point, in which m is the number of sampled points and k is the total number of discrete θs sampled over the rotational angle range (θ_{min}, θ_{max}). One advantage of the algorithms is that the feeding direction was not taken into account. Thus, the resultant A-map can be effectively utilized for cutter selection, which generally should be implemented before the specification of the feeding direction in tool-path generation.

- Surface decomposition technique has been introduced into the accessibility analysis of a cutter to the surface for better computation efficiency. Although the cutter accessibility analysis can be achieved by employing CA algorithm at each sampled surface point, the computation load is much heavy with computation complexity of O(kmn), in which n denotes the number of points to be checked on the surface. One way to reduce the computation load is to minimize n for a specific cutter. The proposed surface decomposition technique can achieve certain success in this direction by decomposing the surface into interference-prone regions and interference-free regions. One example in this study showed that about 37.8877% of the sampled points are found as the interference-free and will be excluded from the time-costing CA algorithm.

- Accessibility comparison between cutters has been investigated to further speed up the processing of cutter selection. Generally, the cutter selection task can be achieved by repeating the search procedure to find the accessibility of all the available cutters till the optimal one. Redundancy has been found when the search procedure is

applied from a larger cutter to a smaller one. The proposed algorithm for accessibility comparison can effectively eliminate this redundancy. At a point where a larger cutter is accessible, the A-map of a smaller cutter is checked based on geometrical characteristics of two cutters, instead of the CA algorithm with computation complexity of O(*km*). The computation complexity of this quick algorithm is O(*k*). The examples in this study demonstrated that the A-map from the simple accessibility comparison algorithm was a reasonable estimation of the real range for the accessibility analysis. Therefore, using this approach, the smaller cutter might be excluded from the CA algorithm if a larger cutter could access this point without interference. One example in this study showed that the computation time could be greatly reduced by introducing this algorithm into the cutter selection task.

In summary, computation-saving is achieved for cutter selection task in this study. The most important reason is to consider the relationship of the geometrical characteristic of the machined surface and the available cutters with the cutter accessibility, instead of consideration at individual points for each individual cutter (Lee and Chang, 1996, Jensen *et al*., 2002). However, although computation load has been greatly alleviated, the algorithms proposed might have consumed more time than that claimed in Jensen *et al*.'s approach (2002). One possible reason is that a trial-and-error method was used in Jensen *et al*.'s approach (2002) to identify a valid cutter orientation instead of A-map at a point of interest. In the worst case, deadly iterative corrected loop might be produced with the trial-and-error approach for interference detection and correction. A suitable cutter might be regarded as an unsuitable, leading to a cutter with low cutting efficiency. In addition, although it is generally possible to involve shorter time for identification of one cutter orientation, the suitable posture for cutting the surface might not be available. On the contrary, for algorithms

proposed in this study, the accessible information is not only comprehensive for cutter selection, but also can be utilized by subsequent tasks, for example, in making decision for path direction and even more in tool-path generation.

One drawback, however, might be the errors from the sampling strategy in the procedure for cutter selection task. To alleviate the sampling errors, tessellation sampling techniques for surfaces have been well established in much work (Piegl and Richard, 1995; Austin *et al.*, 1997; Piegl and Tiller, 1998). In future, this kind of sampling strategy could be incorporated into the current system to reduce computation time while providing a far superior characterization of the surface.

- A new approach has also been developed in this study to identify the optimal path direction for iso-planar tool-paths. The path direction selected could minimize the change rate of cutter orientation along tool-paths while keeping high machining efficiency by maximizing the machining strip width. In this way, high finished surface quality could be achieved, since minimized change of cutter orientation leads to less cutter deflection and vibration. The significant contribution to optimization of path direction in this study is to consider detailed local geometrical characteristics of the part surface except the boundary curve. Thus, the algorithm proposed could be utilized in process planning for 5-axis sculptured surface machining. In addition, computation efficiency is also achieved by using the checking result, A-maps at all sampled points, from the previous cutter selection phase.

It is worth noting that optimal path direction selected might be different from results from the approach in Park and Choi (2000). One possible reason for this difference might be the varied perspectives in study. In Park and Choi (2000), the focus is on the minimum number of tool retractions by taking only the boundary information of the given surface account but not the local geometrical property of

inside surface. It is limited to 3-axis machining. On the contrary, the focus of this study is on the variation of cutter posture in the influence of geometrical characteristics of the whole surface. However, these two approaches are not contradicting. The choice of method used depends on the specified requirement from the user. Furthermore, the algorithms in Park and Choi (2000) could be incorporated into the work in this study to find a path direction not only with small change rate in cutter orientation but also with minimum number of retraction to ensure high cutting efficiency and finished surface quality. This might be one possible direction for future work.

- A quick approach has been developed to specify the cutter posture at a surface point by utilizing the checking result from cutter selection. In tool-path generation, the determination of cutter posture needs to be conducted at a large number of CC points and the traditional CA algorithm is time-consuming because of its numerical nature. To reduce the computation time, an interpolation algorithm was designed in this study to obtain the nearly optimal interference-free cutter posture. One case study showed that the posture obtained with this algorithm is interference-free and very close to the real one derived from CA algorithm. More than 90% time has been saved by applying this algorithm into tool-path generation in the case study. Moreover, the combination of an optimal cutter with high cutting efficiency and an optimal cutting direction with smooth tool dynamics could ensure the finished surface with high quality and productivity.

In general, rotational freedom in 5-axis machining complicates the procedure for process planning. To the best knowledge of the author, this study is the first to propose the idea for the determination of the optimal cutter before tool-path generation in 5-axis finish cut except the trial-and-error methods. In addition, the

checking result from cutter selection has also been used for path direction determination and tool-path generation. In this way, the process planning problems can be solved in an integrated and efficient manner. Furthermore, algorithms were also designed to partly relieve the extensive computation load in complicated process planning for 5-axis finish cut, such as surface decomposition technique for accessibility analysis of a cutter, accessibility comparison between cutters to reduce the computation redundancy. Based on these points, the methods developed in this study can reduce the undesirable user interaction in automated process planning and produce suitable machining parameters for efficient and accurate metal cutting.

8.2 Future Work

Several limitations might exist in this study and the future work is recommended as follows:

- A surface with C^2 continuity was assumed for the evaluation of gouging problem. In practice, this assumption might not be valid. A part surface might include several C^2 surfaces connected by C^1 or even C^0 curves. Thus, new algorithms need to be developed with respect to local-gouging avoidance for this kind of surfaces, while algorithms for avoidance of rear-gouging and global-collision are still effective since they have been developed based on sampled points but not local surface geometry.

- In this study, a single cutter is considered from available ones to finish a given free-form surface. In practice, a set of cutters might be more efficient for some kinds of part surfaces. Thus, one possible future direction could be the investigation of a set of cutters to finish a free-form surface by using the conclusions and results for a single cutter in this study.

- In the algorithm for collision avoidance, the constraints by the cutter's holder and the cutter length are not considered. However, similar algorithm as that in Section 2.3.3 can be easily developed to incorporate the constraints of the cutter's holder into the collision consideration, if the geometry of the holder is given with a regular shape. This can be covered in the future work for the prototype system, in which the user will specify the geometry of the holder before conducting the process planning tasks.

  On the other hand, only one unique cutter was introduced with one dimension (R, r_f) in this work for cutter selection. In practice, however, there might exist a set of cutters with same (R, r_f) but different length L for flexible accessibility and high cutting efficiency. Thus, the cutter with longer length might be considered instead of the one with smaller dimension, when one cutter is not feasible to finish a surface. This might be one aspect for future work.

- In the algorithm for cutter selection, the optimal cutter is selected by checking the available cutters in descending order beginning from the largest cutter in this work. One challenging direction for future work is to apply some intelligent searching techniques to further shorten the computation time rather than checking one by one in sequential order.

- The decomposition approach is applied in this study for the computational acceleration in cutter selection. One promising idea is whether this approach is able to be performed to subdivide the part surface into regions where different machining strategies are employed to improve the cutting efficiency.

- Only iso-planar tool-paths were taken into account in the model system. For some kind of surfaces, another path pattern might be a better choice than iso-planar,

such as contour tool-paths, constant scallop height method. Which one is best for a given sculptured surface by using the checking result from cutter selection might be for future work.

- The optimization of cutting direction in this study focuses on the variation of cutter posture and largest machining strip width in cutting process. In future work, other criteria might be considered to satisfy different user requirements, such as the constant machining strip width, the minimum number of tool retraction, and so on.

REFERENCES

Austin SP, Jerard RB, and Drysdale RL, 1997, Comparison of discretization algorithms for NURBS surfaces with application to numerically controlled machining, *Computer-Aided Design*, Vol. 29, pp. 71-84.

Bala M, and Chang TC, 1991, Automatic cutter selection and optimal cutter path generation for prismatic parts, *International Journal of Production Research*, Vol. 29, pp. 2163-2176.

Balasubramaniam M, Laxmiprasad P, Sarma SE, and Shaikn Z, 2000, Generating 5-axis NC roughing paths directly from a tessellated representation, *Computer-Aided Design*, Vol. 32, pp. 261-277.

Balasubramaniam M, Sarma SE, and Marciniak K, 2003, Collision-free finishing toolpaths from visibility data, *Computer-Aided Design*, Vol. 35, pp. 359-374.

Bard JF, and Feo TA, 1989, The cutting path and tool selection problem in computer aided process planning, *Journal of Manufacturing Systems*, Vol. 8, pp. 17-26.

Beaman J, Barlow J, Bourell D, Crawford R, Marcus H, and Mcalea K, 1997, *Solid Feeform Fabrication: A New Direction Manufacturing*, Kluwer Academic Publishers, Dordrecht, Netherlands.

Chiou CJ, 2004, Accurate tool position for five-axis ruled surface machining by swept envelope approach, *Computer-Aided Design*, Vol. 36, pp. 967-974.

Chiou CJ, and Lee YS, 2002a, A machining potential field approach to tool-path generation for multi-axis sculptured surface machining, *Computer-Aided Design*, Vol. 34, pp. 357-371.

Chiou CJ, and Lee YS, 2002b, Swept surface determination for five-axis numerical control machining, *International Journal of Machine Tools and Manufacture*, Vol. 42, pp. 1497-1507.

Choi BK, Lee CS, Hwang JS, and Jun CS, 1988, Compound surface modeling and machining, *Computer-Aided Design*, Vol. 20, pp. 127-136.

Choi BK, Park JW, and Jun CS, 1993, Cutter-location data optimization in 5-axis surface machining, *Computer-Aided Design*, Vol. 25, 377-386.

Choi BK, Kim DH, and Jerard RB, 1997, C-space approach to tool-path generation for die and mould machining, *Computer-Aided Design*, Vol. 29, pp. 657-669.

Choi BK, and Jerard RB, 1998, *Sculptured Surface Machining: Theory and Application*, Kluwer Academic Publishers, Dordrecht/Boston/London.

DelCAM, http://www.delcam.com.

Ding S, Mannan MA, Poo AN, Yang DCH, and Han Z, 2003, Adaptive iso-planar tool-path generation for machining, *Computer-Aided Design*, Vol. 35, pp. 145-153.

Ding XM, Fuh JYH, and Lee KS, 2001, Interference detection for 3-axis mold machining, *Computer-Aided Design*, Vol. 33, pp. 561-569.

Dragomatz D, and Mann S, 1997, A classified bibliography of literature on NC milling path generation, *Computer-Aided Design*, Vol. 29, pp. 239-247.

DoCarmo MP, 1976, *Differential Geometry of Curves and Surfaces*, Prentice-Hall Inc, New Jersey.

Elber G, and Cohen E, 1994, Tool-path generation for freeform surface models, *Computer-Aided Design*, Vol. 26, pp. 490-496.

Elber G, 1994, Accessibility in 5-axis milling environment, *Computer-Aided Design*, Vol. 26, pp. 796-802.

Elber G, 1995, Freeform surface region optimization for 3-axis and 5-axis milling, *Computer-Aided Design*, Vol. 27, pp. 465-470.

Farin G, 1996, *Curves and Surfaces for Computer-Aided Geometric Design: Practical Guide*, 4th edition, Academic Press.

Faux ID, and Pratt MJ, 1979, *Computational geometry for design and manufacture*, Chichester, Eng., Horwood, New York, Halsted Press.

Gray P, Bedi S, Ismail F, Rao N, and Morphy G, 2001, Comparison of 5-axis and 3-axis finish machining of hydroforming die inserts, *International journal of Advanced Manufacturing Technology*, Vol. 17, pp. 562-569.

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*, Vol. 7, pp. 296-308.

Held M, 1994, Pocket machining based on contour-parallel tool-paths generated by means of proximity maps, *Computer-Aided Design*, Vol. 26, pp.189-203.

Hwang JS, 1992, Interference-free tool-path generation in the NC machining of parametric compound surfaces, *Computer-Aided Design*, Vol. 14, pp. 667-676.

Jamil ATM, 1994, A computerized algorithm for milling non-convex pockets with numerically controlled machines, *International Journal of Production Research*, Vol. 35, pp. 1843-1855.

Jensen CG, and Anderson DC, 1992, Accurate tool placement and orientation for finish surface machining, *Concurrent Engineering*, *ASME*, PED-Vol.59, pp. 127-145.

Jensen CG, and Anderson DC, 1996, A review of numerically controlled methods for finish sculptured surface machining, *IIE transactions*, Vol. 28, pp. 30-39.

Jensen CG, Red WE, and Pi J, 2002, Tool selection for five-axis curvature matched machining, C*omputer-Aided Design*, Vol. 34, pp. 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*, Vol. 35, pp.549-566.

Kim BH, and Choi BK, 2000, Guide surface based tool-path generation in 3-asis milling: an extension of the guide plane method, *Computer-Aided Design*, Vol. 32, pp.191-199.

Kim BH, and Choi BK, 2002, Machining efficiency comparison direction-parallel tool-path with contour-parallel tool-path, *Computer-Aided Design*, Vol. 34, pp. 89-95.

Lakkaraju R, and Raman S, 1990, Optimal NC path planning: is it really possible? *Computers and Industrial Engineering*, Vol. 19, pp. 462-464.

Lee K, Kim TJ, and Hong SE, 1994, Generation of toolpath with selection of proper tools for rough cutting process, *Computer-Aided Design*, Vol. 26, pp. 822-831.

Lee YS, Choi BK, and Chang TC, 1992, Cut distribution and cutter selection for sculptured surface cavity machining, *International Journal of Production Research*, Vol. 30, pp. 1447-1470.

Lee YS, and Chang TC, 1995, 2-phase approach to global tool interference avoidance in 5-axis machining, *Computer-Aided Design,* Vol. 27, pp. 715-729.

Lee YS, and Chang TC, 1996, Automatic cutter selection for 5-axis sculptured surface *machining, International Journal of Production Research*, Vol. 34, pp. 977-998.

Lee YS, 1997, Admissible tool orientation control of gouging avoidance for 5-axis complex surface machining, *Computer-Aided Design*, Vol. 29, pp. 507-521.

Lee YS, 1998, Non-isoparametric tool-path planning by machining strip evaluation for 5-axis sculptured surface machining, *Computer-Aided Design*, Vol. 30, pp. 559-570.

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*, Vol. 42, pp. 2403-2417.

Li SX, and Jerard B, 1994, 5-axis machining of sculptured surfaces with a flat-end cutter, *Computer-Aided Design*, Vol. 26, pp. 65-178.

Manson F, 1995, Die and mold finishing, how fast? *Manufacturing Engineering*, September.

Marshall S, and Griffiths JG, 1993, A new cutter-path topology for milling machines, *Computer-Aided Design*, Vol. 26, pp. 204-211.

Monreal M, and Rodriguez CA, 2004, Influence of tool-path strategy on the cycle time of high-speed milling, *Computer-Aided Design*, Vol. 35, pp.395-401.

Morishige K, Takeuchi Y, and Kase K, 1999, Tool-path generation using C-Space for 5-axis control machining, *ASME Journal of Manufacturing Science and Engineering*, Vol. 121, pp. 144-149.

Mullins SH, Jensen CG, and Anderson DC, 1993, Scallop elimination based on precise 5-axis tool placement, orientation, and step-over calculations, *American Society of Mechanical Engineers, Design Engineering Division (Publication) DE*, Vol. 65, Part 2, *Advances in Design Automation*, pp. 535-544.

O'Neil B, 1966, *Elementary Differential Geometry*, Academic Press, New York.

Park SC, 2003, Tool-path generation for Z-constant contour machining, *Computer-Aided Design*, Vol. 35, pp. 27-36.

Park SC, and Choi BK, 2000, Tool-path planning for direction-parallel area milling, *Computer-Aided Design*, Vol. 32, pp. 17-25.

Pi J, Red WE, and Jensen CG, 1998, Grind-free tool-path generation for five-axis surface machining, *Computer Integrated Manufacturing Systems*, Vol. 11, pp. 227-350.

Piegl LA, and Richard AM, 1995, Tessellating trimmed NURBS surface, *Computer-Aided Design*, Vol. 27, pp. 6-26.

Piegl LA, and Tiller W, 1998, Geometry-based triangulation of trimmed NURBS surface, *Computer-Aided Design*, Vol. 30, pp. 11-18.

Prabhu PV, Gramopadhye AK, and Wang H, 1990, General mathematical mode of optimizing NC tool-path for face milling of flat convex polygonal surfaces, *International Journal of Production Research*, Vol. 28, pp. 101- 130.

Press WH, Teukolsky SA, Vetterling WT, and Flannery BP, 2002, *Numerical Recipes in C*, Second Edition, http://www.nrbook.com/a/bookcpdf.php.

Rao A, and Sarma R, 2000, On local gouging in five-axis sculptured surface machining using flat-end tools, *Computer-Aided Design*, Vol. 32, pp. 409-420.

Sarma SE, 1999, The crossing function and its application to zig-zag tool-paths, *Computer-Aided Design*, Vol. 31, pp. 881-890.

Schulz H, 1996, High-speed machining in die and mold manufacturing, *Advanced Manufacturing System and Technology*, Springer, New York.

Sheltami K, Bedi S, and Ismail F, 1998, Swept volumes of toroidal cutters using generating curves, *International Journal of Machine Tools and Manufacture*, Vol. 38, pp.855-870.

Smith TS, and Farouki RT, 2001, Gauss map computation for free-form surfaces, *Computer Aided Geometric Design*, Vol. 18, pp. 831-850.

Suh, SH, and Kang JK, 1995, Process planning for multi-axis NC machining of free surfaces, *Internal Journal of Production Research*, Vol. 33, pp. 2723-2738.

Unigraphics, http://www.ugs.com/products/nx/.

VERICUT, http://www.cgtech.com/.

Vickers GW, and Quan KW, 1989, Ball-mills versus end-mills for curved surface machining, *Journal of Engineering for industry*, Vol. 111, pp. 22-26.

Wang HP, Chang H, Wysk RA, and Chandrawarkar A, 1987, On the efficiency of NC tool-path planning for face milling operations, *ASME, Journal of Engineering for Industry*, Vol. 109, pp. 370-376.

Wang HP, Chang H, and Wysk RA, 1988, An analytical approach to optimization NC tool-paths for face milling, *IIE Transactions (Institute of Industrial Engineers)*, Vol. 20, pp. 325-332.

Warkentin A, Bedi S, and Ismail F, 1996, Five-axis milling of spherical surfaces, *International Journal of Machine Tools and Manufacture*, Vol. 36, pp. 229-243.

Woo TC, Visibility maps and spherical algorithms, 1994, *Computer-Aided Design*, Vol. 26, pp. 6-16.

Xu XJ, Bradley C, Zhang YF, Loh HT, and Wong YS, 2002, Tool-path generation for five-axis machining of free-form surfaces based on accessibility analysis, *International Journal of Production Research*, Vol. 40, pp. 3253-3274.

Yang DCH, Chuang JJ, and OuLee TH, 2003, Boundary-conformed toolpath generation for trimmed free-form surfaces, *Computer-Aided Design*, Vol. 35, pp. 127-139.

Yoon JH, Pottmann H, and Lee YS, 2003, Locally optimal cutting positions for 5-axis sculptured surface machining, *Computer-Aided Design*, Vol. 35, pp. 69-81.

Zhang W, Zhang YF, and Ge QJ, 2005, Interference-free tool-path generation for 5-axis sculptured surface machining using rational Bezier motions of a flat-end cutter, *International Journal of Production Research*, Vol. 43, pp. 4103-24.

APPENDIX A

SURFACE DATA

3 2
14 2

.0804608234568067	.0299290898961521	.005	1
.0804608234568067	.0799290898961521	-.015	1
.0804608234568067	.129929089896152	.005	1
.0904608234568067	.0299290898961521	.0025	1
.0904608234568066	.0799290898961521	-.0175	1
.0904608234568067	.129929089896152	.0025	1
.1054608234568066	.0299290898961521	.005	1
.1054608234568066	.0799290898961521	-.005	1
.1054608234568066	.129929089896152	.005	1
.1192108234568066	.0299290898961521	.01875	1
.1192108234568066	.0799290898961521	.03375	1
.1192108234568066	.129929089896152	.01875	1
.1304608234568066	.0299290898961521	.03	1
.1304608234568066	.0799290898961521	.06	1
.1304608234568066	.129929089896152	.03	1
.1429608234568066	.0299290898961521	.03125	1
.1429608234568066	.0799290898961521	.05875	1
.1429608234568066	.129929089896152	.03125	1
.1604608234568065	.0299290898961521	.015	1
.1604608234568065	.0799290898961521	.015	1
.1604608234568065	.129929089896152	.015	1
.1768670734568065	.0299290898961521	.01265625	1
.1768670734568065	.0799290898961521	.00171875	1
.1768670734568065	.129929089896152	.01265625	1
.1896795734568065	.0299290898961521	.03296875	1
.1896795734568065	.0799290898961521	.04015625	1
.1896795734568065	.129929089896152	.03296875	1

.1985662922068065 .0299290898961521 .04380859375 1
.1985662922068065 .0799290898961521 .06642578125 1
.1985662922068065 .129929089896152 .04380859375 1

.2068084797068065 .0299290898961521 .04470703125 1
.2068084797068065 .0799290898961521 .07787109375 1
.2068084797068065 .129929089896152 .04470703125 1

.2137078937693065 .0299290898961521 .0398291015625 1
.2137078937693065 .0799290898961521 .0789404296875 1
.2137078937693065 .129929089896152 .0398291015625 1

.2214764484568065 .0299290898961521 .027734375 1
.2214764484568065 .0799290898961521 .069765625 1
.2214764484568065 .129929089896152 .027734375 1

.2273358234568065 .0299290898961521 .01375 1
.2273358234568065 .0799290898961521 .055 1
.2273358234568065 .129929089896152 .01375 1

.2304608234568065 .0299290898961521 .005 1
.2304608234568065 .0799290898961521 .045 1
.2304608234568065 .129929089896152 .005 1

0 0 0 0 .125 .25 .375 .5 .625 .75 .8125 .875 .90625 .9375 .96875 1 1 1 1
0 0 0 1 1 1

Notes:

The format of the data is as followings:

1. The 1st line presents the 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) ($i = 0,\ldots n_i$; $j = 0,\ldots, n_j$) are presented.
4. The 2nd line from the bottom presents the knots along u direction.
5. The 1st line from the bottom presents the knots along v direction.

APPENDIX B

PART OF PATH G-CODE IN VERICUT®

```
SPINDL/4000.0000
COOLNT/ON
FEDRAT/750.0000
RAPID
CUTTER/12, 0.5, 5.5, 0.5, 0, 0, 80
FROM/-50.0000, 50.0000, 150.0000   0      0      1
RAPID

RAPID
GOTO/81.5451   142.909          150              0              0              1
RAPID
GOTO/81.5451   142.909          7.17821          0.207298       -0.430282      0.87857

SPINDLE/ON
GOTO/81.5451   134.909          7.17821          0.207298       -0.430282      0.87857
GOTO/81.568    129.27           4.93327          0.206757       -0.413204      0.886856
GOTO/81.5892   123.55           2.90276          0.206224       -0.392977      0.896125
GOTO/81.6083   117.754          1.0985           0.205715       -0.369301      0.906255
GOTO/81.6245   111.887          -0.468042        0.205262       -0.341902      0.917045
GOTO/81.6389   105.957          -1.78651         0.204844       -0.31055       0.928223
GOTO/81.6477   99.9727          -2.84647         0.204579       -0.27509       0.9394
GOTO/81.6813   93.9439          -3.64601         0.203546       -0.23548       0.950326
GOTO/81.6374   87.8796          -4.15666         0.20491        -0.191758      0.959813
GOTO/81.693    81.7918          -4.40466         0.20316        -0.145251      0.968313
GOTO/81.6443   75.6932          -4.36206         0.204684       -0.0931201     0.974389
GOTO/81.6454   69.5936          -4.04203         0.204647       -0.0390618     0.978056
GOTO/81.6371   63.5063          -3.43366         0.204898       0.0163566      0.978647
GOTO/81.6213   57.4414          -2.55711         0.205353       0.0753803      0.975781
GOTO/81.6041   51.4117          -1.40224         0.205829       0.134295       0.969329
GOTO/81.5856   45.429           0.0247948        0.206324       0.192162       0.959429
GOTO/81.5623   39.4974          1.69658          0.206895       0.251796       0.945406
GOTO/81.5451   35.2038          3.07964          0.207298       0.293786       0.933123
GOTO/81.5451   27.7752          6.04861          0.207298       0.293786       0.933123

SPINDLE/OFF
RAPID
GOTO/81.5451   35.2038          150              0.207298       0.293786       0.933123
RAPID
GOTO/81.5451   35.2038          150              0              0              1


RAPID
GOTO/87.5067   142.885          150              0              0              1
RAPID
GOTO/87.5067   142.885          6.30095          0.0804796      -0.450143      0.889322

SPINDLE/ON
GOTO/87.5067   134.885          6.30095          0.0804796      -0.450143      0.889322
```

GOTO/87.501	129.303	4.21875	0.0558061	-0.440459	0.896037
GOTO/87.4893	123.661	2.33808	0.0332722	-0.427028	0.903626
GOTO/87.4738	117.96	0.6659	0.0132297	-0.409563	0.912186
GOTO/87.4564	112.203	-0.790167	-0.00400988	-0.387824	0.921725
GOTO/87.4394	106.394	-2.0222	-0.018181	-0.361634	0.932143
GOTO/87.4246	100.54	-3.02237	-0.0290658	-0.330877	0.943226
GOTO/87.4136	94.6472	-3.78347	-0.0364966	-0.295512	0.954642
GOTO/87.4076	88.7241	-4.29938	-0.0403563	-0.255577	0.965946
GOTO/87.4073	82.7803	-4.56556	-0.0405757	-0.211197	0.976601
GOTO/87.4126	76.8257	-4.57929	-0.0371412	-0.162588	0.985995
GOTO/87.4231	70.8705	-4.33989	-0.0300899	-0.110073	0.993468
GOTO/87.4376	64.9246	-3.84853	-0.019514	-0.0540837	0.998346
GOTO/87.4547	58.9967	-3.10804	-0.00556223	0.00483053	0.999973
GOTO/87.4723	53.0949	-2.12245	0.0115597	0.0660018	0.997753
GOTO/87.4897	47.2355	-0.85133	0.0315472	0.120332	0.992232
GOTO/87.5035	41.4098	0.619357	0.0541353	0.181769	0.98185
GOTO/87.5126	35.63	2.32703	0.0789416	0.241593	0.967161
GOTO/87.5127	35.3139	2.42391	0.0803615	0.245426	0.966079
GOTO/87.5127	27.8419	5.28221	0.0803615	0.245426	0.966079
SPINDLE/OFF					
RAPID					
GOTO/87.5127	35.3139	150	0.0803615	0.245426	0.966079
RAPID					
GOTO/87.5127	35.3139	150	0	0	1
RAPID					
GOTO/93.398	142.912	150	0	0	1
RAPID					
GOTO/93.398	142.912	6.26833	-0.0802809	-0.447078	0.890885
SPINDLE/ON					
GOTO/93.398	134.912	6.26833	-0.0802809	-0.447078	0.890885
GOTO/93.3354	129.238	4.42352	-0.133192	-0.439237	0.888443
GOTO/93.265	123.474	2.74851	-0.179938	-0.427909	0.885729
GOTO/93.2029	117.629	1.20783	-0.220461	-0.405309	0.887199
GOTO/93.145	111.68	-0.135324	-0.254192	-0.380492	0.889164
GOTO/93.0974	105.632	-1.27752	-0.281062	-0.351483	0.893008
GOTO/93.0619	99.4925	-2.19914	-0.300929	-0.319052	0.898692
GOTO/93.0391	93.277	-2.88223	-0.313672	-0.283841	0.906115
GOTO/93.0293	87.0058	-3.31258	-0.319188	-0.246294	0.915127
GOTO/93.0325	80.7045	-3.4814	-0.317402	-0.206611	0.92551
GOTO/93.0487	74.4011	-3.38637	-0.308274	-0.164731	0.936926
GOTO/93.0782	68.1234	-3.03208	-0.291801	-0.120354	0.948877
GOTO/93.1206	61.8965	-2.42944	-0.268026	-0.0729988	0.960642
GOTO/93.1687	55.7415	-1.56967	-0.236799	-0.0266961	0.971192
GOTO/93.2283	49.6716	-0.497145	-0.198624	0.0239205	0.979784
GOTO/93.2921	43.6951	0.790559	-0.153628	0.0752262	0.985261
GOTO/93.3581	37.8098	2.26815	-0.102348	0.128869	0.986366
GOTO/93.3851	35.4443	2.90735	-0.0799212	0.153298	0.984943
GOTO/93.3851	27.861	5.45565	-0.0799212	0.153298	0.984943
SPINDLE/OFF					
RAPID					
GOTO/93.3851	35.4443	150	-0.0799212	0.153298	0.984943
RAPID					
GOTO/93.3851	35.4443	150	0	0	1

RAPID
GOTO/99.2032 143.043 150 0 0 1
RAPID
GOTO/99.2032 143.043 7.1336 -0.272152 -0.397005 0.876539

SPINDLE/ON
GOTO/99.2032 135.043 7.1336 -0.272152 -0.397005 0.876539
GOTO/99.0645 129.009 5.66882 -0.349863 -0.395122 0.849396
GOTO/98.9947 122.816 4.16538 -0.416224 -0.361565 0.834283
GOTO/98.9624 116.388 2.77662 -0.470629 -0.320447 0.822084
GOTO/98.9202 109.697 1.61336 -0.512154 -0.289013 0.808808
GOTO/98.9109 102.78 0.626746 -0.543041 -0.251558 0.80114
GOTO/98.911 95.6618 -0.118224 -0.563148 -0.215462 0.797772
GOTO/98.9088 88.3956 -0.584022 -0.57255 -0.183488 0.799074
GOTO/98.909 81.0518 -0.762362 -0.57154 -0.153574 0.806075
GOTO/98.9119 73.7078 -0.646948 -0.559929 -0.125367 0.819001
GOTO/98.9205 66.4416 -0.245229 -0.537287 -0.0980212 0.837684
GOTO/98.9404 59.3234 0.421442 -0.503001 -0.0701553 0.861434
GOTO/98.9785 52.4083 1.31974 -0.456357 -0.0399351 0.8889
GOTO/99.0221 45.7338 2.45059 -0.395999 -0.0136943 0.918149
GOTO/99.0891 39.3117 3.75709 -0.321927 0.0136189 0.946667
GOTO/99.1453 35.5156 4.61577 -0.270181 0.0327322 0.962253
GOTO/99.1453 27.7632 6.59058 -0.270181 0.0327322 0.962253

SPINDLE/OFF
RAPID
GOTO/99.1453 35.5156 150 -0.270181 0.0327322 0.962253
RAPID
GOTO/99.1453 35.5156 150 0 0 1

RAPID
GOTO/105.061 143.296 150 0 0 1
RAPID
GOTO/105.061 143.296 8.87544 -0.470982 -0.264773 0.84147
SPINDLE/ON

GOTO/105.061 135.296 8.87544 -0.470982 -0.264773 0.84147
GOTO/104.883 128.532 8.03469 -0.559637 -0.274189 0.782065
GOTO/104.9 121.335 7.00987 -0.630569 -0.230181 0.741215
GOTO/104.958 113.616 6.08922 -0.682635 -0.185358 0.70686
GOTO/105.009 105.392 5.34328 -0.718065 -0.149452 0.67974
GOTO/105.084 96.7427 4.76249 -0.740177 -0.111847 0.663045
GOTO/105.094 87.7879 4.45019 -0.749048 -0.0924417 0.656035
GOTO/105.091 78.693 4.37565 -0.746168 -0.0775938 0.661221
GOTO/105.074 69.6412 4.54512 -0.7309 -0.0668378 0.679204
GOTO/105.048 60.8155 4.94859 -0.701455 -0.0595259 0.710223
GOTO/105.013 52.3745 5.56988 -0.65471 -0.0568383 0.75374
GOTO/104.991 44.4299 6.36699 -0.58685 -0.0557766 0.807772
GOTO/104.959 37.0368 7.37152 -0.492234 -0.0702727 0.867622
GOTO/104.978 35.4974 7.57412 -0.46845 -0.066965 0.880948
GOTO/104.978 27.5674 8.63046 -0.46845 -0.066965 0.880948

SPINDLE/OFF
RAPID
GOTO/104.978 35.4974 150 -0.46845 -0.066965 0.880948
RAPID
GOTO/104.978 35.4974 150 0 0 1

……