A New Geometric-and-Physics Model of Milling and An Effective Approach to Medial Axis Transforms of Free-form Pockets for High Performance Machining

Qiang Fu

A Thesis

In the Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy at

Concordia University

Montreal Quebec, Canada

May 2010

Your file Votre référence
ISBN: 978-0-494-71161-3
Our file Notre référence
ISBN: 978-0-494-71161-3

ABSTRACT

A New Geometric-and-Physics Model of Milling and An Effective Approach to Medial Axis Transforms of Free-form Pockets for High Performance Machining

Qiang Fu, Ph.D.

Concordia University, 2010

Mechanical part quality and productivity depend on many parameters in CNC milling processes, such as workpiece material, cutters, tool paths, feed rate, and spindle speed, etc. To pursue high performance machining, the cutting parameter optimization is in high demand in industry, though it is quite challenge. This innovative research successfully addresses some essential problems in optimizing the cutting parameters by developing a new geometric-and-physics integrated model of milling and proposing an effective approach to the medial axis transforms of free-form pockets.

In this research, an original geometric model of 2½- and 3-axis CNC milling is developed and integrated with a well-established mechanistic model. A main research contribution is that this integrated model can predict complex milling processes in higher fidelity with instantaneous material remove rates, cutting forces and spindle powers, compared to prior machining models. In the geometric model, an in-process workpiece model is introduced by using a group of discrete Z-layers and applying the B-Rep scheme to represent the workpiece shape on each layer, in order to accurately represent instantaneous cutter-and-workpiece engagement in

2½- and 3-axis milling. Hence, the un-deformed chip geometry can be found even for complex part milling, which is then fed to the mechanistic model to predict instantaneous cutting forces. By using this integrated model, cutting parameters can be optimized for profiling, pocketing, and surface milling to ensure steady cut and the maximum material removal rates. This model has been verified by experiments, and will be implemented into a software tool for Bombardier Aerospace.

Another important research in this work is to propose aggressive roughing of free-form pockets for ultimately high cutting efficiency. For this purpose, an accurate, efficient approach to the medial axis transforms of free-form pockets and an optimal approach to multiple cutters selection and their path generation are proposed. The main contributions of this research include (1) a new mathematical model of medial axis point, (2) an innovative global optimization solver, the hybrid global optimization method, (3) an optimization model of selecting multiple cutters for the maximum material removal rate. This research can substantially promote aggressive roughing in the machining industry to increase cutting efficiency of free-form pockets. The technique has been validated using considerable number of cutting tests and can be directly implemented into commercial CAD/CAM software.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and gratitude to my supervisor Prof. Chevy Chen for his invaluable guidance, support and enthusiasm throughout this work. I have gained valuable research experience and significantly improved my technical skills under his supervision.

Thanks to my fellow graduate students, Xie Shuangxi, Maqsood Ahmed Khan, Liu Gang, Zhang Hongda, Saeed Al-Taher, Huang Guogui, and Zheng Hong, for providing all sorts of help and creating an excellent working atmosphere.

Finally, I sincerely thank my parents and wife for their never-ending love and support.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Research Problems

Computer numerical control (CNC) milling machine tools are widely used in the manufacturing industry. Milling is a process of removing the excess material from the workpiece in the form of small individual chips by a rotating cutting tool moving along predetermined tool paths. These chips are formed by the intermittent engagement of cutting edges or teeth of the milling cutter with the workpiece. The term CNC is defined as a self-contained numerical control (NC) system for a single machine tool that uses a dedicated computer controlled by stored instructions in the memory to implement some or all of the basic NC functions [1]. With the application of the CNC technology, considerable improvements of accuracy, repeatability and productivity are achieved; setup time and lead time are greatly reduced; most importantly, curved profiles and surfaces can be cut by the computer guided cutter, and complex 3D parts are relatively easy to manufacture.

The CNC milling process depends on many parameters, including tool path strategy, stock geometry, cutter geometry, feed rate, spindle speed, and etc. Selecting proper cutting parameters to safely and effectively run the CNC machine tools is a major concern in industry. The common practice in machine shops is to select cutting parameters from machining handbooks, and to verify the tool paths

using geometric simulation software. Since the machining handbooks only provide ranges of recommended values for simple steady cutting conditions, such as slot and shoulder cutting, the selection of cutting parameters for general applications still heavily depends on the user's experience; consequently, conservative values are often used to ensure safety. The geometric simulation software used in industry, which only verify the geometric correctness of tool paths and machined parts, cannot provide sufficient information for the optimization of cutting parameters. To facilitate the selection of optimal cutting parameters, it is essential to develop systems that are capable to predict the physics characteristics of the machining process, including material remove rate, cutting forces, torque, power and vibrations, rather than the simple geometric verification of tool paths.

Cutting forces modeling [2-4] is one of the classic research areas of machining processes. The empirical cutting force models consider that cutting forces vary as a function of cutting conditions, normally including undeformed chip geometry (thickness, width), material properties and etc. For certain cutting tool versus workpiece material, the cutting forces coefficients can be identified by mechanistic modeling methods. The existing mechanistic models are well established for cutting forces prediction of simple machining operation with fixed cutting conditions.

In reality, because of the increasing complexity of the part geometry, manual part programming has become extremely cumbersome and expensive. Commercial Computer Aided Manufacturing (CAM) software is used to generate tool paths for

complex mechanical parts. Since the generated tool paths are composed of many segments, including straight lines, circular arcs and spline curves, transient cuts frequently occur. Even in the simple steady machining operations, such as slot and shoulder milling, transient cuts still occur when the cutter engages with and disengages from the workpiece. For more complicate 3-axis milling of sculptured surface parts, several stages, including roughing, semi-finish, finishing and clean-up, are involved. After each stage, the cusp shape left on the workpiece will influence the cutter and workpiece engagement in the following process. For example, the stairs-like shape generated by the z-level rough machining strategy will significantly affect the engagement in the finish machining. The challenge in the milling process simulation is to predict cutting forces, torque, and power while the cutter and workpiece engagement varies along the tool paths. Since the well established cutting forces models can be used to predict cutting forces at discrete cutting edge elements, the key issue in the accurate process modeling is to find out the instantaneous cutter and workpiece engagement geometry.

Another part of this dissertation stems from our previous research on the cutter sizes selection and tool path generation for 2½-axis pocket milling [5,6]. After the cutter size is determined, the pocketing tool paths can be generated for roughing and finishing. In the stage of roughing, the objective is to get rid of raw materials as fast as possible. In the stage of finishing, the objective is to achieve the satisfied surface quality and the tolerance. The machining efficiency and the tool path pattern rely on the size of the cutter. Generally, a big cutter can reach a high machining efficiency by reducing the number of tool paths through adopting a larger distance

between two adjacent tool paths; and its high rigidity can reduce the tool vibration. But big cutters are prone to gouging and interference.

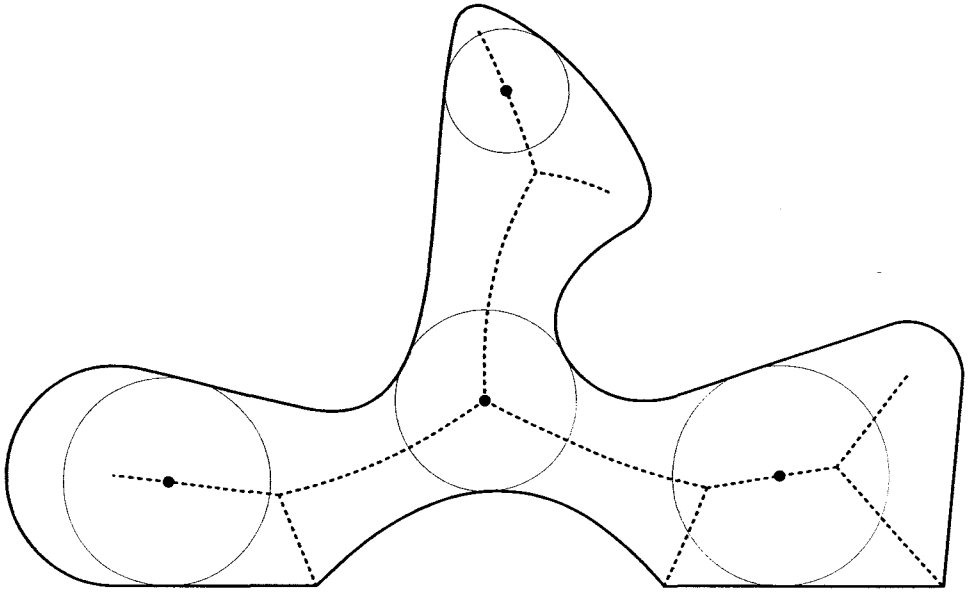


Figure 1.1 A pocket and its MAT

The *medial axis transform* (MAT) has emerged as an automatic shape interrogation tool for pocket milling [7,8]. The medial axis of a planar shape provides a compact representation of its geometric and topological characteristics. In planar domain, the *medial axis* (MA) of a close profile is the locus of centers of circles which are maximal within the profile, together with the limit points of the locus. The MAT is the *medial axis* together with the associated *radius function* which can calculate the radius of the maximal circle at any given point on the MA. By definition, the MAT fully describes the maximal circles that can be fitted into any location of the pocket profile to be machined, as shown in Figure 1.1. This geometric property provides an excellent cue to facilitate cutter size selection and gouging free tool paths generation. Efficient and accurate algorithms to calculate the MAT of free-

form curve boundaries will greatly benefit the process planning of the 2½-axis pocket milling operations.

1.2 Research Objectives

The first objective of this research is to develop efficient, accurate process modeling methodologies for the CNC milling. To achieve this objective, it is essential to model the material removal process to obtain the accurate in-process chip geometry. One goal of the research is to make the geometric model as generic as possible so the methodology can be applied to 2½- and 3-axis milling. And it is necessary to experimentally verify the developed process modeling system.

The second objective is to investigate and develop a computational efficient methodology to calculate the MATs of pockets with boundary of piecewise connected free-form parametric curves. The application of the MAT is to facilitate cutter size selection and gouging free tool paths generation for pocket machining.

Overall, this research is aimed at developing efficient and robust techniques and software tools to facilitate the selection of optimal cutting parameters for the CNC milling process.

1.3 Dissertation Organization

The remaining sections of this dissertation are organized as follows. Chapter 2 reviews the current technologies of geometric modeling, mechanistic modeling, CNC interpolator, medial axis transform and pocket machining. Chapter 3 presents a

geometric modeling methodology for 2½-axis milling with flat end mills. Experiments are conducted to verify the correctness and effectiveness of the developed milling process modeling system. Chapter 4 presents applications of the process modeling on optimal machining parameter selection, including steady cut optimization and automatic feed rate selection for pocket machining. Chapter 5 presents a geometric modeling methodology for 3-axis sculptured surface milling using general APT cutter. Chapter 6 develops a new, efficient geometric approach to approximate the MATs of general pockets with boundaries represented as piecewise connected free-form parametric curves. Chapter 7 proposed a new roughing tool path strategy based on the MAT, and a GA based algorithm is developed to identify an optimal set of cutter sizes. Chapter 8 summarizes this work.

Chapter 2 Literature Review

This chapter reviews the state-of-the-art technologies of geometric modeling, mechanistic modeling, CNC interpolator, medial axis transform and pocket machining. The limitations of the current methods are also discussed.

2.1 Geometric Modeling of the Milling Process

2.1.1 Cutter Swept Envelop

Extensive research has been carried out in geometric modeling of machining processes by studying cutter swept envelops, which represent the volumes being removed when the cutter moves along tool paths. According to their methodologies, the proposed methods normally can be classified into two categories: the mathematical formulation and computer graphics techniques.

Martin and Stephenson [9] initially applied the envelope theory from differential geometry to find volumes swept by surfaces. The swept surface is found by solving a system of implicit equations. Abdel-Malek et al. [10,11] refined this method by adopting the Jacobian rank deficiency method in swept volume calculation. A generalized method for determining swept volume for geometric entities with two or more parameters is carried out. Blackmore et al. [12,13] introduced a sweep-envelope differential equation (SEDE) method, which computes

swept volumes for three-dimensional smooth objects undergoing arbitrary smooth motions by employing a boundary flow formula. The grazing points only need to be computed at the initial point of the object, thus reduce the computational complexity.

The above mathematical formulation methods provide a general framework for calculating cutter swept envelopes. Several vector methods have been proposed as approximations based on the envelop theory. Chiou and Lee [14,15] developed an explicit representation of the swept profile for generic APT (Automatically Programmed Tools) cutter by applying a tangency function derived from the property that the cutter surface normal vector at a grazing point is perpendicular to the instantaneous moving direction at that point. Roth and Bedi et al. [16] compute grazing curve on a toroidal cutter by slicing the torus into two circles and calculating grazing points on each circle by a cross product method. Mann and Bedi [17] extend the same method to general surfaces of revolution. In above methods, the grazing points are connected to form a piecewise linear approximation to the grazing curve, and then these grazing curves are connected to form the swept surface. Weinert and Du et al. [18] applied the same tangency function to find the swept profile, and then approximated the swept volume with a NURBS (Non-uniform rational B-spline) surface representation.

Compared to modeling the cutter swept volume, several researchers have investigated the milling process by using swept sections. Sarma [19] studied flat-ended tool swept sections for five-axis CNC machining. The swept section here is

defined as the profile formed by the tool passing through the plane perpendicular to the tool path. Partial swept sections are calculated based on the plane-circle intersection in this work. Bohez and Minh et al. [20] used the sweep plane approach to approximate the swept volume at discrete cutter locations. The APT cutter intersects with a large number of flat planes, and general polygon clipping algorithm is used to build a geometric model of the swept volume. Accuracy is determined by slice polygons and distance between slice planes. This algorithm is implemented using the computer graphics technique.

2.1.2 In-Process Workpiece Geometric Model

Since the cutter swept volumes only represent the space being occupied during the tool motion, the instantaneous workpiece geometry during machining is also required for modeling the cutter and workpiece engagement (CWE) geometry.

For geometric modeling of the 2½-axis milling process with flat end mills, there are two types of methods that have been discussed most: the analytical formulation and solid modeler based technique. Gupta and Saini et al. [21] described a geometric algorithm to extract continuous closed-form function of the cutter engagement based on analytical techniques for a given workpiece and a cutter path. However, their method is restricted to parts with linear and circular edges, and the cutter path only consists of linear and circular moves. In practice, parts may be more complex, and the spline tool path already becomes a standard function of modern CNC machine tools. Compared to the analytical approach, the solid modeler based solutions can deal with more general machining operations. Spence and Altintas

[22] presented a Constructive Solid Geometry (CSG) based milling process simulation system for 2½ dimensional parts. The cutter-part intersection and an analytic mechanistic milling process model are used to predict cutting forces and re-schedule the feed rate. The CSG based approach is limited to strict 2½D milling with constant axial depth of cut. Actually, even in 2½D milling with flat end mills, the engagement along the cutter axis may not be constant in the following situations: (a) cutting a non-prismatic stock obtained from a primary process such as forging and casting; (b) cutting region previously machined at a different depth; (c) cutting with different cutter sizes; (d) cutting one part with different setups. Yip-Hoi and Xuemei [23] addressed part of these problems by applying a feature-based approach to characterize the cutter/workpiece engagement extracted from a solid modeler in 2½-axis end milling. Unfortunately, their work only deals with orthogonal engagement features.

For 3-axis sculptured surfaces milling, due to the complexity of workpiece geometry changes during machining, it is impractical to get the analytical solution of instantaneous cutter and workpiece intersection. Two types of approximate methods, solid modeling and discrete modeling, have been applied to the milling process simulation. Mounayri and Spence et al. [24] applied the CSG approach to the simulation of 3-axis milling of complex parts. Imani and Sadeghi et al. [25] developed a simulation system for 3-axis ball-end milling of sculptured surfaces by using a commercial solid modeler ACIS. The major problem of the solid modeling approach is the computational expense. The cost of simulation using the CSG approach is reported to be proportional to the fourth power of the number of tool

movements, $O(N^4)$ [26]. The solid modeling based simulation systems found in the literature only deal with related simple milling operations. It is impractical to use the CSG approaches on multi-axis sculptured surfaces milling with generic cutters.

Among discrete modeling approaches, the Z-map model is the most widely used model for NC verification [27]. Like the Z-buffer algorithm developed for computer graphics hidden surface removal, the Z-map model is a discrete non-parametric surface representation in which the heights at grid points are stored in a two dimensional array. Researchers have applied the Z-map to the machining process simulation. Yun and Hoon Ko et al. [28] developed a moving edge-node Z-map model for the cutting process simulation in transient cuts. Kim and Cho et al. [29] calculated cutting forces in 3-axis ball-end milling of sculptured surface by using a Z-map method to find the cutter contact area, but tool motions are limited to ramping and contouring of slanted surfaces. Zhu and Kapoor et al. [30] developed a mechanistic modeling approach to predicting cutting forces for multi-axis ball end milling employing the Z-map method. A typical problem with the Z-map model is that it cannot describe the cutter boundary between grids. If the cutter movement distance is less than the grid size, there is no node point in the machining region, which will result in substantial error. To increase the accuracy of the Z-map method, the number of nodes must be increase, which will increase the computational load consequently.

2.2 Mechanistic Modeling

Cutting forces modeling is one of the classic research areas of machining processes, and extensive research has been carried out [2,3,31-33]. The empirical cutting force models consider that cutting forces vary as functions of cutting conditions, normally including undeformed chip geometry (thickness, width), material properties and etc. The linear edge force model [4] is a widely used force model for the milling process. In this model, the instantaneous tangential (F_t), radial (F_r) and axial (F_a) cutting forces are modeled as

$$\begin{aligned} F_t &= K_{tc} bh + K_{te} b \\ F_r &= K_{rc} bh + K_{re} b \\ F_a &= K_{ac} bh + K_{ae} b \end{aligned} \tag{2.1}$$

where b is the width of cut, h is the uncut chip thickness, K_{tc}, K_{rc}, K_{ac} are the cutting coefficients contributed by the shearing action in tangential, radial and axial directions, respectively, and K_{te}, K_{re}, K_{ae} are the edge coefficients due to the rubbing of the tool flank against the workpiece.

For certain cutting tool versus workpiece material, the cutting forces coefficients can be identified by the orthogonal to oblique cutting transformation method or the mechanistic modeling method. The orthogonal to oblique transformation method uses orthogonal cutting parameters such as shear angle, shear stress, and friction coefficient to determine oblique milling constants. This method can be applied to a variety of milling cutter geometries, e.g. Lee and Altintas [34] obtained the cutting force coefficients for a ball-end mill from a set of

orthogonal cutting tests at various cutting speeds and feeds. Another advantage of this method is that the cutting constants of a cutter can be predicted even before the real cutter being manufactured, as long as the design of the cutter and corresponding orthogonal cutting database are available. However, some cutting tools may have complex cutting edges, and creating a time-consuming orthogonal cutting database may not be possible. In such cases, a quick method of calibrating the cutting constants, the mechanistic modeling approach, is used.

Mechanistic modeling of flat-end milling has been the focus of many studies [4,35-37]. Generally, for flat-end mill with constant helix angle, the cutting coefficients are constant values which can be identified from the average cutting forces per tooth period measured from a set of milling experiments conducted at different feed rates but constant immersion and axial depth of cut.

For more general cutting tools, such as ball-end, bull-nose and tapered end mill, the helical angle can be designed arbitrarily and may vary from cutter to cutter. Popular designs include constant lead and constant helix angle [38]. Mechanistic modeling methods have been developed to identify cutting force coefficients for general cutters. Feng and Menq [39,40] presented a mechanistic model for the prediction of cutting forces in the ball-end milling process. The cutting coefficients for the cutting edges on the ball part are approximated by polynomial expressions along the axial direction. Azeem and Feng et al. [41] developed a simplified method to calibrate the empirical force coefficients of a ball-end mill by performing a single half-slot cut experiment. Gradiek and Kalveram et al. [42] presented expressions for

semi-empirical mechanistic identification of specific cutting and edge force coefficients for a general helical end mill from milling tests at an arbitrary radial immersion. Wan and Zhang et al. [43] presented a unified approach to identify the cutting forces coefficients together with the cutter runout parameters for general end mills. The cutting forces are analyzed and separated into two terms a nominal component independent of the runout and a perturbation component induced by the runout. The instantaneous cutting force coefficients are calibrated from the nominal component, and the runout parameters are determined from the perturbation component.

2.3 CNC Interpolator

During machining, the relative velocity between the cutter and workpiece is related with the machine kinematics and controller functions. For accurate process modeling, the real-time CNC interpolators need to be studied and integrated into the simulation system for accurate evaluation of cutter locations along the tool paths. In practice, tool paths are transferred to the CNC machine tool as part programs, sometimes called G-code. The motion commands of the part program are processed in real-time by the CNC interpolator, in order to generate the reference commands for the control loop to drive the machine tool movements [44].

The CNC interpolator operates at a small sampling time interval, which is a constant dependent on the control loop's sampling rate. It is common to have a CNC that generates motion commands in 0.002s intervals for today's machine tool. For a

general curve, given the desired feed rate along the curve and current reference point on the curve, the task of the real-time interpolation algorithm is to find the next reference point in one sampling interval to which the tool will be driven to at the desired feed rate. Interpolation algorithms based on the first order Taylor's approximation are widely used in the implementation of parametric interpolators [45-48].

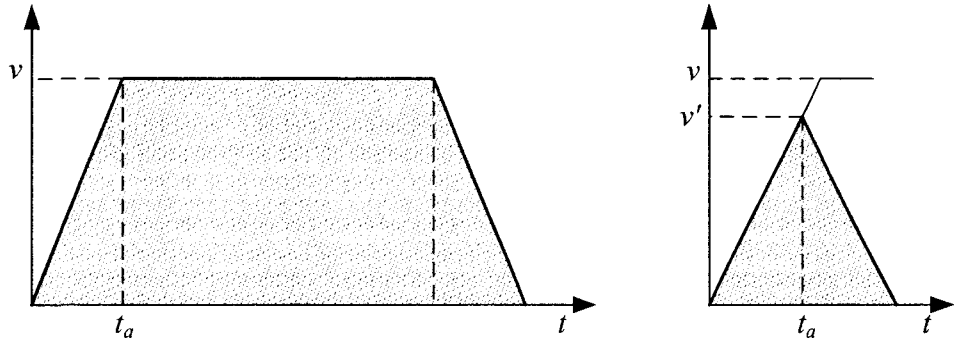


Figure 2.1 Trapezoidal velocity profile for real-time interpolator

To prevent mechanical shock, different types of acceleration/deceleration schemes are used to control the feed rate along the tool path. Trapezoidal velocity profile is a simple and widely used scheme with constant acceleration and deceleration [49,50]. Because of the automatic acceleration and deceleration applied at the beginning and end of each tool path segment, the actual feed rate along a tool path is not constant. If the segment is too short, it may never reach the desired feed rate.

To apply a velocity profile, the CNC interpolator begins with calculating the length of the tool path segment, and then calculates the acceleration/deceleration time and the total motion time according to the given feed rate. The velocity profile

is used to determine the velocity at any sampling time. Finally, the real-time interpolation formula (based on the first-order Taylor's approximation) is used to calculate the increase of parameter value. This approach is integrated in our simulation system to calculate the actual cutter location.

2.4 Medial Axis Transform

The medial axis concept was first proposed by Blum [51] in 1967 to characterize biological shapes. Since then, it has been actively developed and used in many fields, such as NC milling, image processing, computer vision, solid modeling, and mesh generation, etc.

The differential and topological properties of the MAT were studied [52,53] and then a general framework for developing practical algorithms to calculate the MAT is rendered. For profiles with piecewise-linear/circular boundaries, whose bisectors are conics and can be solved analytically, algorithms have been developed to construct the exact MAT. Culver et al. [54,55] presented a tracing algorithm to compute the exact medial axis for polyhedral. Their algorithm applies modular arithmetic, floating point filters, lazy evaluation, and spatial subdivision techniques. This approach is accurate and robust; however, its computation time is long, and it is limited to geometry with known bisectors. For pocket shape with free-form curve boundaries, because there is no simple exact solution to general curve/curve bisectors, exact computation of the MAT is difficult, sometimes impossible. Consequently, most practical algorithms approximate the MAT in different ways.

The Voronoi diagram based approaches have been developed by many researchers to approximate the MAT. Computing the Voronoi diagram of a group of point sites has been extensively studied in computational geometry [56,57]. Several efficient algorithms with $O(n \log(n))$ complexity are available even with free program source codes [58]. It has been proposed by researchers that the medial axis of a shape can be computed using the Voronoi diagram of discrete sample points of the shape boundary and the Voronoi vertices converge to the medial axis for a curve in 2D as the sample density approaches infinity. Brandt [59] examined the finite approximations of the continuous-domain skeleton and showed that the Voronoi diagram converge appropriately when the Voronoi function of the shape boundary is a continuous function. Dey and Zhao [60,61] proposed an algorithm that approximates the medial axis straight from the Voronoi diagram in a scale and density independent manner with convergence guarantees. Their approach does not pay any special attention to poles, but rather computes a sub complex from the Voronoi diagram that lies close to the medial axis and converges to it as sampling density approaches infinity.

Driven by the need for applications, the research of Voronoi diagram has been extended to sites of free-form curved objects. Chou [62] developed a tracing algorithm to calculate Voronoi diagram for planar, simply connected, closed shape with boundaries composed of free-form curve segments. The algorithm generates the Voronoi diagram by tracing the bisectors starting from terminal points, such as convex corners and centers of the maximal, positive local curvatures. Ramamurthy and Farouki [63,64] presented an incremental approach to computer Voronoi

diagrams based on thorough investigation of point/curve and curve/curve bisectors. They described the similarities and differences between the Voronoi diagram and the medial axis, and obtained a medial axis from a Voronoi diagram by removing certain edges of the diagram that do not belong to the medial axis, and adding certain edges that do belong to the medial axis but are absent from the diagram. Helmut et al. [65] presented a randomized incremental algorithm to compute the Voronoi diagram of curved objects by breaking up the curves into "harmless" pieces, which include line segments, circular arcs and spiral arcs.

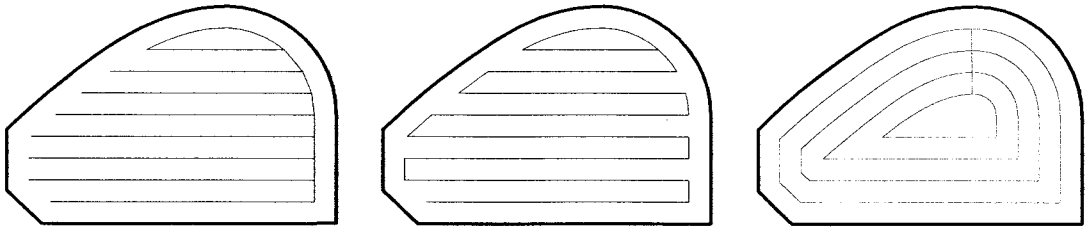
Beside Voronoi diagram based approaches, the wave-front propagation and divide-and-conquer methods have been published [66-69]. Gursoy and Patrikalakis [66,67] proposed a wave-front propagation method of the MAT through determining inward offset distances and the associated branch points at which the topology of the contour changes. Their method analytically defines the MA in terms of conic sections, and free-form boundary curves are approximated within a prescribed tolerance using line and circular arc segments. Likewise, Hassouna and Farag [68] presented a wave front propagation method. The front propagates at each object point with a speed that is proportional to its Euclidean distance from the boundary. The motion of the front is governed by a nonlinear partial differential equation and is solved using level set methods. Then the medial axis is computed by tracking the propagating front points with the maximal positive curvature. Choi et al. [69] presented an approximation algorithm based on the domain decomposition lemma. A tree data structure and its related operations are developed to keep track of the information produced by the domain decomposition procedure. This

algorithm adopts a "from within" approach to calculate contact circles of medial axes.

2.5 Pocket Machining

Pocket machining is one of the most common operations in NC machining applications. This is due to the facts that most mechanical parts consist of boundary faces parallel or normal to a reference plane, and more complicated free-form parts are usually produced from a raw stock by 2½-axis roughing and 3/5-axis finishing. This machining process includes first rough pocketing for higher cutting efficiency and second finish machining for better pocket accuracy.

Conventional tool path strategies for pocket machining can be classified into two major types, direction-parallel and contour-parallel, see Figure 2.2. These two tool path strategies are provided by most commercial CAM software.



(a) Direction-parallel path One way

(b) Direction-parallel path Zigzag

(c) Contour-parallel path

Figure 2.2 Conventional pocketing tool path strategies

The direction-parallel tool path is generated from straight line segments parallel to an initial reference line. According to the connection between line segments, direction-parallel path can be further classified into one-way and zigzag.

The one-way path can maintain a consistent up milling or down milling. One drawback of the one-way path is that a significant amount of time is used for the cutter to return to the start position of each path. Thus the machining efficiency is reduced. The zigzag path can reduce the non-productive time, but the alternative up and down milling cut may lead to problems such as short tool life and poor kinematic of the machine tool. The contour-parallel tool path is derived from the pocket contour. The pocket area is milled in a spiral-like fashion cutting along offset curves to the contour and stepping outwards or inwards for the next pass. The cutter is kept in contact with the workpiece material most of the time, and a consistent up or down milling can be maintained. Generally, less idle time is spent in the contour-parallel path than the direction-parallel path, and the contour-parallel strategy is widely used for pocket machining.

In present machining companies, for roughing a pocket, a large tool is selected subjectively by NC programmers to generate direction-parallel or contour-parallel tool paths with CAM software; for finishing the pocket, a small tool is selected conservatively and tool paths are planned by offsetting the pocket boundary curves. The size of the end mill cutter chosen for machining the given pocket has a significant impact on the machining time. With the availability of high-speed automatic tool change mechanisms in modern CNC machines, tool change can be achieved within seconds. To take full advantage of the high-speed tool change capability, multiple cutters can be selected to achieve higher efficiency. However, due to the complicated pocket shape, it is difficult to automatically select the optimal

cutter sizes. Several researchers have addressed the topic of using multiple cutters in pocketing.

Hatna et al. [70] conducted a review of literature on pocket machining published before 1998. The paper shows the relationships between the shape, cutter, machine tool, and cutting conditions in the process of optimizing the cost of and leading time of pocket milling. This paper also mentioned an important method of planning tool paths using Voronoi diagram. Among the Voronoi diagram based articles published until now, Held [7] firstly established a mathematical theory of introducing Voronoi diagram in pocketing. Further, Veeramani and Gau [71,72] employed a concept called the Voronoi mountain to calculate the material volume that can be removed by a specific cutter size, and a dynamic programming approach was used for optimal selection of cutter sizes on the basis of the process time. Hinduja and Sandiford [73] developed a procedure to determine the optimum pair of tools in terms of the cutting conditions and tool path length. The geometrical constraints including minimum concave radius, bottleneck width and the entry distance are determined from the Voronoi diagram. Nadjakova and McMains [74] described an approach to finding an optimal set of cutter radii for machining a pocket by using the Voronoi diagram. Elber et al. [8] provided a scheme to generate trochoidal tool paths for high speed machining of free-form pockets, based on its Voronoi diagram. Chen and Zhang [6] used the maximum circle radius graph with regard to the medial axis to find the largest cutter to finish cut the pocket free-form boundary without gouging and interference.

Besides the Voronoi diagram based methods, several other methods have been proposed. Li et al. [75] investigated various feasible tool path patterns for a single tool to cut sculptured parts layer by layer in 2½D rough machining. An intelligent approach to automatically identifying the most productive tool path pattern for a given cutting layer was introduced. Yang et al. [76,77] decomposed pockets into regular features to best fit multiple cutters of various sizes so that they can efficiently cut their corresponding features without overcut, and then Joneja et al. [78] applied this greedy technique to different layers of pockets with sculptured surfaces. Lim et al. [79,80] developed a method for determining a theoretical optimal combination of cutting tools given a set of 3D volumes or 2D profiles of the selected machining features. Optimal tools can be selected by considering residual material that is inaccessible to oversized cutters and the relative clearance rates of cutters that can access these regions. Yao et al. [81,82] presented a geometric algorithm for finding an optimal set of milling cutters for machining a given set of parts. The problem of finding an optimal sequence of cutters is represented as a least-cost path problem, and is solved using Dijkstra's algorithm. Zhang and Li [83] tried to select multiple tools to achieve the optimal roughing of pockets with arbitrary shape in terms of least machining time and the maximum material removal rate. They improved the scan-line filling algorithm to compute the cumulative cutting area for every tool to estimate the contribution of different tool diameters. Narayanaswami and Choi [84] provided a grid-based 3D navigation algorithm for generating NC tool-path for three-axis CNC milling of general pockets with sculptured bottom surfaces. The pocket surface is discretized by defining a grid

and the navigation algorithm plans the tool motion. The grid size and the cutter diameter are chosen so that a predefined tolerance for surface roughness is satisfied. Shih and Chuang [85] proposed a robust method to generate tool paths for NURBS-based machining of arbitrarily shaped freeform pockets with islands. Although the input and output are all of higher degree NURBS curves, only one simple category of geometric entities, i.e., line segments, is required for initial offsetting and for detecting and removing self-intersecting loops. Furthermore, using those linear non-self-intersecting offsets as the legs of NURBS control polygons, NURBS-format tool paths can be smoothly reconstructed with G1-continuity, no overcutting, no cusps, and global error control. D'Souza et al. [86] developed a method to find the lowest cost tool sequence for rough machining free-form pockets. The free-form pocket is approximated to within a predefined tolerance of the desired surface using series of 2.5-D layers of varying thicknesses that can be efficiently removed with flat-end milling cutters. A graph-based method is used to find an optimal sequence of tool for rough machining the approximated pocket.

2.6 Summary

In this chapter we reviewed current techniques closely related to this research, including geometric modeling of the cutter swept envelop and the in-process workpiece model, mechanistic cutting forces modeling of the milling process, CNC interpolator, medial axis transform algorithms and pocket machining. Some of the current methods still suffer certain limations. In the following chapters, new methods for modeling the cutter workpiece engagement geometry and its

integration with existing mechanistic cutting force models are presented for 2½- and 3-axis milling operations, and a new efficient algorithm for calculating medial axis of planar shapes with free-form boundaries is developed.

Chapter 3 Process Modeling for 2½-Axis Milling

3.1 Introduction

In the 2½-axis CNC milling process, the cutter stays at the same height in Z direction when it machines on a layer parallel to the worktable, the X-Y plane. For the term 2½-axis, "2" means the X- and Y-axis motions while cutting, and "½" indicates the discrete Z-axis motion which step-by-step positions the cutter on different machining planes. Most of the 2½-axis milling is carried out on 3-axis CNC milling machines. Due to the high material removal rate and the rigid setup, the 2½-axis milling is widely used for rough machining.

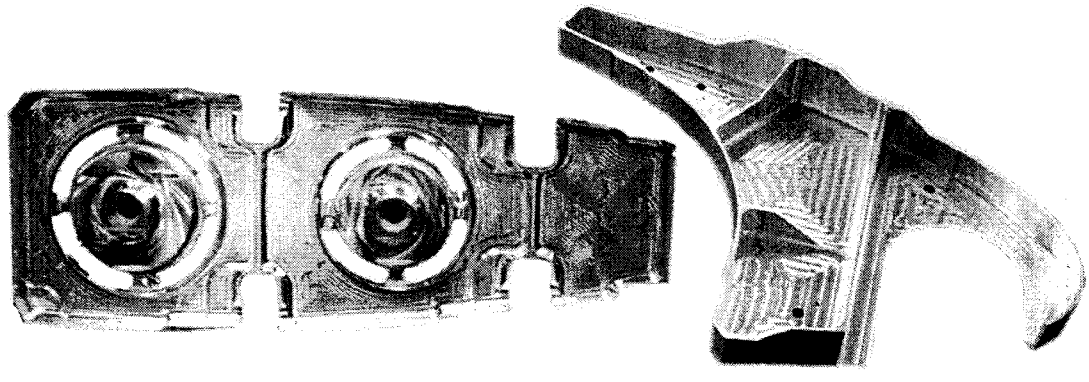


Figure 3.1 Parts machined by 2½-axis milling

This chapter presents a geometric modeling methodology for 2½-axis milling with flat end mills. A Z-layer B-Rep model is proposed to represent the in-process

workpiece of a part, and an innovative geometric approach is used to find the engagement geometry of different tool motions. The Z-layer B-Rep model is created from the initial workpiece geometry and updated after each cut during machining. The proposed approach can handle complicated multiple cutter-workpiece engagements and can accurately compute the chip thickness. By integrating the geometric modeling results with a mechanistic cutting force model, a 2½-axis milling process simulation system is developed. The developed system is experimentally verified by comparing the simulation results with cutting forces data collected from actual cutting tests.

3.2 Geometric Model of the Chip Geometry

3.2.1 In-Process Model Definition

The in-process model is defined as the intermediary state of the workpiece during the machining process. Since the intersection between the cutter and the workpiece varies along the tool path, the in-process model changes accordingly. In this work, it is generated by slicing the initial workpiece stock into layers by horizontal planes at different z levels. Each layer contains a z position, height and a 2D profile which represents the shape of this layer. A layer can be treated as a disc of stock material, and all layers stack together to form the in-process workpiece model. The planes used to slice workpiece stock are determined by taking into account both part geometry and 2½-axis tool path property. Three types of slicing planes are used to determine the z levels.

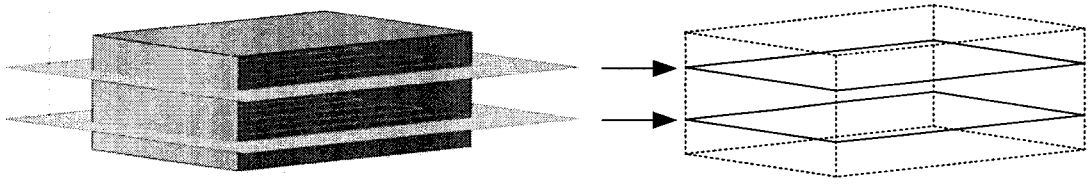


Figure 3.2 In-process model layers determined by tool paths

[Type 1] Typical 2½D tool path cycle includes three stages, which are approaching vertically, machining at horizontal plane, and retracting. At each horizontal machining stage, the flat end mill moves at a plane parallel to X-Y plane, this plane is used to slice the workpiece.

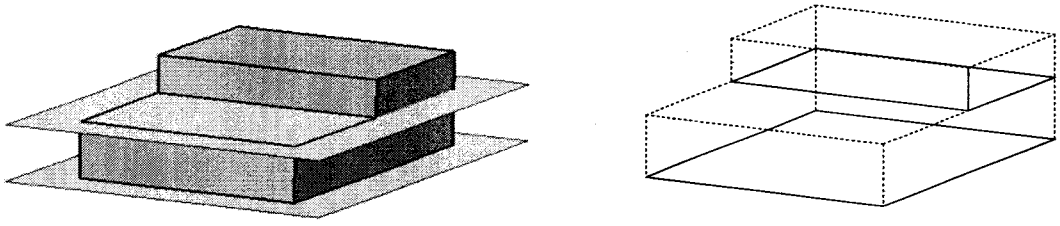


Figure 3.3 In-process model layers determined by stock geometry

[Type 2] If the initial workpiece stock only has boundary faces parallel or normal to the XY-plane, each face parallel to XY-plane will determine a layer, except the top surface of the stock, because there is no material above.

For rectangular prismatic workpiece stocks, the layer based in-process model is exactly the same as the input workpiece stock geometry. But sometimes stock obtained from primary processes such as forging and casting may not always be rectangular shape. A discretization method is used to deal with this situation.

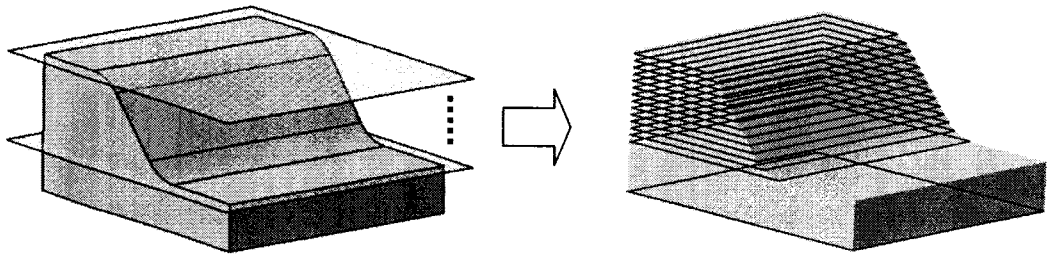


Figure 3.4 Discrete in-process layers for non-prismatic geometry

[Type 3] If the initial workpiece stock has geometric features which are neither parallel nor normal to the XY-plane, a discretization technique is applied to slice these features into a number of horizontal layers. The number of layers is controlled by the desired accuracy.

Once the slicing planes are determined, the initial 2D profile for each layer is calculated by intersecting the slice plane with the stock geometry. While the part is being machined by a cutter moving along tool paths, the 2D profile represents the boundary of material left on each layer. In order to deal with real machining, the data structure of the 2D profile should have the following properties:

- Islands may exist due to the previous cut or initial stock with islands. Then, one or more loops are required to describe one layer of the complex workpiece shape during machining.
- Circular holes may exist due to drilling processes before milling, or irregular shape holes due to primary process (forging or casting). Inner loops are required to describe the holes.
- In order to represent material side and empty space side, the direction of each loop must be defined properly. The outer loops are defined as counter-

clockwise and the inner loops are defined as clockwise. Then the material always lies on the left side of the loop boundary.

- There is no self-intersection at each layer.

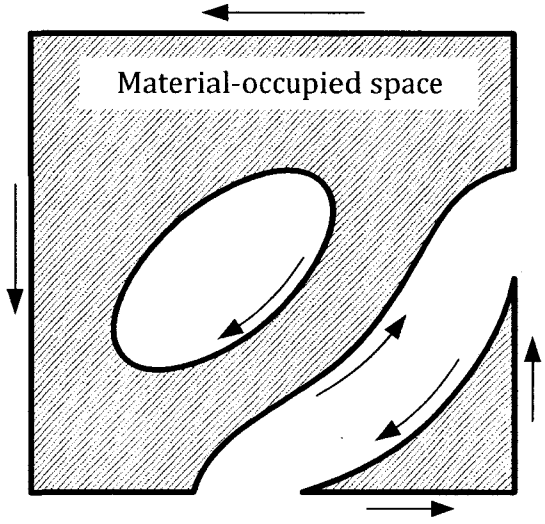


Figure 3.5 B-Rep data structure of the layer profile

Based on the above geometry and topology properties, a B-Rep data structure (see Figure 3.5) is used to describe the layer profile, which is defined as a set of outer counter-clockwise loops with zero or several clockwise loops inside. For the computational efficiency, a polygon is used to represent each loop.

3.2.2 Cutter Workpiece Engagement Geometry

The process of extracting cutter workpiece engagement geometry includes three steps. The first step is to find the layers whose z position is equal to or higher than the z position of the cutter location. These are the layers which are machined and will be processed in the following steps. Then, a semi-circle, which represents the cutter frontier, moves along the tool path to extract the cutter/workpiece

engagement profile at each machined layer. Finally, the engagement geometry is obtained by combining all extracted layer engagement profiles with their z position and height. The result engagement geometry has the same data structure as the in-process model, a Z-level B-Rep representation. The material removal rate (MRR) can be estimated from the volume of the removed material.

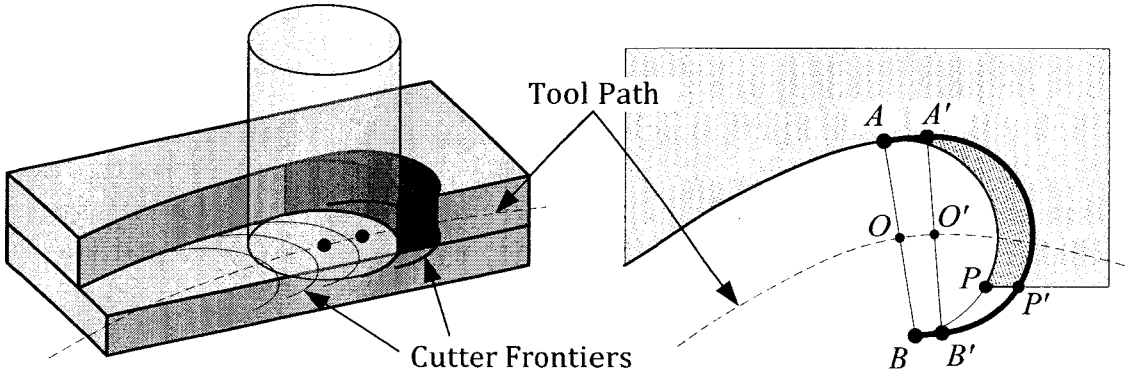


Figure 3.6 Extraction of the engagement profile

At each layer, the advancing semi-circle method is illustrated in Figure 3.6. At one cutter location point O, the cutter circle can be divided into two parts by a line perpendicular to the cutting direction and passing through the center of the cutter. The semi-circle facing the cutting direction is defined as the cutter frontier. Then, the next cutter location O' is determined by moving the cutter along the tool path while it rotates one revolution. In practice, tool paths are transferred to the CNC machine tool as part programs, and are processed in real-time by the CNC interpolator. In order to obtain the cutter location accurately, the real-time interpolator algorithms reviewed in Section 2.3 are integrated into our simulation system for calculating the next cutter location O'.

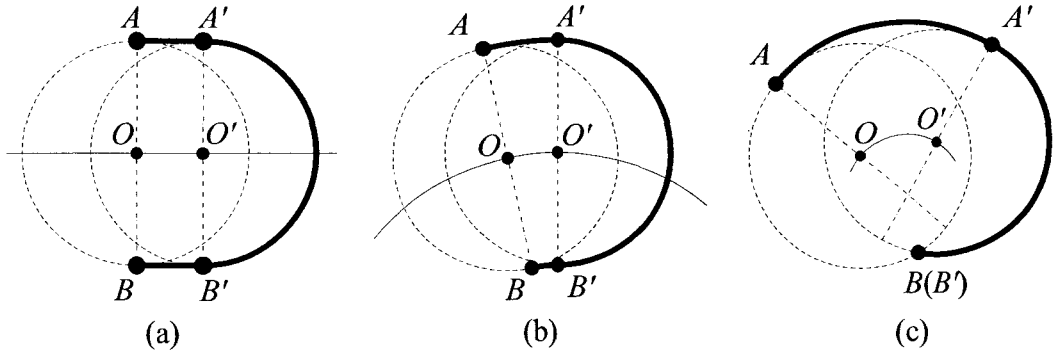


Figure 3.7 Cutter frontier swept profiles

When the cutter moves from O to O', the cutter frontier generates a swept profile. There are three types of tool motions provide by CNC machine tools: linear, circular and NURBS. By represent each tool path segment as a parametric curve, the radius of curvature can be calculated at any location [87]. For linear tool path, Figure 3.7 (a), the cutter swept profile composes two line segments $\overline{AA'}$, $\overline{BB'}$ and a half circle $A'B'$ (the cutter frontier at O'). For circular or NURBS tool path, if the radius of curvature is larger than the cutter radius, Figure 3.7 (b), the swept profile composes AA', BB' (offset curves of segment OO') and cutter frontier $A'B'$; if the radius of curvature is smaller than the cutter radius, Figure 3.7 (c), the swept profile composes two segments, AA' and $A'B'$, where $B(B')$ is the intersection point of two cutter circles at O and O'.

The cutter workpiece engagement profile can be extracted by calculating the intersection points between the in-process workpiece profile and the cutter frontier swept profile. At the same time, the in-process profile is updated after the corresponding stock material is removed. In the case shown in Figure 3.6, first, the pair of intersection points A and P' are found between the cutter frontier profile and

the in-process workpiece profile. Then, the in-process profile is updated by replacing the polygon segment between A and P' with segment $AA'P'$ (the portion of the cutter frontier swept profile between points A and P'). The engagement profile is created by combining the removed portion APP' from the workpiece profile and $AA'P'$ of the cutter frontier swept profile. Since the in-process workpiece profile is represented with polygons, the arc segment $A'P'$ is converted to a polygon based on the desired accuracy before being inserted into the workpiece profile.

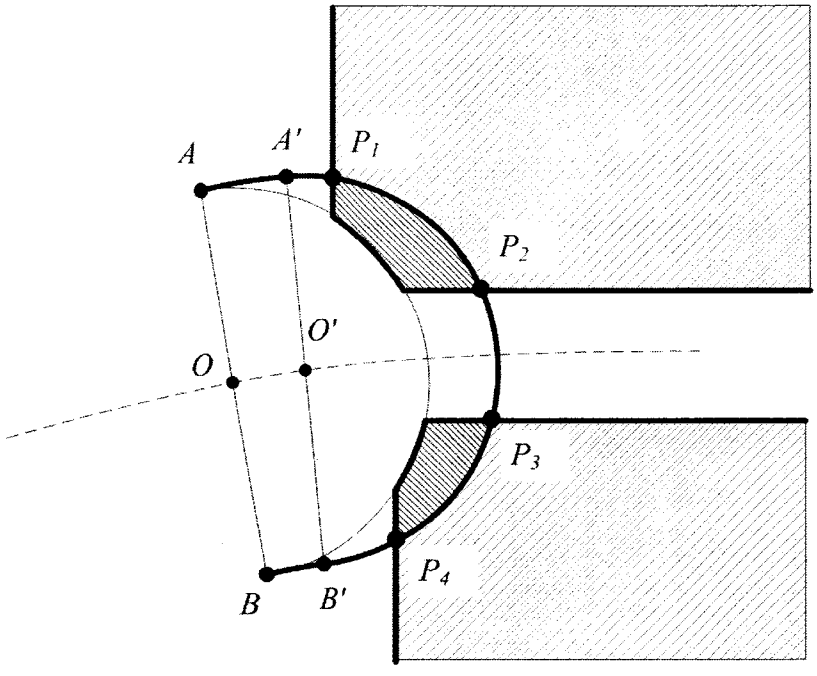


Figure 3.8 Extraction of multi-engagements

More complex engagement situations, such as the multi-engagements in Figure 3.8, can be handled by arranging intersection points in pairs and updating the workpiece profile respectively.

3.2.3 Undeformed Chip Thickness

In conventional methods [31], the chip thickness is approximated as

$$h(\phi) = f_c \sin\phi \tag{3.1}$$

where f_c is the feed per tooth and ϕ is the instantaneous angle of immersion. This approximation is good when the cutter moves along a straight line. For curve tool motions, especially curves with large curvature, this approximation is not valid anymore.

In this work, we calculate the undeformed chip thickness from the extracted cutter/workpiece engagement geometry at any instantaneous immersion angle. Since the cutter can move along any tool paths, the part coordinate system is used as the reference. To determine the chip thickness along the radial direction, the extracted engagement profile is intersected with a ray, which starts from the cutter center and inclines at the immersion angle ϕ. The distance between two intersection points A and B represents the thickness of the material removed in this revolution. Then, the undeformed chip thickness function $h(\phi)$ is calculated as:

$$h(\phi) = |AB| / N \tag{3.2}$$

where N is the number of flutes.

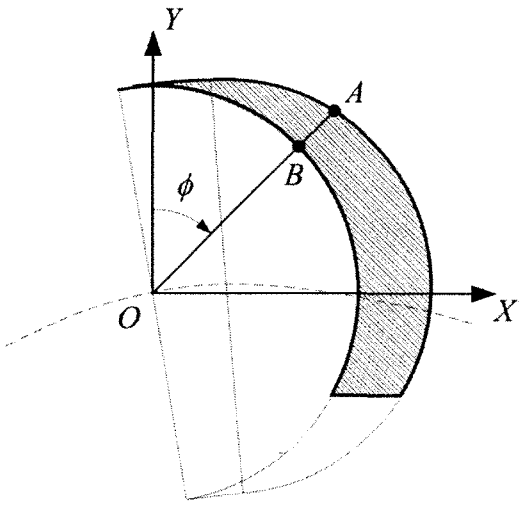


Figure 3.9 Undeformed chip thickness

3.3 Mechanistic Model of 2½-Axis Milling

In order to calculate the cutting forces, the end mill cutter is modeled by several helical flutes wrapped around the cutting surface, and is divided into a number of small differential discs along the cutter axis. A cutting disc at axial height z is shown in Figure 3.10.

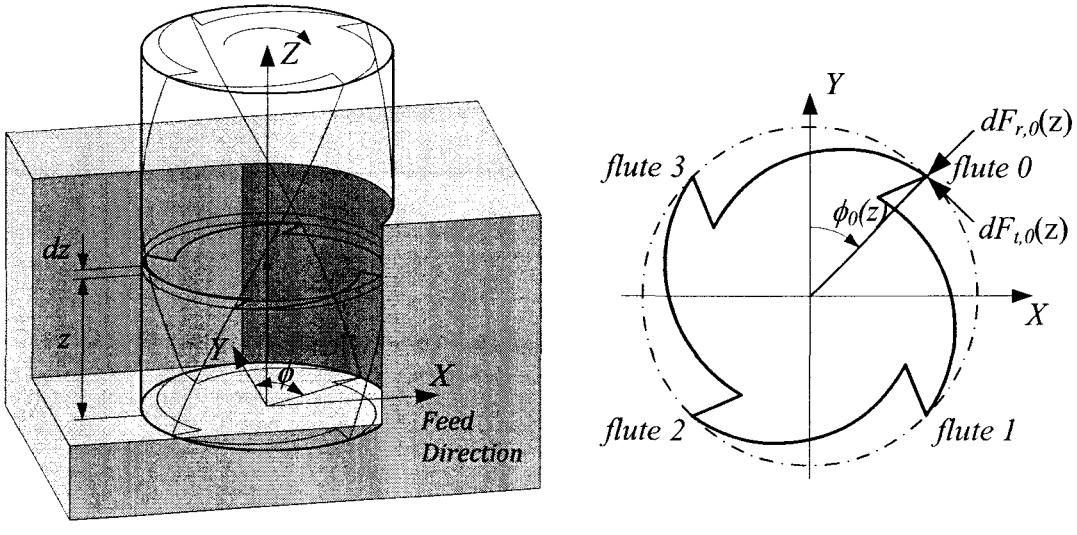


Figure 3.10 Mechanics of end milling

An end mill with helix angle β, radius R and N number of flutes is used. The immersion angle ϕ at the bottom end of one flute is used as the reference, then the bottom end points of other tooth are at $\phi_j = \phi + j \cdot \phi_P$, $j = 0,1,...(N-1)$, where the cutter pitch angle $\phi_P = 2\pi/N$. At axial height z the lag angle is $\psi = z \cdot \tan\beta / R$. For the jth flute, the angular position at axial height z is $\phi_j(z) = \phi + j\phi_P - \psi$, and the differential tangential $(dF_{t,j})$, radial $(dF_{r,j})$ and axial $(dF_{a,j})$ cutting forces are calculated using the linear edge force model:

$$\begin{aligned} dF_{t,j}(z) &= \left[\mathrm{K_{tc}} h_j(\phi_j(z)) + \mathrm{K_{te}} \right] dz \\ dF_{r,j}(z) &= \left[\mathrm{K_{rc}} h_j(\phi_j(z)) + \mathrm{K_{re}} \right] dz \\ dF_{a,j}(z) &= \left[\mathrm{K_{ac}} h_j(\phi_j(z)) + \mathrm{K_{ae}} \right] dz \end{aligned} \tag{3.3}$$

where $h_j(\phi_j(z))$ is the undeformed chip thickness, dz is the differential flute element height, K_{tc}, K_{rc}, K_{ac} are the cutting coefficients contributed by the shearing action in tangential, radial and axial directions, respectively, and K_{te}, K_{re}, K_{ae} are the edge constants. The undeformed chip thickness function $h(\phi)$ is evaluated geometrically from the cutter/workpiece engagement geometry, see Eq. (3.2).

The instantaneous cutting torque on the spindle is

$$dT_c(z) = R \cdot \sum_{j=0}^{N-1} dF_{t,j}(z) \tag{3.4}$$

The forces are then projected to the Cartesian tool coordinate system

$$
\begin{aligned}
dF_{x,j}(z) &= -dF_{t,j}\cos\phi_j(z) - dF_{r,j}\sin\phi_j(z) \\
dF_{y,j}(z) &= dF_{t,j}\sin\phi_j(z) - dF_{r,j}\cos\phi_j(z) \\
dF_{z,j}(z) &= dF_{a,j}
\end{aligned}
\tag{3.5}
$$

The total cutting forces F_x, F_y, F_z and torque T_c can be obtained by integrating the differential elements along the cutter axis. The axial integration limits for each flute are determined from the cutter workpiece engagement area obtained by combining all engaged layers, as shown in Figure 3.11.

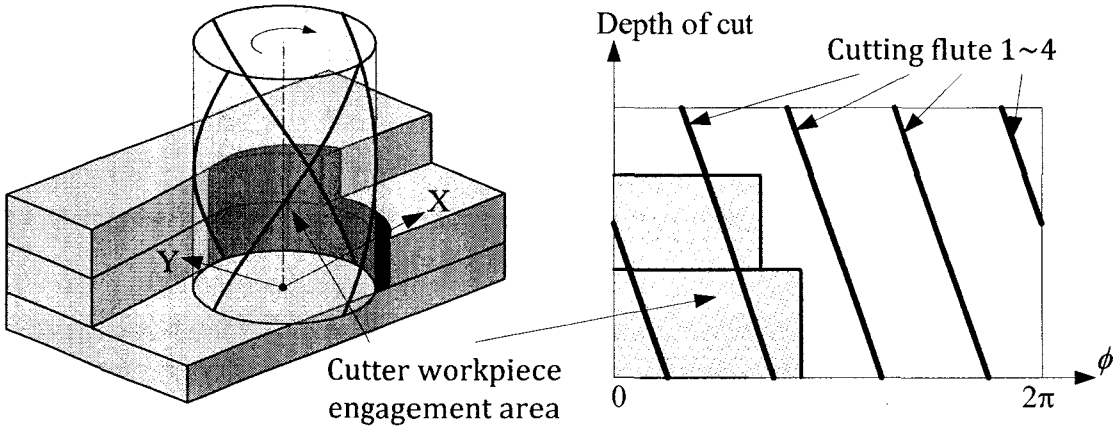


Figure 3.11 Cutter and workpiece engagement area

The resultant blending force on the cutter is defined as the vector summation of the x, y forces, which has a magnitude

$$
F = \sqrt{F_x^2 + F_y^2} \tag{3.6}
$$

The cutting power draw from the spindle is calculated as

$$
P = 2\pi n \cdot T_c / 60 \tag{3.7}
$$

where n (rev/min) is the spindle speed.

3.4 Computer Implementation

The proposed methodology has been implemented on personal computer by using the C++ programming language for calculation and OpenGL for visualization. The implementation is a process simulation system for 2½-axis end milling. The inputs to the system include tool paths, workpiece geometry, CNC interpolator parameters, cutter geometry and cutting constants, and the outputs are cutting forces, torque, power, material remove rate, chip and workpiece geometry at each instantaneous moment. By adopting industrial standard file formats, including APT files for exchange tool paths with CAM software and IGES files for exchange geometry with CAD software, the system can be easily integrated with commercial CAD/CAM software packages.

Two examples, shown in Figure 3.12, illustrate the system's capability to extract cutter/workpiece engagement geometry for different machining situations. Figure 3.12(a) shows a rectangular workpiece machined by an end mill along curved tool paths. The axial engagement remains constant while the radial engagement varies. Figure 3.12(b) shows a non-prismatic workpiece being machined. The engagement geometry changes in both axial and radial direction when the cutter passes the cylindrical feature.

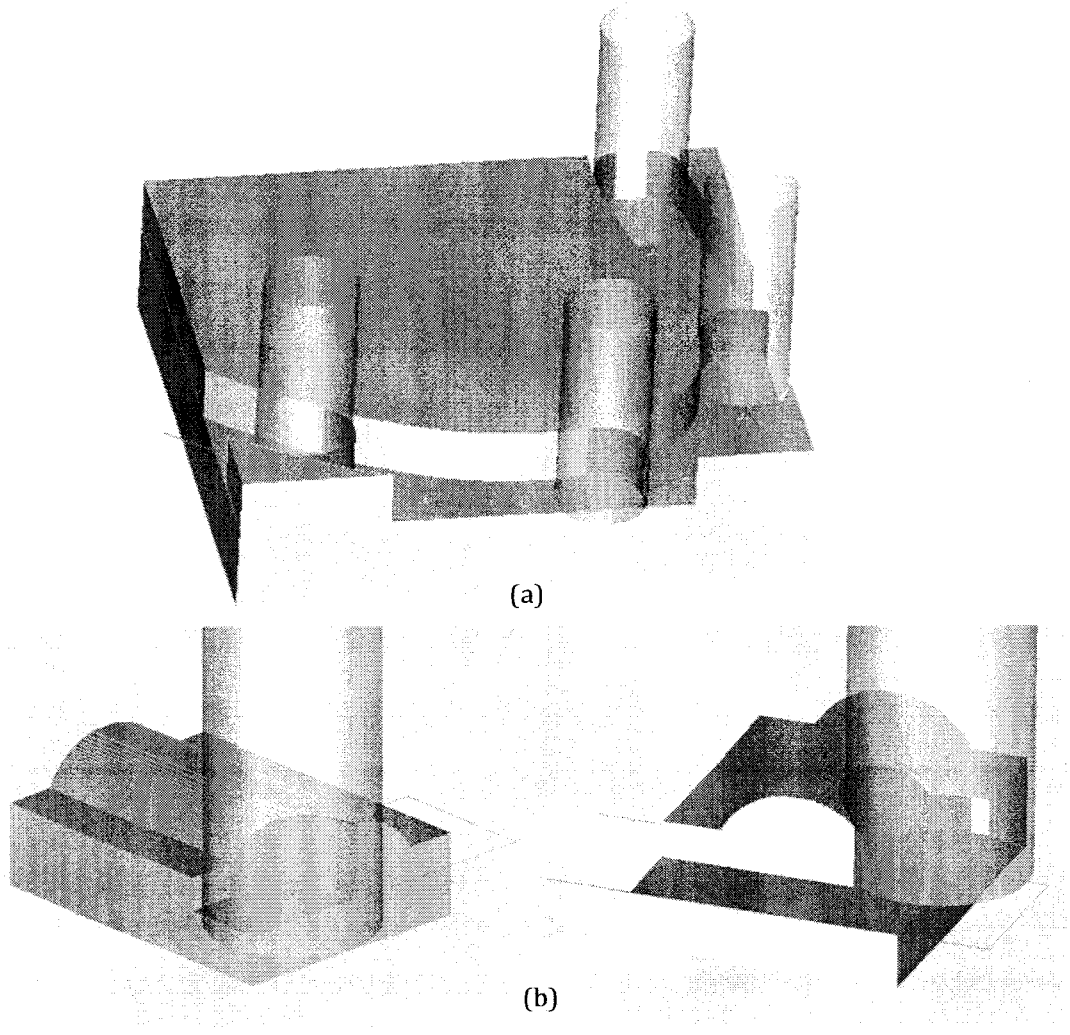


Figure 3.12 Illustrations of CWE geometry

3.5 Experimental Verification

Instantaneous cutting forces and torque can be predicted from the extracted CWE geometry. In order to verify the developed system, experimental verification is performed by comparing the simulation results with the cutting forces measured from actual cutting tests.

3.5.1 Experimental Setup

The cutting tests are conducted on a Makino A81M-5XR horizontal machining center. The cutting forces are measured along the x, y and z axis by a Kistler 9255B 3-axis dynamometer connected to a Kistler 5011B charge amplifier. The data acquisition is done by a laptop computer equipped with a National Instruments DAQCard-AI-16E-4 data acquisition card.

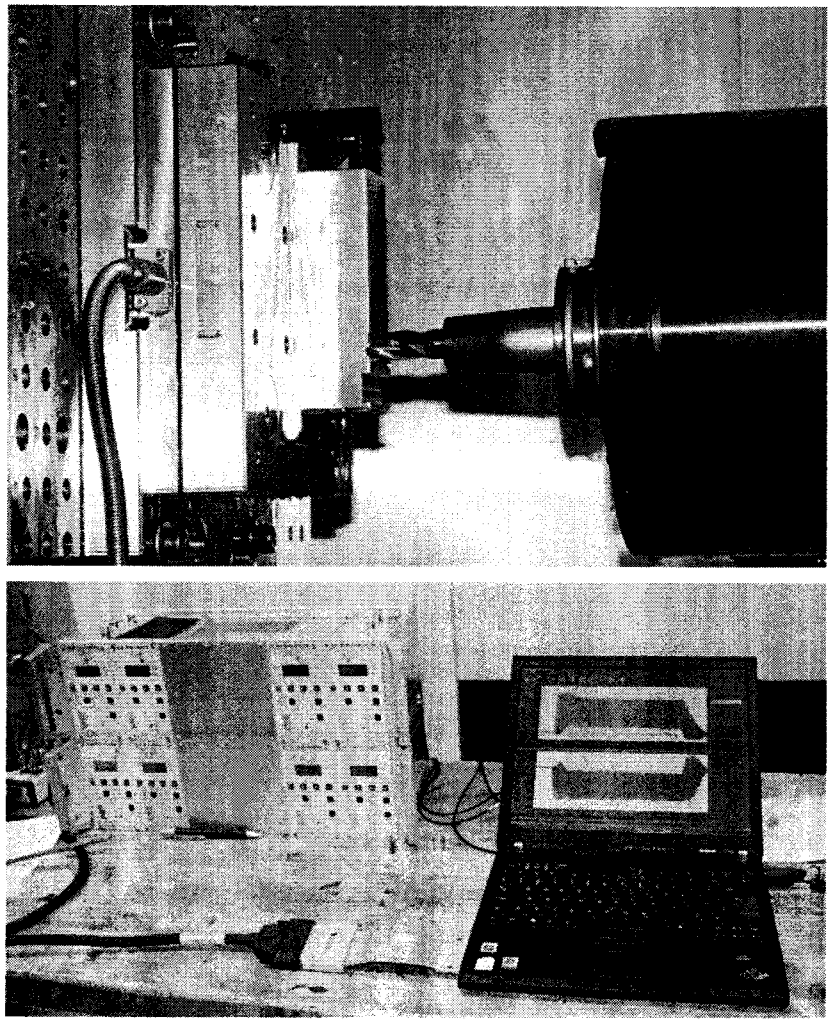


Figure 3.13 Experimental setup for cutting forces measurement

A solid carbide flat-end mill cutter with *4* flutes, *30 degree* helix angle and *19.05mm* diameter is used to cut aluminum alloy Al7075 workpiece material with cutting fluid. The spindle speed used in all experiments is *1000 RPM*. The data acquisition sampling frequency used is *10000 Hz*, which gives *600* sample points per cutter revolution.

3.5.2 Determining the Cutting Force Coefficients

The cutting forces coefficients of the experimental cutter are determined through a set of full immersion slot cutting at a constant axial depth of cut (*5mm*) and different feed rates (*200-700 mm/min*). Linear regressions of the average cutting forces measured are shown in Figure 3.14, and the calculated cutting force coefficients are given in Table 3.1.

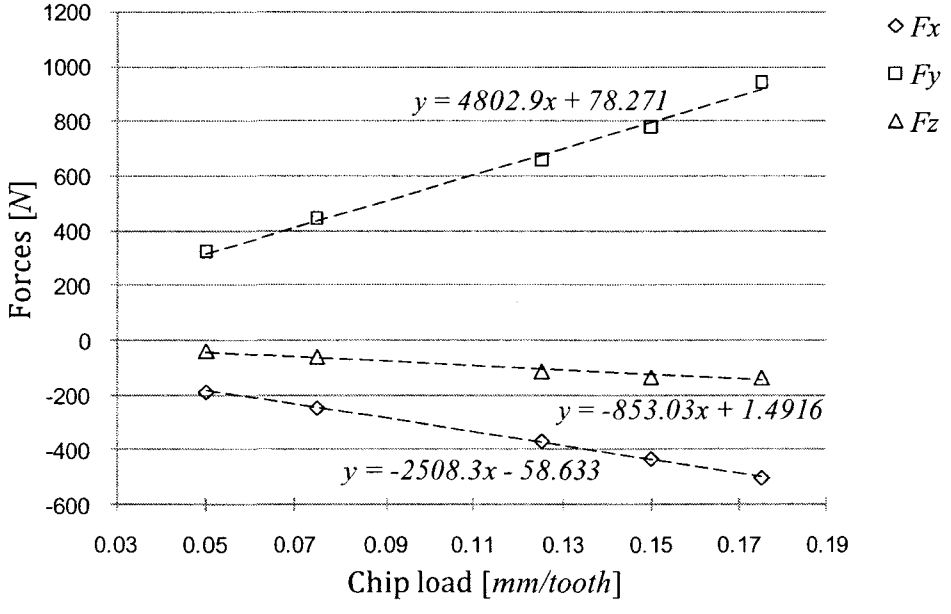


Figure 3.14 Linear regressions of average cutting forces

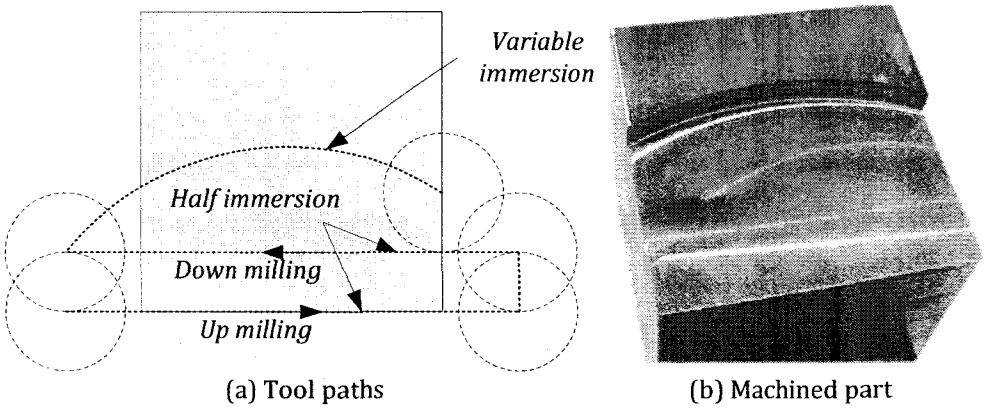
Table 3.1 Cutting forces coefficients of the experimental cutter machining Aluminum alloy Al7075 with cutting fluid

K_{tc}	960.580 N/mm^2	K_{te}	12.295 N/mm
K_{rc}	401.660 N/mm^2	K_{re}	9.21 N/mm
K_{ac}	-133.994 N/mm^2	K_{ae}	0.149 N/mm

3.5.3 Comparing Experimental and Simulation Results

Three experiments are conducted using the above setup. The developed simulation system is used to predict cutting forces for each experiment, and simulation results are compared to the measured cutting forces.

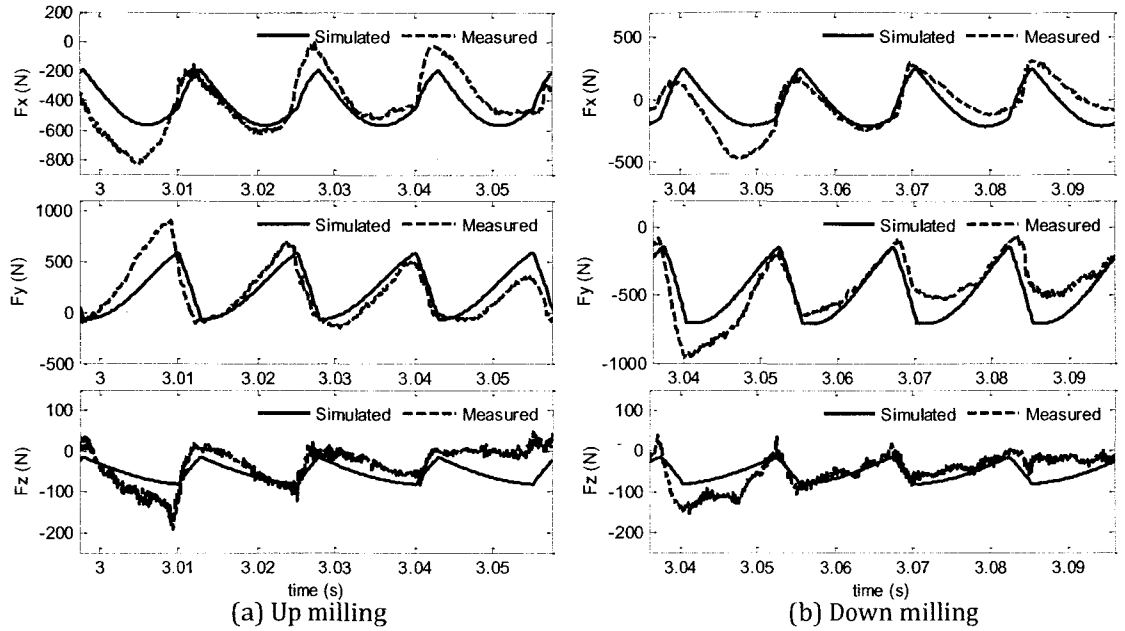
Experiment 1 - End milling with constant depth of cut



(a) Tool paths (b) Machined part

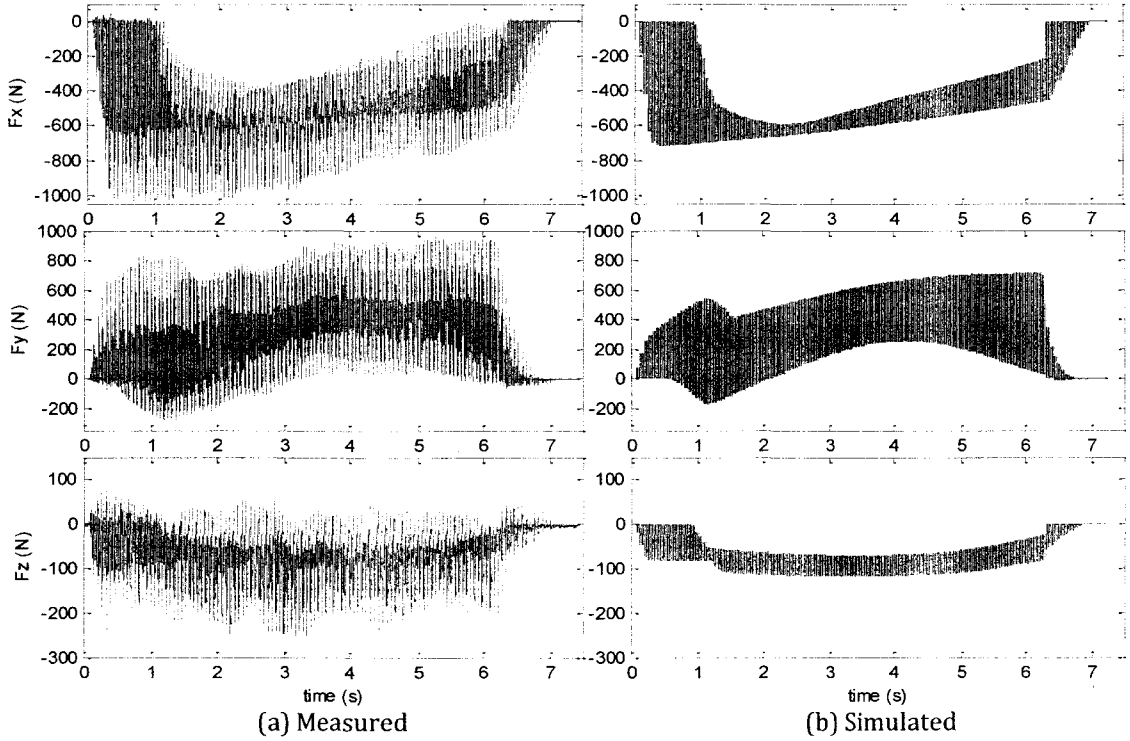
Figure 3.15 Experiment 1 - Constant depth of cut

Three tool path segments are performed at *5mm* axis depth of cut, and the feed rate is *500 mm/min*. During the first two paths, half immersion up/down milling, the CWE geometry remains constant; and simulated and measured cutting forces of one cutter revelation are plotted in Figure 3.16. The cutting forces along the circular arc tool path are shown in Figure 3.17, and two instantaneous sample locations are shown in Figure 3.18.



(a) Up milling (b) Down milling

Figure 3.16 Experiment 1 - Cutting forces of half immersion up/down milling



(a) Measured (b) Simulated

Figure 3.17 Experiment 1 - Cutting forces of the circular arc tool path

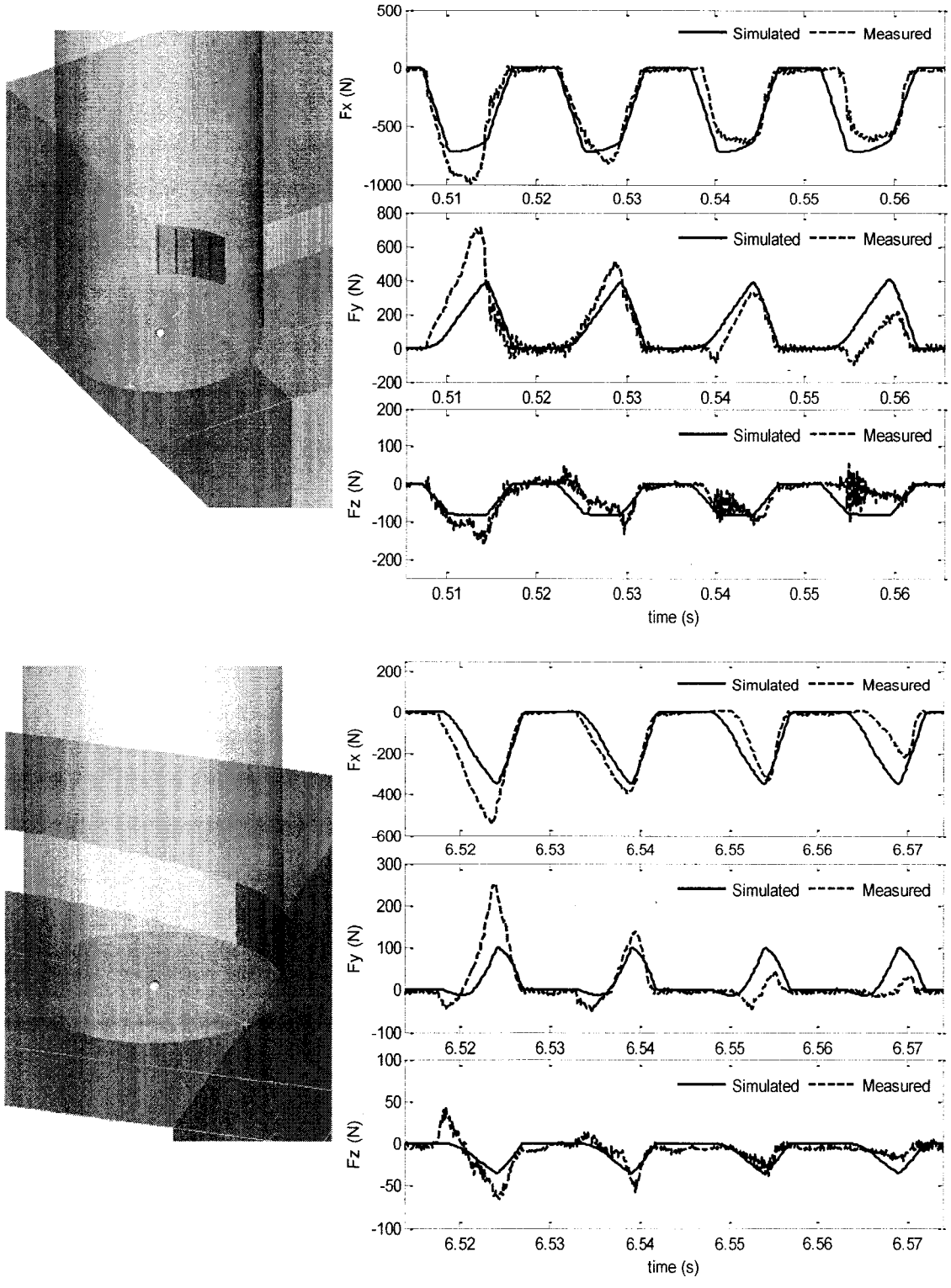
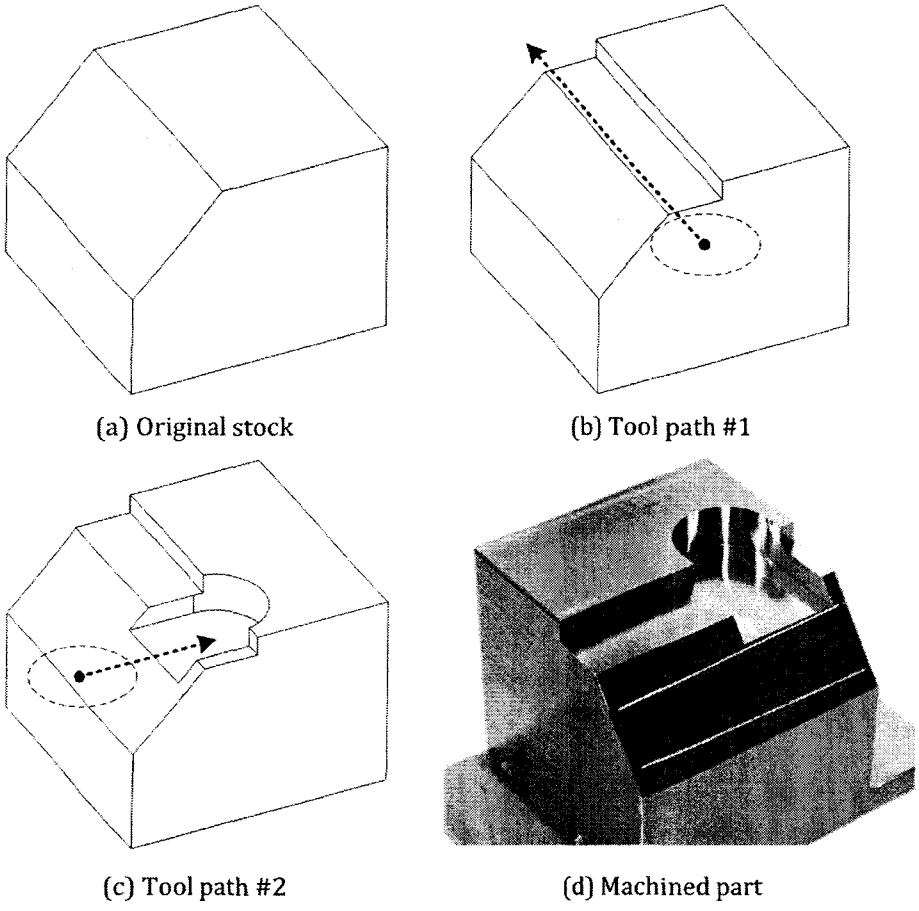


Figure 3.18 Experiment 1 - Instantaneous CWE geometry and cutting forces

Experiment 2 - End milling with variable depth of cut



(a) Original stock (b) Tool path #1

(c) Tool path #2 (d) Machined part

Figure 3.19 Experiment 2 - Variable depth of cut

A non-prismatic stock is machined by two tool paths at different depth of cut (*4mm* and *8mm*), and the feed rate is *500mm/min*. The engagement area of the first path is a non-orthogonal shape, see Figure 3.20. The engagement of the second path changes along the tool path; the cutting forces are plotted in Figure 3.21. The instantaneous CWE geometry and cutting forces of two sample locations on the second path are shown in Figure 3.22.

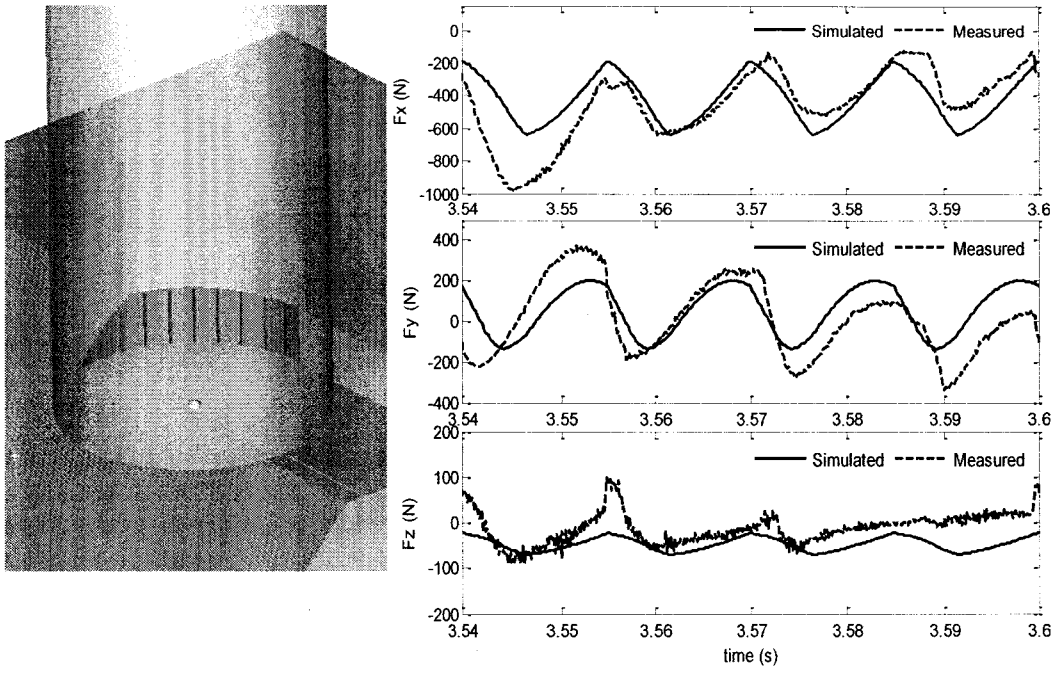
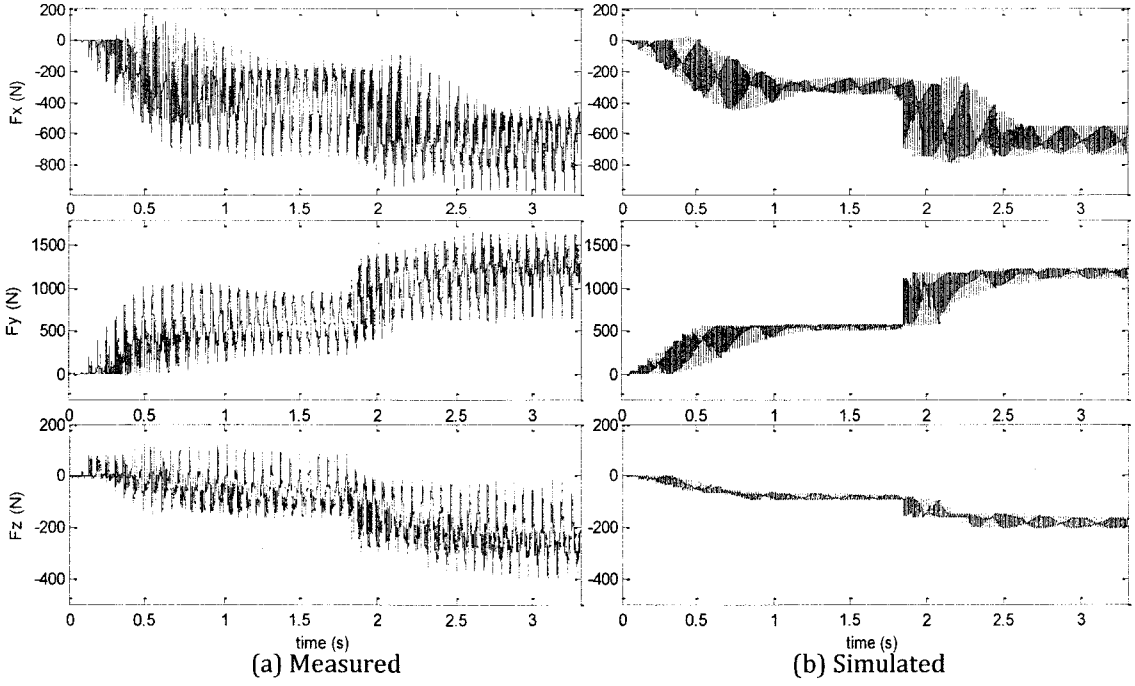


Figure 3.20 Experiment 2 - CWE geometry and cutting forces of path#1



(a) Measured (b) Simulated

Figure 3.21 Experiment 2 - Cutting forces of path#2

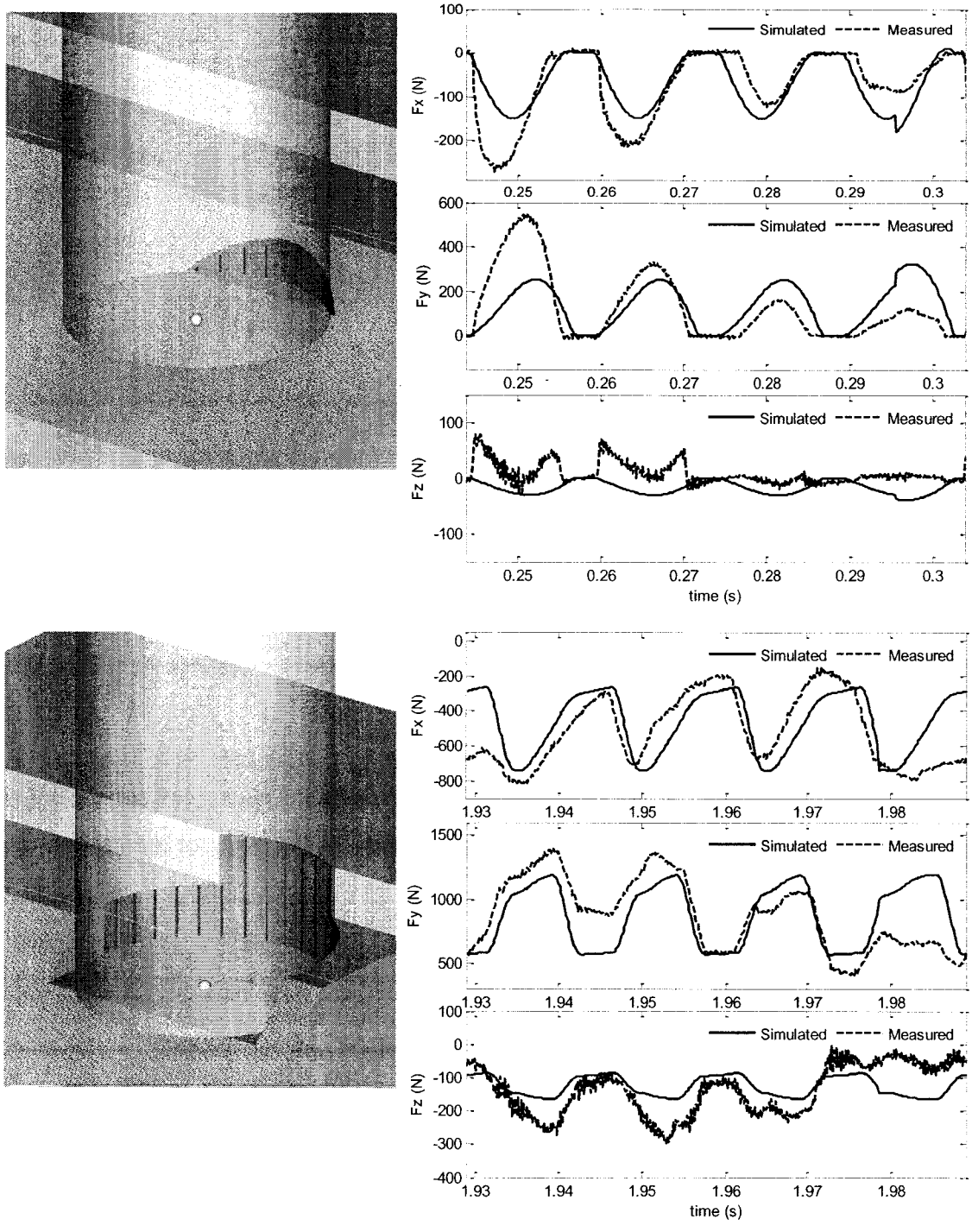
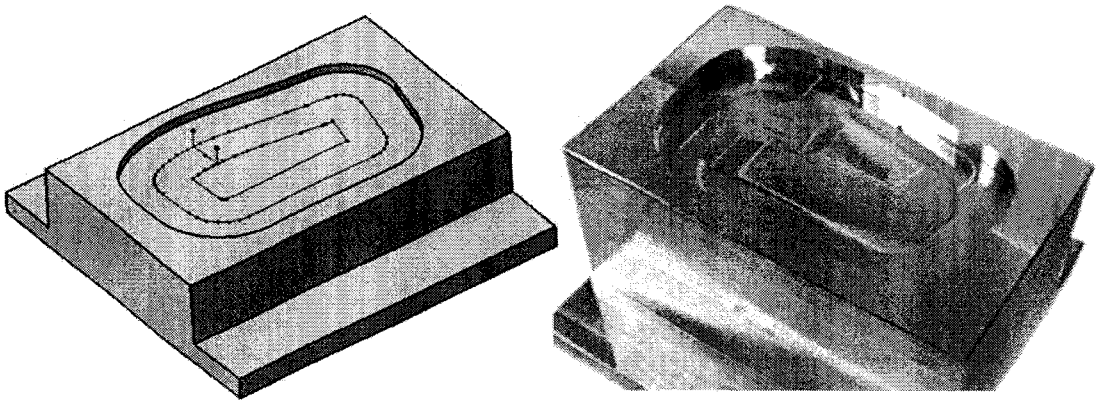


Figure 3.22 Experiment 2 - Instantaneous CWE geometry and cutting forces

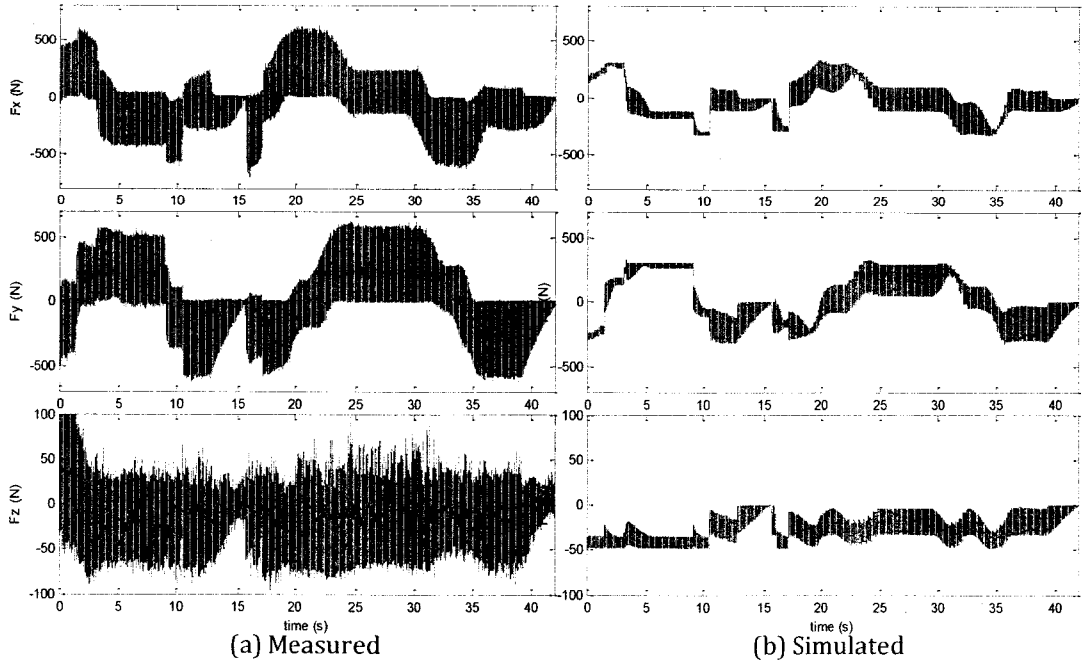
Experiment 3 - Free-form contour pocketing



(a) Tool paths generated by CATIA (b) Machined pocket

Figure 3.23 Experiment 3 - Pocketing

A pocket of free-form curve contour is machined. The tool paths are generated using CATIA V5. The axial depth of cut is *2.54mm,* and the feed rate is *400mm/min.*



(a) Measured (b) Simulated

Figure 3.24 Experiment 3 - Cutting forces

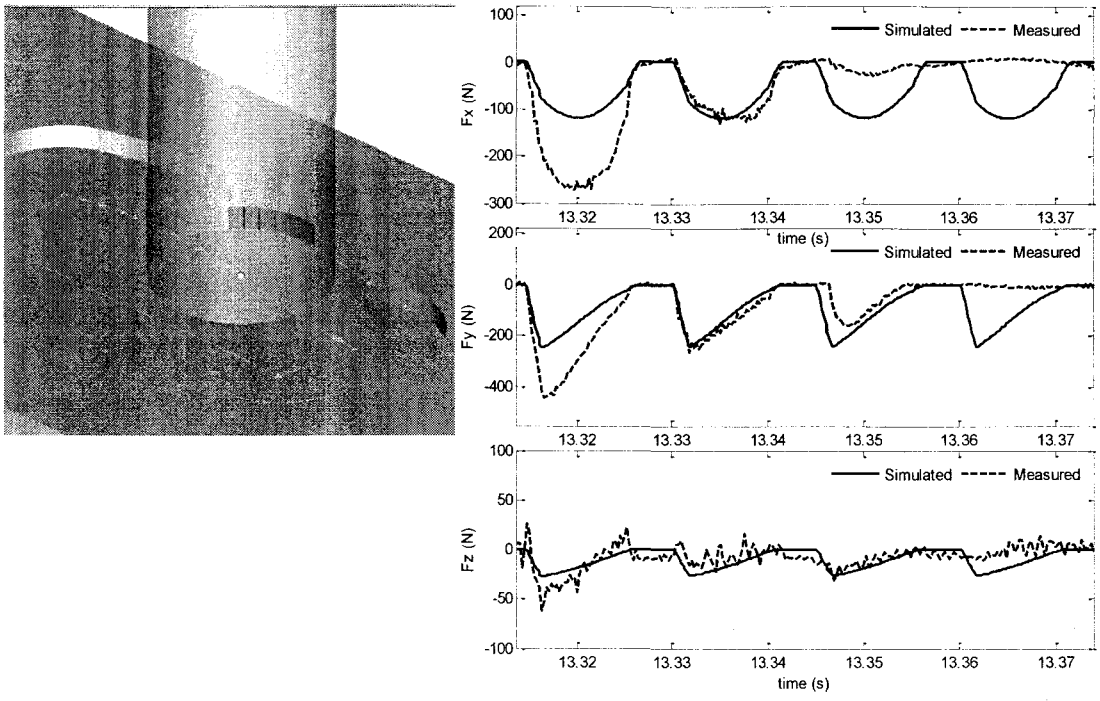
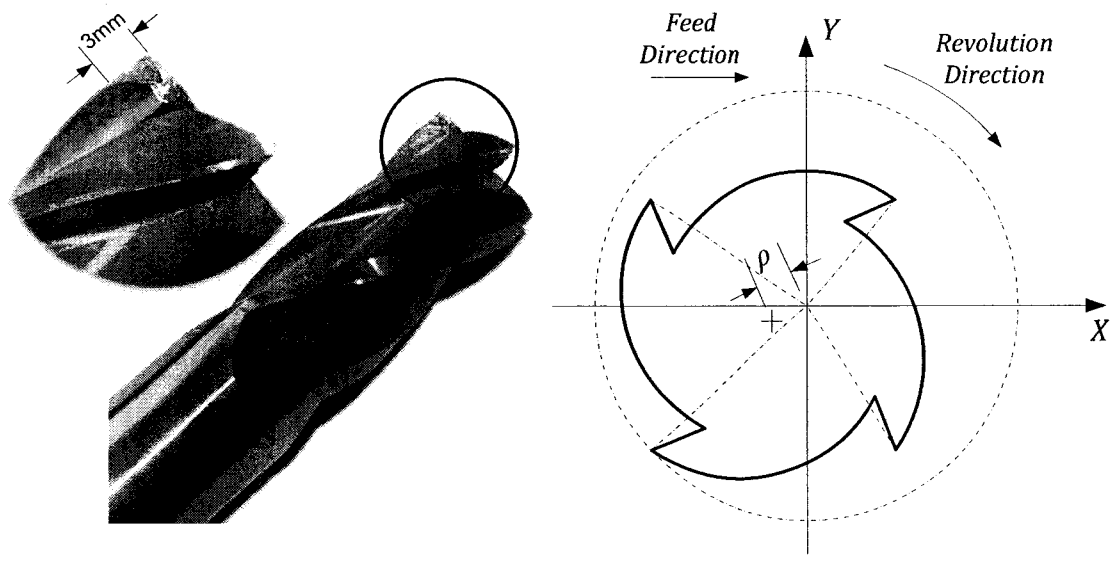


Figure 3.25 Experiment 3 - Instantaneous CWE geometry and cutting forces

From the above figures, we can see that the simulated cutting forces are in good agreement with the experiments. Since the simulation considers no tool wear and cutter runout (eccentricity of the tool rotation), the peak forces encountered by each flute are identical in the simulation results. While in the experiments, these setup errors are unavoidable, which lead to uneven distribution of chip load and cutting forces on each flute in one revolution. Since the same cutter has been used for all experiments in this report, similar phenomenon occurs in all experiments. Especially for the pocketing experiment, see Figure 3.25, there is almost no chip load on one tooth. This is due to the small depth of cut (*2.54mm*) compared to the tool wear on one flute, a crack of length close to *3mm*, see Figure 3.26(a).



(a) Tool wear on the experimental cutter (b) Cutter runout

Figure 3.26 Experimental setup errors

For different instantaneous cutter location, the average resultant force during one revolution of the simulation results are within 10% of the experimental results.

3.6 Summary

In this chapter a geometric modeling methodology for 2½-axis milling with flat end cutter is presented. The in-process workpiece is represented using a Z-layer B-Rep model, and both rectangular and non-prismatic stock shapes can be handled properly. The CWE geometry is extracted by calculating the intersection between the cutter frontiers swept profile and the in-process workpiece layer profile, and complex engagement situations, such as multi-engagements, can be handled by arranging intersection points in pairs and updating the workpiece profile respectively. Then, the extracted CWE geometry is integrated with a mechanistic

force model for predicting cutting forces, torque and power. The accuracy of the developed simulation system is demonstrated through experimental verifications. Several cutting tests, including constant depth of cut, variable depth of cut and general pocketing, have been conducted, and the simulated cutting forces are in good agreement with the experimental results. Applications of the developed system are presented next in Chapter 4.

Chapter 4 Applications of 2½-Axis Milling Process Modeling

The 2½-axis milling process modeling system developed in previous chapter can provide useful information for understanding the effect of different cutting parameters, and therefore optimal cutting parameters can be selected. Two applications, including steady cut optimization and automatic feed rate selection for pocket machining, are presented in this chapter.

4.1 Optimal Cutting Parameters Selection for Steady Cut

In the roughing operations, especially for aerospace structural parts, a lot material needs to be removed layer by layer. At each layer, during most time the cutting is steady with constant MRR. To determine the optimal feed rate, the radial and axial depth of cut in order to achieve the maximal MRR within the machine tool's capability is the main concern in industry. The MRR (mm^3/s) of steady cut can be formulated as

$$MRR = W \cdot H \cdot f / 60 \tag{4.1}$$

where W (mm) is the radial depth of cut, H (mm) is the axial depth of cut, and f (mm/min) is the feed rate, as show in Figure 4.1. In order to achieve high machining

efficient, the MRR need to be maximized, while the cutting power draw from the spindle cannot exceed the machine tool's limitation. For this cutting parameters optimization problem, the object function is defined as Eq.(4.1), and the constraint is the spindle power. Other constraints, such as tool life and dynamic properties of machine structure (chatter vibrations), are not discussed in this research.

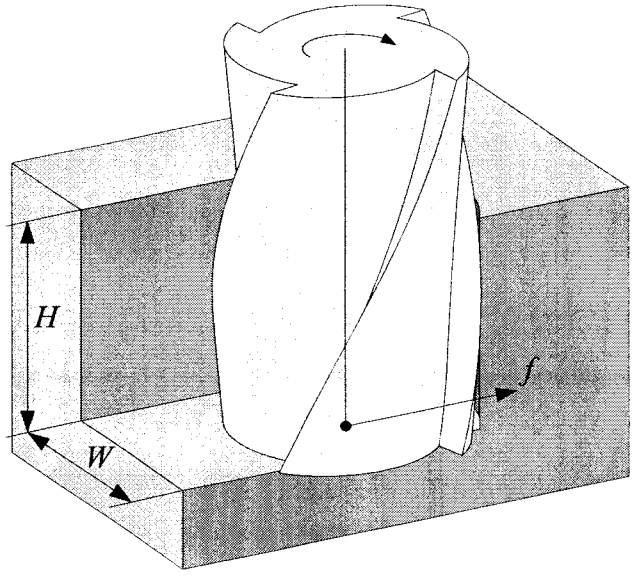


Figure 4.1 Steady cut with constant MRR

4.1.1 Optimal Cutting Parameter Selection Method

The optimization problem involves three variables, which are the feed rate f, axial depth H and radial depth W of cut. The value of these variables should be determined such that the MRR is maximized while the spindle power is equal to the given maximum value. To solve this problem, the methodology proposed here involves two steps.

The first step is to find the maximum MRR and corresponding axial depth and radial depth of cut for a given feed rate f. For a constant feed rate, the Eq.(4.1) is

a monotonically increasing function of W and H, whose maximum value is always on the boundary of the function domain. This boundary is determined by the spindle power constraint, which can be formulated as

$$f_{\text{power}}(W, H) = P_{\max} \tag{4.2}$$

where f_{power} is the max spindle power expressed as a nonlinear function of W and H, and $P_{\max}$ is the given max spindle power limitation. By using the developed simulation program to evaluate the nonlinear function f_{power}, the Eq. (4.2) can be solved numerically. So the optimal W and H, which produce the maximum MRR, can be found on the boundary through Eq.(4.1).

The second step is to change the feed rate f, so the maximum MRR changes accordingly. At an optimal feed rate $f_{\max}$, the MRR reaches the maximum value. Thus, the $f_{\max}$ and corresponding axial depth and radial depth are the solution to the cutting parameter optimization.

4.1.2 Illustrative Example

An illustrative example is presented in this section to explain the process of the proposed method. A solid carbide flat-end mill cutter with *19.05mm* diameter, *4* flutes, *30 degree* helix angle and *38.1mm* flute length is used to cut aluminum alloy Al7075 workpiece material. The cutting coefficients with cutting fluid on are identified in Table 3.1. The spindle speed is fixed at *3000RPM*, while the feed rate f, axial depth H and radial depth W of cut are the variables. The instantaneous cutting forces and spindle power in one revolution for a set of input cutting parameters,

W=13mm, *H=5mm*, and *f=1000mm/min*, are shown in Figure 4.2. The maximum instantaneous spindle power can be found in the spindle power curve.

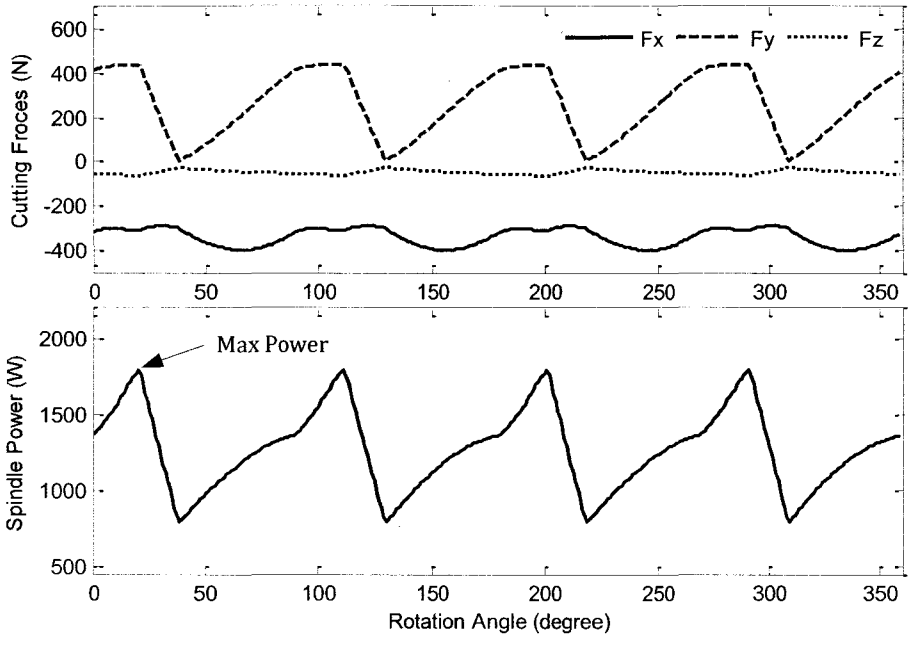


Figure 4.2 Cutting forces and spindle power
(*W=13mm, H=5mm, f=1000mm/min*)

By changing axial and radial depth of cut within the domain bounded by the geometric constraints, cutter diameter and cutting flutes length, the maximum spindle power can be calculated for each pair of inputs; therefore a graph representing the maximum spindle power using axial and radial depth of cut can be obtained. Figure 4.3 shows a maximum spindle power graph at the feed rate *1000mm/min.*

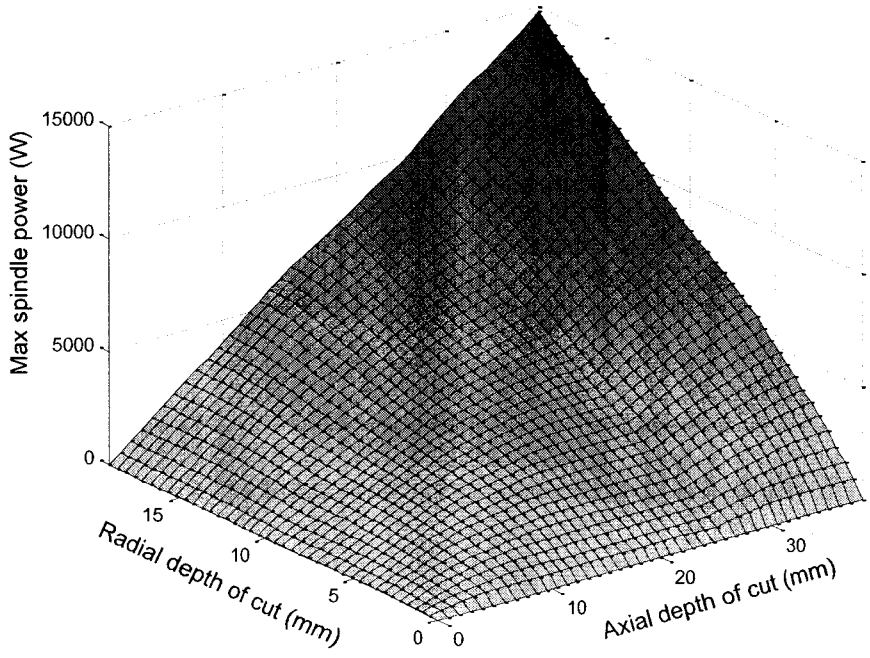
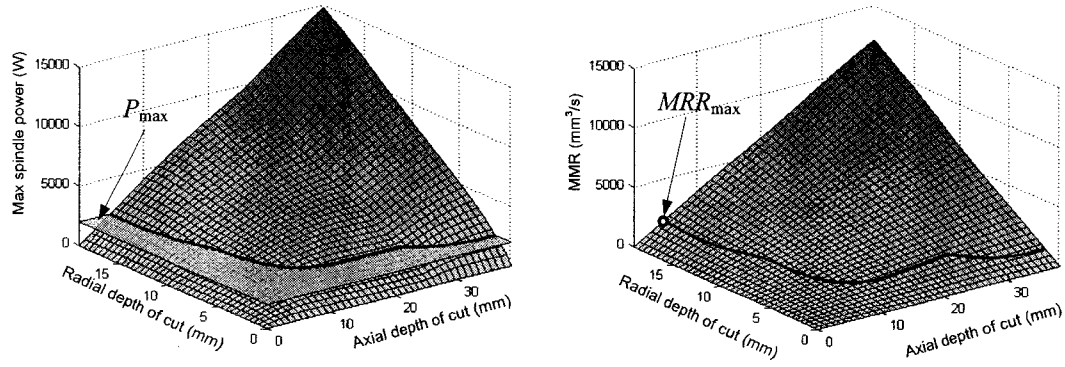
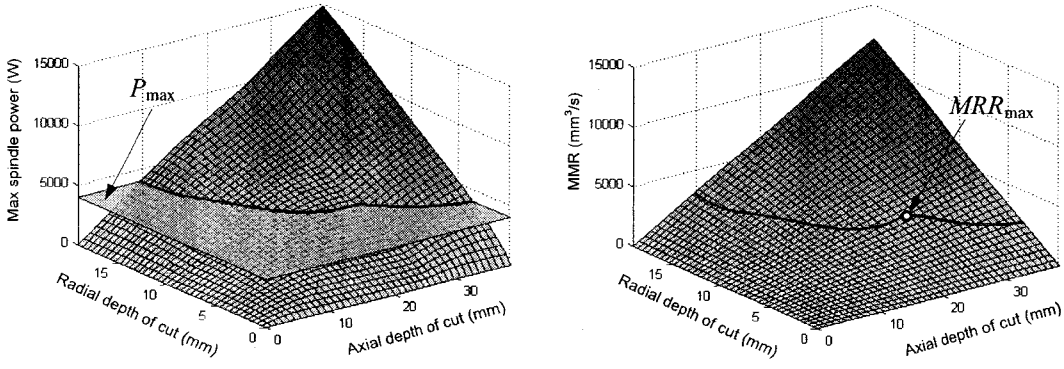


Figure 4.3 Maximum spindle power graph at feed rate *1000mm/min*

The Eq. (4.2) can be solved numerically by intersecting the maximum spindle power graph with a horizontal plane at the height P_{max}. The intersection curve represents the boundary of spindle power constraint. By mapping the boundary to Eq.(4.1), the maximum MRR and corresponding axial depth and radial depth of cut can be found. Two examples are shown in Figure 4.4: (a) set $P_{max}=2000W$, the maximum MRR ($1542.4\ mm^3/s$) can be achieved at full immersion (radial depth equal to cutter diameter) and axial depth $7.8\ mm$; (b) set $P_{max}=4000W$, the maximum MRR ($3271.8\ mm^3/s$) can be achieved at radial depth $7.63\ mm$ and axial depth $25.7\ mm$.



(a) P_{max}=*2000W: MRR=1542.4mm³/s, H=7.8mm, W=19.05mm*



(a) P_{max}=*4000W: MRR=3271.8mm³/s, H=25.7mm, W=7.63mm*

Figure 4.4 Finding the maximum MRR at feed rate *f=1000mm/min*

From the machining handbooks, ranges of recommended feed rates for different workpiece materials can be obtained. In this example, the range of feed rate f is set to *100~2500mm/min*. Assume the maximum spindle power for the machine is *5000W*. By setting P_{max}=*5000W* and following the procedure described above, the maximum MRR at each feed rate can be calculated. The relation of feed rate and corresponding maximum MMR is shown in Figure 4.5. In this case, the highest MRR is achieved at the highest feed rate, 2500mm/min.

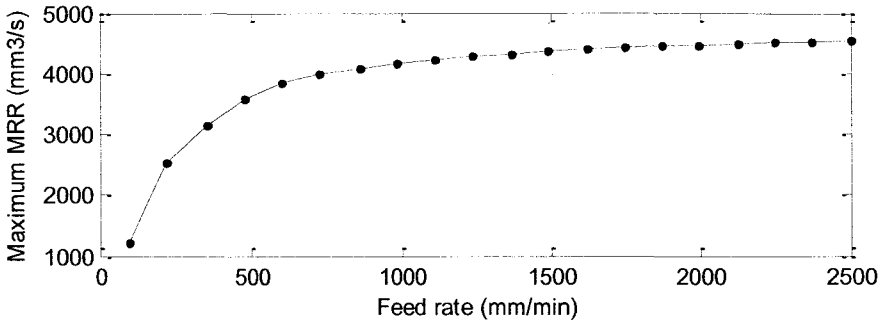


Figure 4.5 Relations of feed rate and maximum MRR (P_{max}=*5000W*)

The relation diagram also shows that the increasing rate for the maximum MRR is smaller at higher feed rate. From *1000* to *2500 mm/min*, the increase maximum MRR is less than 9%. This result can served as guidance for selection optimal cutting parameters. By combining with other constraints, such as the dynamic properties of the machine tool, smaller feed rate, such as *1000mm/min* may be selected over the higher values, since the increase of efficiency is not big.

4.2 Automatic Feedrate Selection for Pocket Machining

Pocketing is one of the most common operations in NC machining applications. For example, in aerospace industry, most structure parts are produced with pocket machining with the support structure leaved between pockets. Generally, a pocket is described as a closed profile in CAD software, sometimes with islands. Two major approaches, contour parallel and direction parallel, are used in commercial CAM software to generate pocketing tool paths. The CAM software only generates tool paths based on the pocket geometry; while the feed rate is still specified as one constant value for all path segments, and is heavily depended on the user's experience. Although the axial depth of cut remains constant for pocket

machining, the radial depth of cut still changes along the tool paths due to the changing in-process workpiece geometry; therefore the cutting load changes accordingly. At certain locations cutting load may be too small and the machine tool is not utilized at its full capacity; while at some locations cutting load may be too big and may exceed the machine tool limit. Thus, for more efficient and safe pocket machining, it is required to regulate the cutting load while machining.

The cutting load regulation can be achieved by adjusting the feed rate of each tool path segment, so the spindle power can be maintained at a specified value. The developed 2½-axis milling process modeling system can provide useful information of the pocket machining process, including instantaneous cutter engagement, cutting forces, and maximum spindle power consumed. An automatic feed rate adjustment method based on the process simulation result is presented in this section.

4.2.1 Optimal Feed Rate Selection Method

For the pocket machining where the axial depth of cut is constant, the radial depth of cut and feed rate are considered as the variables. Under the assumption that no complex multi-engagement occurs in the pocket machining, a simplified chip load model is described in Figure 4.6, where α is the cutter engagement angle and W is the radial depth of cut.

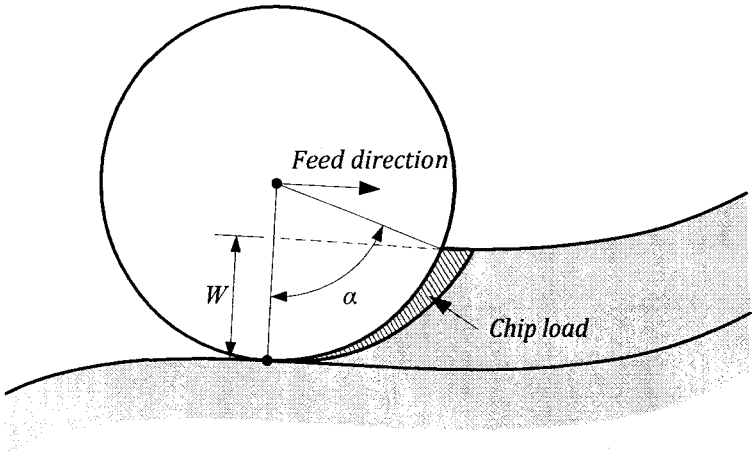


Figure 4.6 Chip load model of pocketing

The max spindle power can be considered as a nonlinear function of the feed rate f and the engagement angle α. For certain feed rate and engagement angle, the max spindle power can be calculated using the simulation program by considering only one revolution. The calculation is similar to the steady cut with the same radial depth of cut W, as shown in Figure 4.2. By changing both radial depth of cut and feed rate, the maximum spindle power can be calculated for each pair of inputs; therefore a graph representing the maximum spindle power using feed rate and engagement angle can be obtained.

For example, a solid carbide flat-end mill cutter with *19.05mm* diameter, *4* flutes and *30 degree* helix angle is used to cut aluminum alloy Al7075 workpiece material. The cutting coefficients with cutting fluid on are identified in Table 3.1. The spindle speed is fixed at *3000RPM*, and the axial depth of the pocket is *10mm*. By changing the feed rate in the range *600~2400 mm/min*, the max spindle power graph is obtained in Figure 4.7.

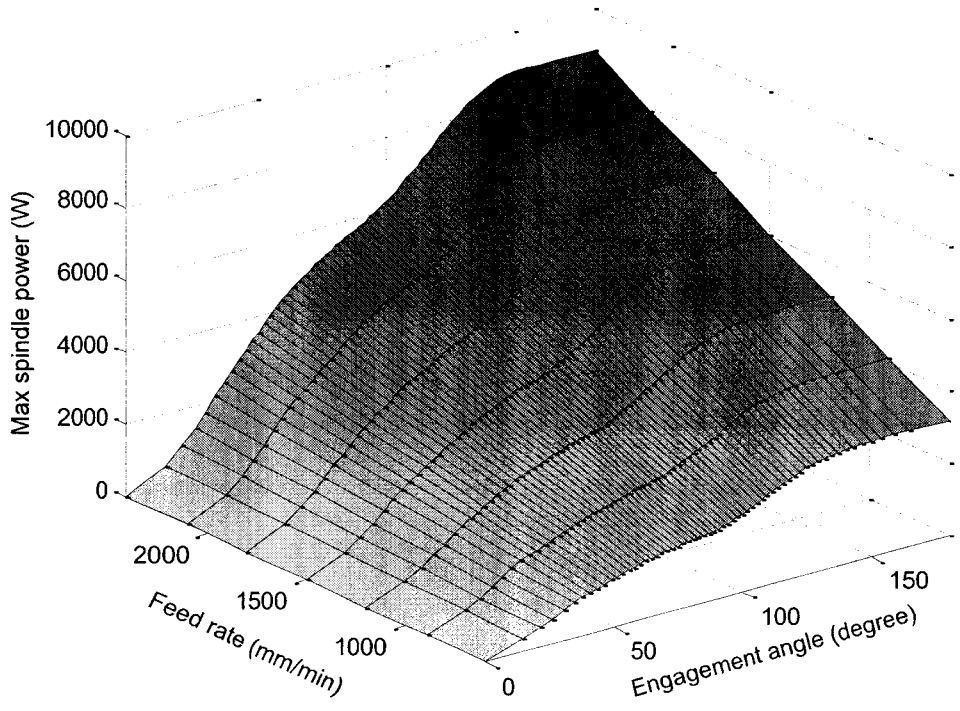


Figure 4.7 Maximum spindle power graph at *10mm* depth of cut

For a specified maximum spindle power limit P_{max}, a horizontal plane is constructed at the height P_{max} and intersects with the spindle power graph. The intersection curve is the iso-spindle power curve, which represents the engagement angle and corresponding feed rate when spindle power is equal to P_{max}. The iso-spindle power curve is used for automatic feed rate selection. By given an engagement angle, the maximum feed rate can be found on the iso-spindle power curve. Shown in Figure 4.8 is the iso-spindle power curve for P_{max}=*5000W*. For engagement angle *100 degree*, the maximum feed rate can be selected is *1885mm/min*; for engagement angle *140 degree*, the maximum feed rate is *1199 mm/min*.

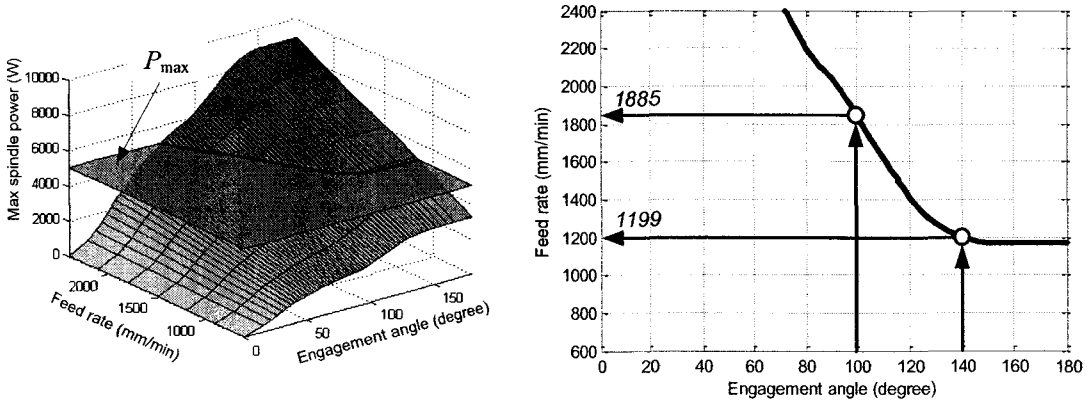


Figure 4.8 Iso-spindle power curve for automatic feed rate selection

4.2.2 Tool Paths Optimization Procedure

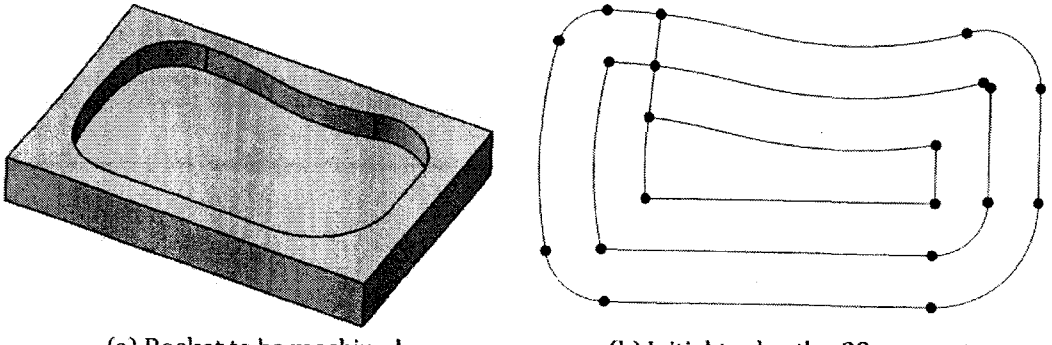
Given a pocket and the cutter size, the initial tool paths can be generated using existing CAM software. For a user specified maximum spindle power, the tool paths optimization process involves the following steps.

1) Construct the iso-spindle power curve for the given pocket depth, cutter size, and maximum spindle power.
2) One feed rate value is assigned to the initial tool paths, and the developed 2½-axis milling simulation system is used to predict cutting forces and power.
3) Scan on the simulation results. If dramatic changes of chip load are detected along a tool path segment, the tool path segments can be further split into two or more shorter segments. By considering the kinematic properties of the machine tool, short segments are avoided in the tool path splitting process.
4) Determine the feed rate of each tool path segment based on the peak engagement angle along the path. The iso-spindle power curve is used to calculate optimal feed rate for each segment.

4.2.3 Illustrative Example

To demonstrate the effectiveness of the proposed method, a pocket machining example is presented in this section. The example uses the same cutter (*19.05mm* diameter, *4* flutes and *30 degree* helix angle) and axial depth of cut (*10mm*) mentioned in previous section. The pocket to be machined is shown in Figure 4.9, and the initial tool paths consist *22* segments including linear, circular and NURBS tool path segments. By setting the max spindle power P_{max}=*5000W*, the iso-spindle curve can be obtained, see Figure 4.8.

A constant feed rate *1500 mm/min* is assigned to all segments, the developed 2½-axis modeling system is used to simulate the pocketing process. The simulation results are shown in Figure 4.10, and the total machining time is *26.01s*. The results show that the max spindle power is beyond P_{max}(*5000W*) which is not acceptable; while most parts of the tool paths are below *4000W*, which is not efficient.



(a) Pocket to be machined (b) Initial tool paths, 22 segments

Figure 4.9 Pocket and initial tool paths

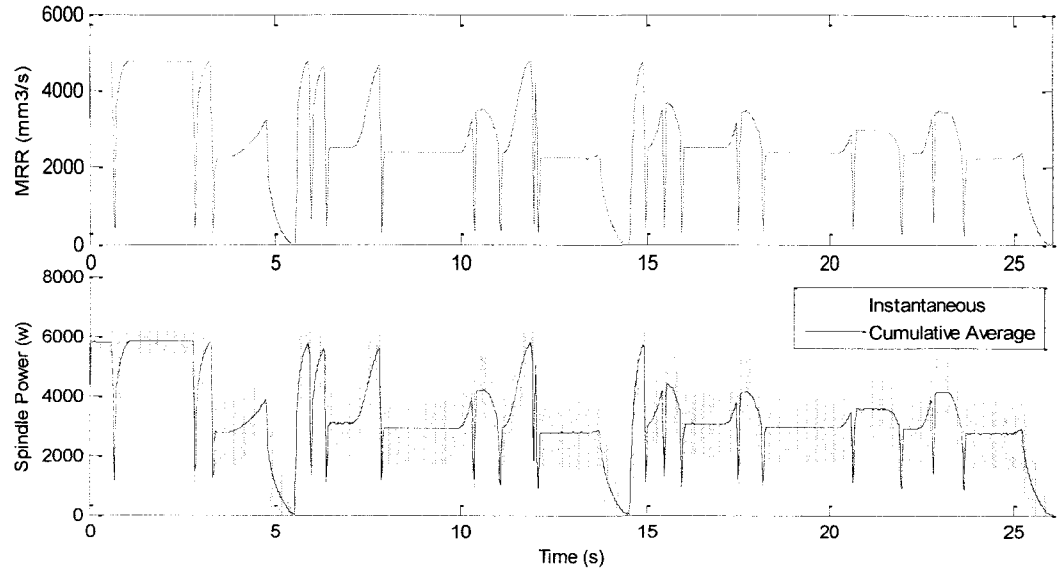


Figure 4.10 Simulation results at feed rate *1500mm/min*

By scanning the maximum spindle power, six splitting locations are determined according to the change of chip load, and corresponding splitting points on the tool paths are calculate (see Figure 4.11 and Figure 4.12).

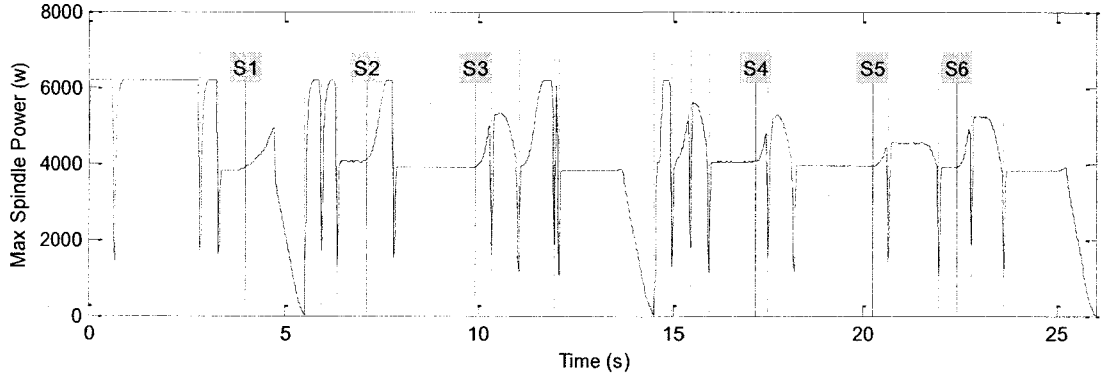


Figure 4.11 Splitting tool paths according to chip load

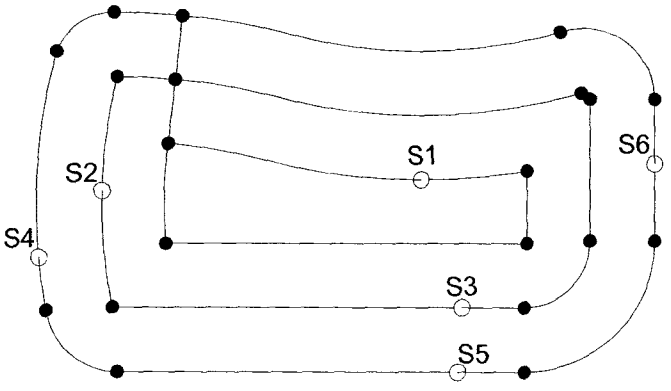


Figure 4.12 Tool paths after splitting, 28 segments

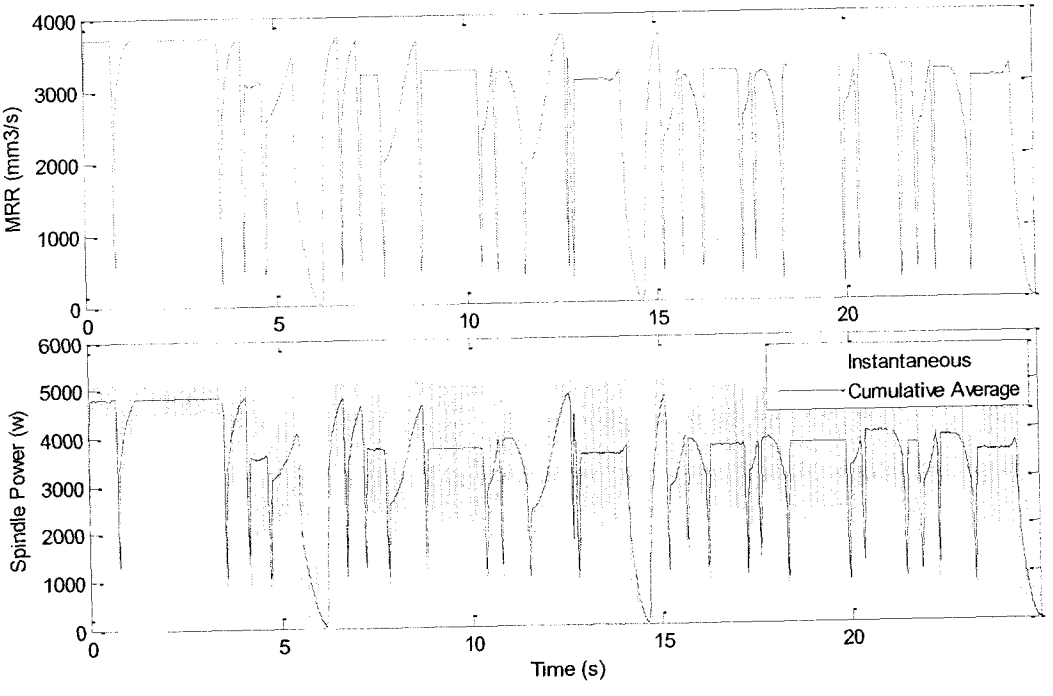


Figure 4.13 Simulation results of the optimal tool paths

The 2½-axis milling process simulation system records the instantaneous chip geometry along the tool paths. By collecting the peak engagement angle for each segment, the optimal feed rate can be obtained from the iso-spindle power curve (Figure 4.8). The simulation results of the adjusted tool paths are shown in Figure 4.13. The total machining time is reduced to *24.97s*, and the maximum spindle power is kept around *5000W*.

4.3 Summary

This chapter presents applications of the 2½-axis milling process modeling on optimal machining parameters selection. Effective and practical algorithms are proposed for steady cut optimization and automatic feed rate selection for pocket machining.

The proposed graph based method is a general framework that can be extended to other applications. Both applications presented in this chapter use the maximum spindle power as the constraint for cutting parameters optimization. For application considering the tool deflection, where the primary concern is the resultant blending force, the same methodologies can still be applied. By replacing the maximum spindle power constraint with maximum resultant blending force, the developed methodologies can be used to optimize cutting parameters in order to maintain certain cutting force.

Chapter 5 Process Modeling for 3-Axis Milling of Sculptured Surfaces

5.1 Introduction

Sculptured surfaces are widely used in mechanical products, especially in the die and mold, aerospace, automotive and appliances industries. Thanks to the advanced CAD technology, complex sculptured surfaces can be easily designed by using commercial CAD packages. Generally, the free-form surface is defined mathematically as a collection of interconnected and bounded parametric patches. Several types of parametric surfaces, including Bi-cubic, Bezier, Non-Uniform Rational B-Spline (NURBS), and Coons surfaces [88], are used in CAD software. Multi-axis CNC milling machines, including 3- and 5-axis machines, are used to produce these sculptured surface parts.

To machine a sculptured surface in 3-axis finish milling, theoretical tool path curves are discretized into a number of cutter location (CL) points. During machining, the cutter cuts through the stock material from one CL point to the next, sweeping a 3D envelope surface. The stock material contained in the cutter swept envelop is removed and a furrow patch is formed on the machined surface, see Figure 5.1.

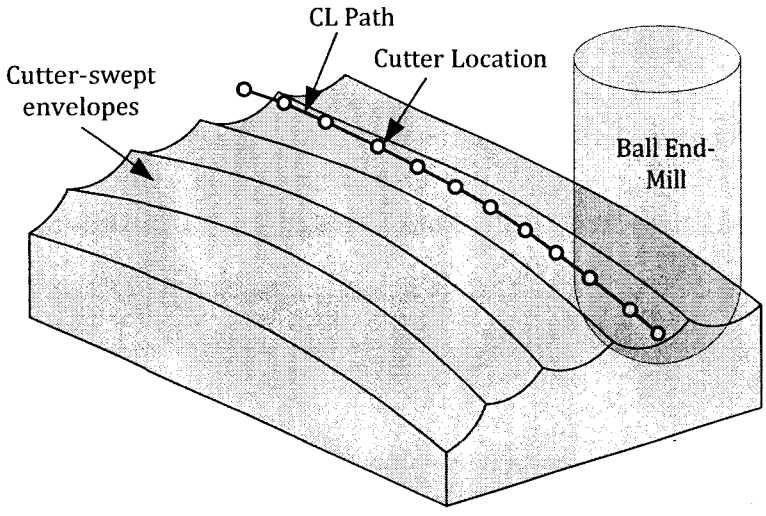


Figure 5.1 3-axis milling of sculptured surface

Due to the complexity of sculptured surfaces, simple geometries of the standard cutting tools and limited types of tool motions provided by CNC machine tools, accurately machining these surfaces with less costs and shorter time has become a challenging job.

In this chapter, we do not adopt the conventional differential equation of cutter swept volumes; instead, we study the basic mechanism of removing stock material in three-axis milling operation. The fundamental concept is that the profile at a layer of workpiece stock is formed by the cutting tool pass through this layer along a predetermined tool path. First, the geometric model of the cutter swept envelope, which is a simple 2-D, not a 3-D swept profile, is presented, and the mathematical formula of the profile is derived for APT cutter geometry and different tool motions. Second, a Z-level B-Rep model is used to represent the in-process workpiece model, and an innovative geometric approach is used to extract the CWE geometry. Then, an existing cutting forces model is adopted to predict the cutting

forces. The implementation is a process simulation system for 3-axis surface milling. Finally, the developed system is experimentally verified by comparing the simulation results with actual forces data collected from machining a test part.

5.2 Geometric Model of Cutter Swept Profile in 3-Axis Milling

The general geometry of end mill is defined as APT tool [89], and the standard cutters such as flat, bull-nose, ball, and taper end mills are special cases (see Figure 5.2). In order to find the generic model of the cutter swept profile, the APT tool is adopted in our research.

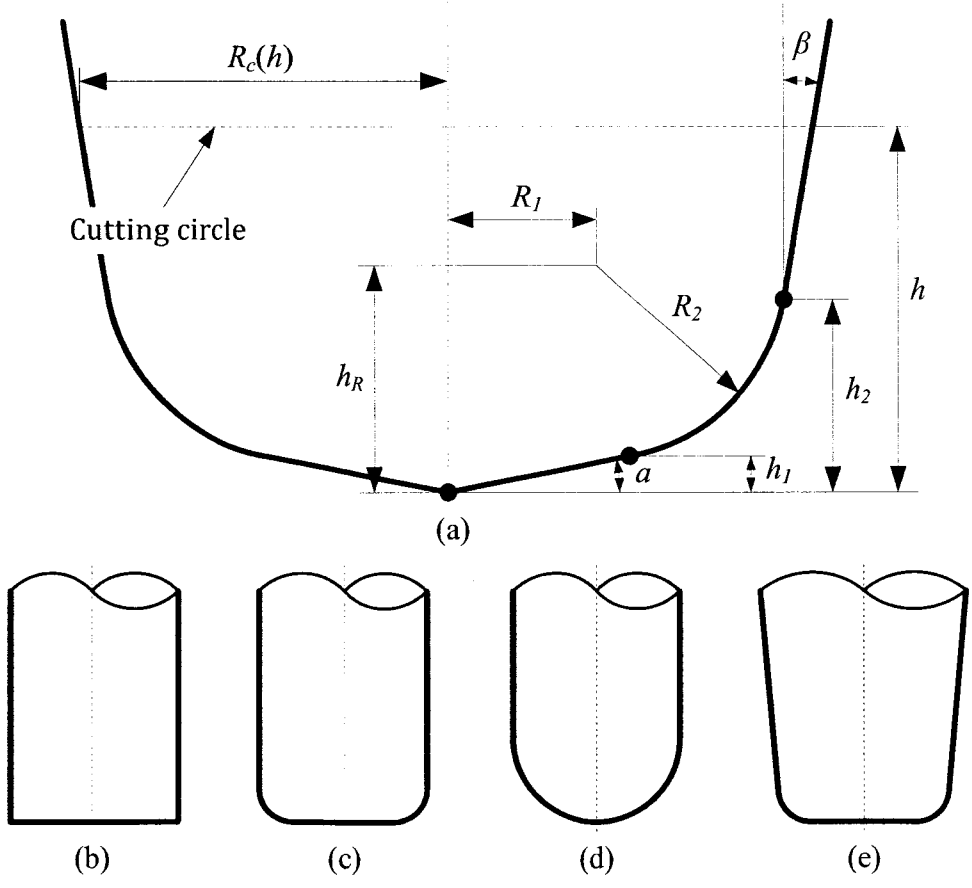


Figure 5.2 (a) APT tool, (b) flat end mill ($\alpha=0$, $\beta=0$, $R_2=0$), (c) bull-nose end mill ($\alpha=0$, $\beta=0$), (d) ball end mill ($\alpha=0$, $\beta=0$, $R_1=0$), (e) taped end mill ($\alpha=0$)

During machining, the cutting tool rotates about its axis, forming a revolving surface called cutting surface. The geometry of a cutting tool can be regarded as a number of thin disks piled up along the tool axis, and all disks have the same number of cutting edges as the tool. Since the cutting speed is much larger than the feed rate during machining, the trochoidal trajectory of each cutting edge point can be approximated as circle. Thus, the cutter is regarded as many cutting circles along the tool axis during machining. Based on the parameters defined Figure 5.2(a), the radius of a cutting circle at axial height h can be represented as

$$R_c(h)=\begin{cases}\dfrac{h}{\tan\alpha} & (0\le h<h_1),\ \text{lower conic surface}\\ R_1+\sqrt{R_1^2-(h_R-h)^2} & (h_1\le h<h_2),\ \text{toroidal surface}\\ R_1+\dfrac{R_2+(h-h_R)\sin\beta}{\cos\beta} & (h_2\le h),\ \text{upper conic surface}\end{cases} \tag{5.1}$$

where $h_1=h_R-R_2\cdot\cos\alpha$, and $h_2=h_R-R_2\cdot\sin\beta$.

In 3-axis tool motion, all cutting circles move along the same tool feed direction to remove the stock material. Figure 5.3 shows an APT tool moving from location one to location two in a certain time period. To study the process of removing stock material, a horizontal layer of workpiece material, Π in Figure 5.3, is examined closely. At the beginning, the cutting circle 1 is aligned with the layer, and all the material at inside the cutting circle has already been removed. When the tool moves along the tool feed direction, the cutting circle 1 leaves the layer and a cutting circle beneath it reaches this layer and removes material inside it. As all the cutting circles reach the layer consecutively, the outside envelope of all these cutting circles

forms a 2D profile on this layer, and all the material inside the profile is removed. By intersecting the profile with the original shape of the part on this layer, both the shape of removed material and the machined shape on this layer after this tool motion can be obtained.

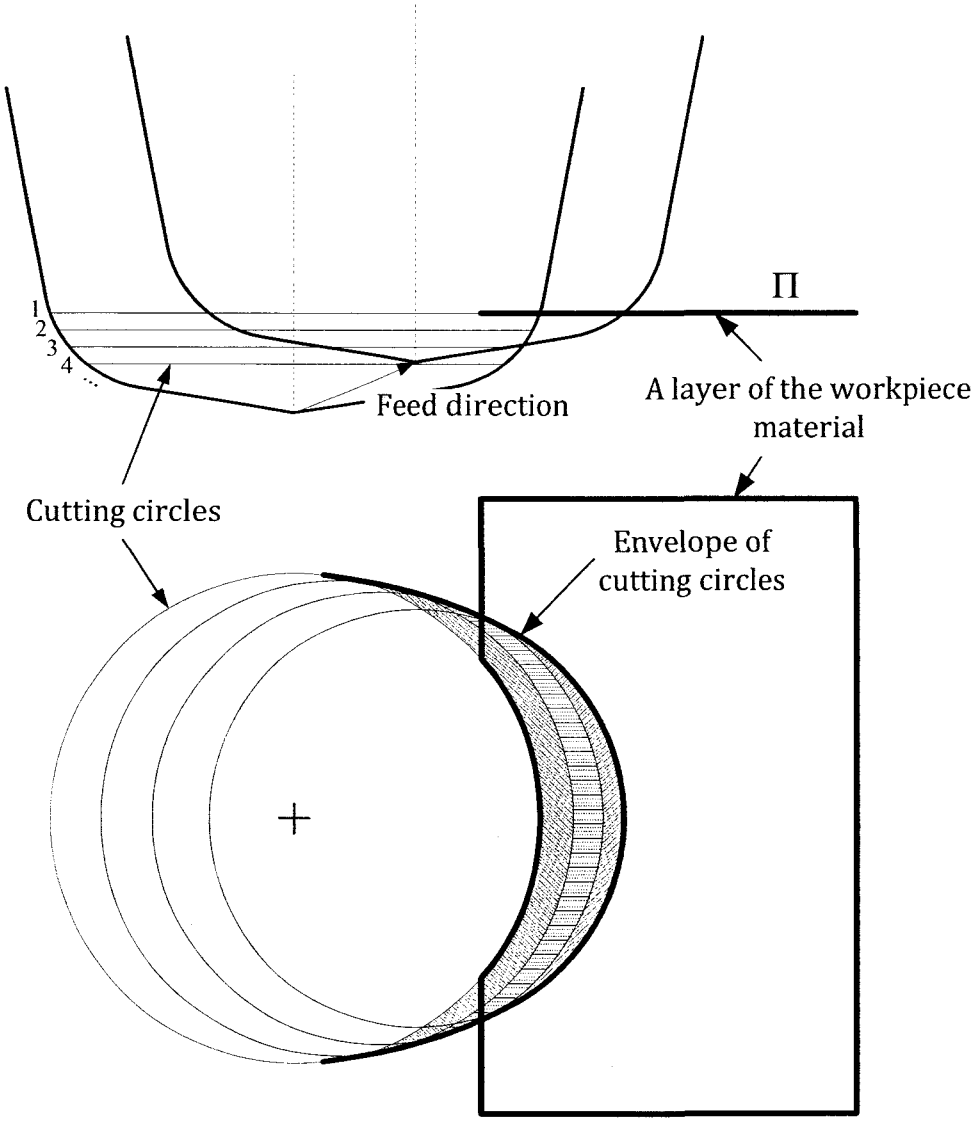


Figure 5.3 Envelope of cutting circles

Based on the mechanism of removing stock material on the horizontal layer, a close form equation of the three-axis cutter swept profile, the envelope of cutting

circles, can be derived. Suppose on a 3-axis vertical CNC milling machine, an APT tool moves from one location $CL_1 = [x_1, y_1, z_1]^T$ to another location $CL_2 = [x_2, y_2, z_2]^T$ in the part coordinate system (X_P, Y_P, Z_P). A local cutter coordinate system (X_C, Y_C, Z_C) is set up at CL_1. The X_C-axis is defined parallel to the vector $\overrightarrow{CL'_1\ CL'_2}$, which is the projection of $\overrightarrow{CL_1\ CL_2}$ on plane $X_P\ O_P\ Y_P$; the Z_C-axis is defined parallel to the Z_P-axis; and the Y_C-axis is defined by the right-hand rule, as shown in Figure 5.4.

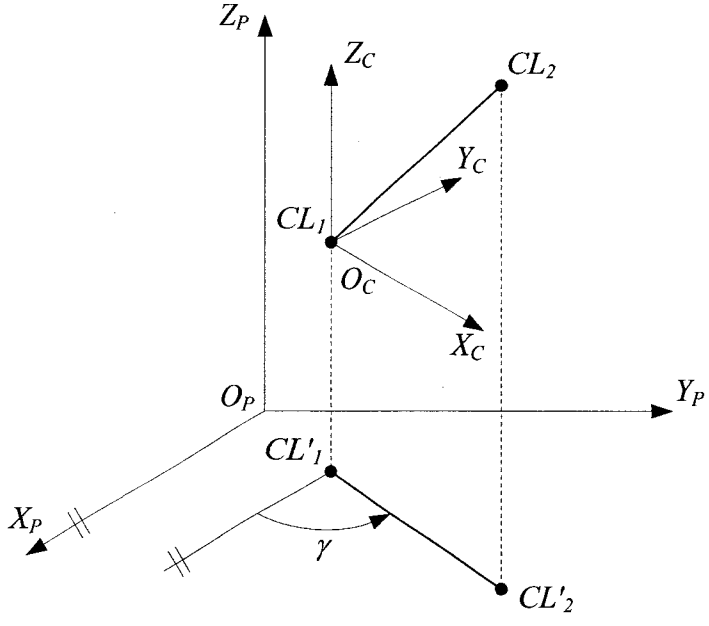


Figure 5.4 Part and local cutter coordinate system

The transformation matrix from the local cutter coordinate to the part coordinate system is

$$T_{C-P} = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & x_1 \\ \sin(\gamma) & \cos(\gamma) & 0 & y_1 \\ 0 & 0 & 1 & z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.2}$$

where γ is the angle between X_P-axis and $\overrightarrow{CL_1' CL_2'}$. And the coordinates of CL_1 and CL_2 in the local cutter coordinate system are

$$CL_{1,C} = [0,0,0]^T, CL_{2,C} = [x_{2,C}, 0, z_{2,C}]^T \tag{5.3}$$

where $x_{2,C} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ and $z_{2,C} = z_2 - z_1$. So the generic equation of the envelope of the cutting circles can be found in the local coordinate system, and then is transformed to the part coordinate system using $T_{C\text{-}P}$.

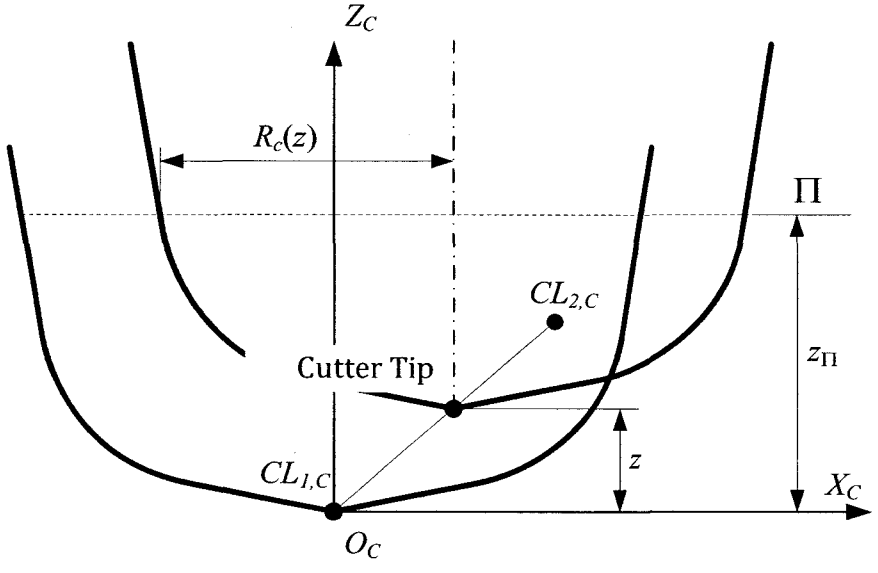


Figure 5.5 Formulation of cutter swept profile

To find the generic equation of the envelope of the cutting circles in the local coordinate system, a horizontal plane Π is set up at z_Π, as shown in Figure 5.5. While the cutter moves from CL_1 to CL_2, the (x, y) coordinate of the cutter tip can be calculated as functions of its z coordinate

$$\begin{cases} x = \dfrac{x_{2,C}}{z_{2,C}} \cdot z & \left(0 \le z \le z_{2,C}\right) \\ y = 0 \end{cases} \tag{5.4}$$

The point (x, y) also represents the center of the cutting circle on plane Π. Based on Eq. (5.1), the radius of cutting circle at plane Π can be calculated as a function of cutter tip's z coordinate

$$R_c(z) = \begin{cases} \dfrac{z_\Pi - z}{\tan\alpha} & \left(z_\Pi - h_1 < z \le z_\Pi\right) \\ R_1 + \sqrt{R_2^2 - \left(h_R - z_\Pi + z\right)^2} & \left(z_\Pi - h_2 < z \le z_\Pi - h_1\right) \\ R_1 + \dfrac{R_2 + (z_\Pi - z - h_R)\sin\beta}{\cos\beta} & \left(z \le z_\Pi - h_2\right) \end{cases} \tag{5.5}$$

Therefore, the equation of the cutting circle on plane Π is

$$\begin{cases} x = \dfrac{x_{2,C}}{z_{2,C}} \cdot z + R_c(z) \cdot \cos\theta & \left(0 \le \theta < 2 \cdot \pi\right) \\ y = R_c(z) \cdot \sin\theta \\ z = z_\Pi \end{cases} \tag{5.6}$$

where θ is the parameter of the circle. When z varies in the interval $\left[0, \min\left(z_{2,C}, z_\Pi\right)\right]$, Eq. (5.6) represents a family of circles on plane Π. The necessary condition for the envelop of the plane curve family $\left(x(\theta,z), y(\theta,z)\right)$ is that the Jacobian of the function x, y must vanish along a locus [90]

$$\frac{\partial x}{\partial z} \cdot \frac{\partial y}{\partial \theta} - \frac{\partial x}{\partial \theta} \cdot \frac{\partial y}{\partial z} = 0 \tag{5.7}$$

Solving Eqs. (5.5)(5.6) and (5.7), the relation between variables θ and z can be found as

$$\cos\theta = -\frac{\partial R_c}{\partial z}\cdot\frac{z_{2,C}}{x_{2,C}} \tag{5.8}$$

where

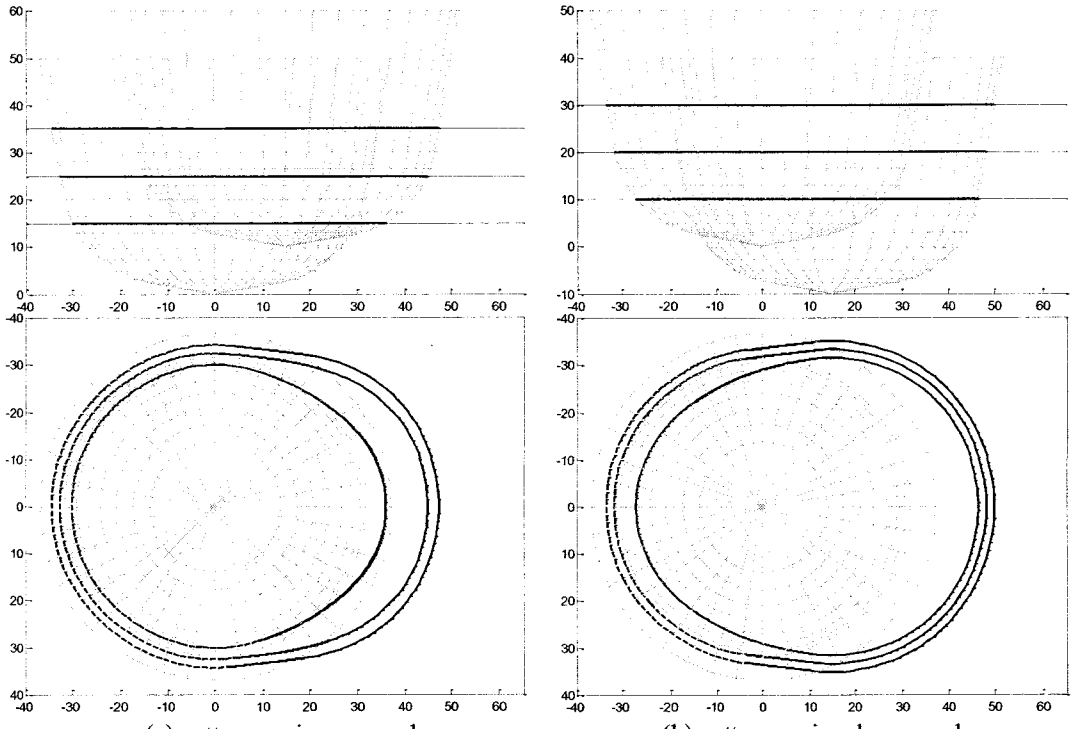
$$\frac{\partial R_c}{\partial z} = \begin{cases} -\dfrac{1}{\tan\alpha} & \left(z_\Pi - h_1 < z \le z_\Pi\right) \\ \dfrac{-\left(h_R - z_\Pi + z\right)}{\sqrt{R_2^2 - \left(h_R - z_\Pi + z\right)^2}} & \left(z_\Pi - h_2 < z \le z_\Pi - h_1\right) \\ -\tan\beta & \left(z \le z_\Pi - h_2\right) \end{cases} \tag{5.9}$$

By plugging θ into Eq.(5.6), the generic formula of the envelope curve at plane Π is obtained as

$$\begin{cases} x = \dfrac{x_{2,C}}{z_{2,C}}\cdot z - R_c(z)\cdot\dfrac{\partial R_c}{\partial z}\cdot\dfrac{z_{2,C}}{x_{2,C}} \\ y = \pm R_c(z)\cdot\sqrt{1-\left(\dfrac{\partial R_c}{\partial z}\cdot\dfrac{z_{2,C}}{y_{2,C}}\right)^2} \end{cases} \quad \left(0 \le z \le \min\left(z_{2,C}, z_\Pi\right)\right) \tag{5.10}$$

Eqs. (5.5) (5.9) and (5.10) can be used to calculated the cutter swept profile at any location z_Π, when $z_{2,C} > 0$ (cutter moving upwards) or $z_{2,C} < 0$ (cutter moving downwards). If $z_{2,C} = 0$, the feed direction is horizontal, and the envelope on a horizontal layer is swept by a cutting circle with constant radius moving along a line segment, which can be solved directly. Figure 5.6 illustrates swept profiles

calculated at three different heights using the swept profile equations while cutter moving upwards and downwards.



(a) cutter moving upwards (b) cutter moving downwards

Figure 5.6 Cutter swept profile samples

5.3 Geometric Model of 3-Axis Milling

The cutter swept profile described above is based on the material removal mechanism on a horizontal layer, which is perpendicular to the tool axis in three-axis milling. To work with this horizontal swept profile, a Z-level B-Rep model is proposed to represent the instantaneous workpiece geometry during machining. Then, the CWE geometry is extracted by calculating the intersection between the workpiece layer profile and the cutter swept profile.

5.3.1 In-Process Model Definition

The in-process model is defined as the intermediary state of the workpiece during the machining process. In this work, it is generated by slicing the initial workpiece stock into layers by a number of planes perpendicular to the tool axis. The accuracy of the model can be adjusted by changing the distance between the slicing planes. Each layer contains a 2D profile which represents the shape of this layer and its height. The B-Rep data structure of the layer profile is the same as the model defined in Figure 3.5 for 2½-axis modeling. Figure 5.7 shows a sculptured surface part, the stair-like workpiece shape after roughing and the layer-based in-process model generated from the roughing geometry.

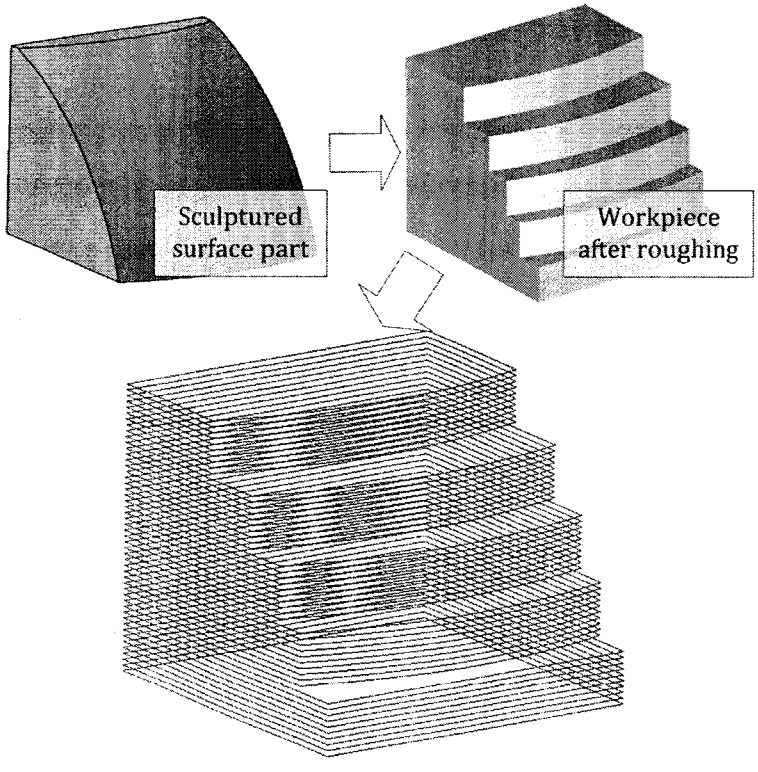


Figure 5.7 In-process workpiece model

5.3.2 Cutter Workpiece Engagement Geometry

The CWE geometry is extracted by moving the cutting tool along the tool path while it rotates one revolution. In practice, tool paths are transferred to the CNC machine tool as part programs, and the motion commands of the part program are processed in real-time by the CNC interpolator. In order to obtain the cutter location accurately, the real-time interpolator algorithms reviewed in Section 2.3 are integrated into our simulation system for calculating tool moving steps.

The process of extracting cutter workpiece engagement geometry includes three steps. First, at each tool movement, by comparing the cutter location and the in-process model, all layers inside the cutter swept range can be found during this motion. Then, the cutter swept profile on each layer can be calculated using the derived equations. By calculating the intersection between the cutter swept profile and workpiece layer profile, the material removal profile on each layer can be calculated, as shown in Figure 5.8. At the same time, the in-process model layer profile is updated to reflect the material removal. Finally, the 3D engagement geometry model is obtained by combining all extracted layer engagement profiles with their location and thickness. The result engagement geometry has the same data structure as the in-process model, a Z-level B-Rep representation. The material removal rate can be estimated from the volume of the removal material.

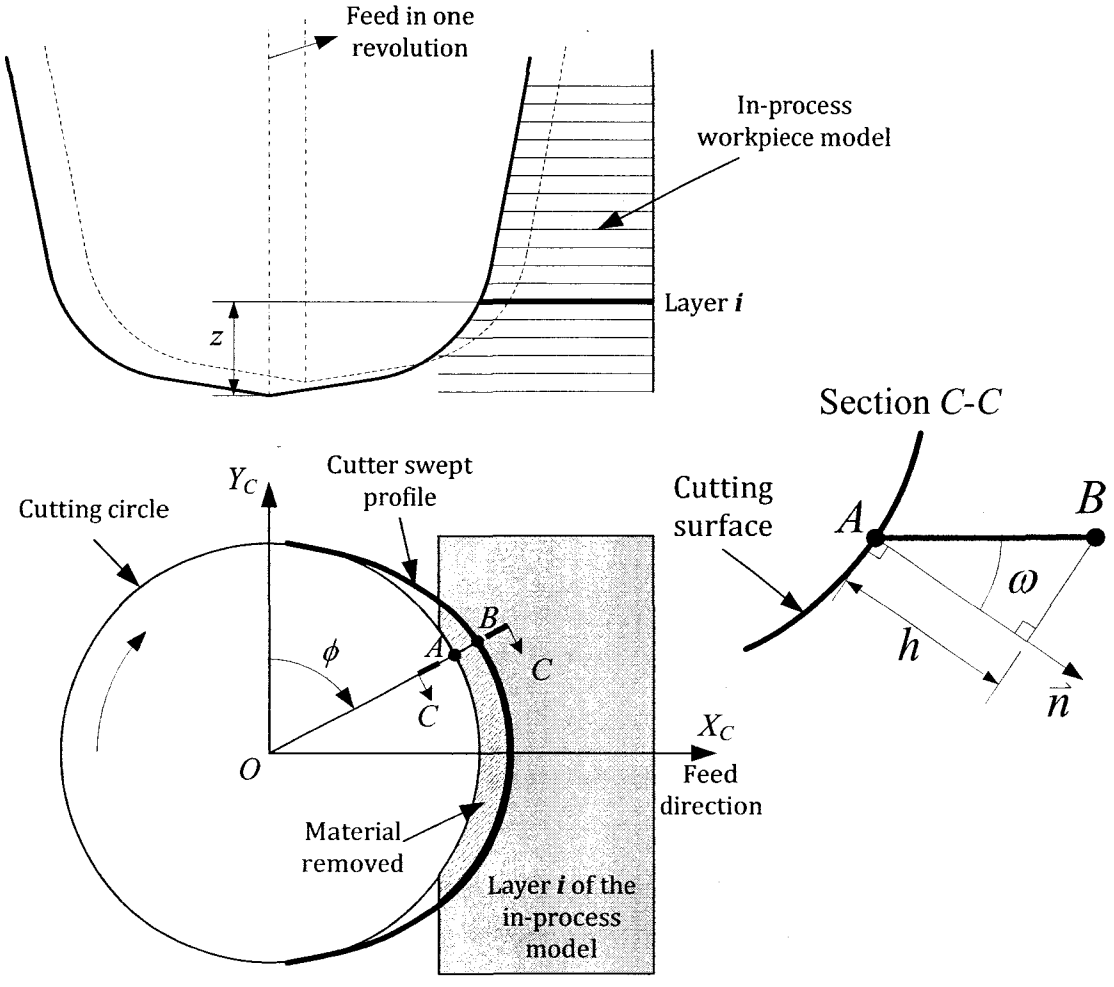


Figure 5.8 Cutter workpiece engagement model

The undeformed chip thickness is calculated from the extracted CWE geometry at any instantaneous immersion angle ϕ. As shown in Figure 5.8, the extracted engagement layer profile is intersected with a ray, which starts from the cutter center and inclines at the immersion angle ϕ. The distance between two intersection points A and B represents the thickness of removal material along the horizontal plane. The undeformed chip thickness normal to the cutting surface is calculated as

$$h(\phi) = |AB| \cdot \cos\omega / N \tag{5.11}$$

where N is the number of flutes and ω is the angle between the horizontal plane and cutting surface normal vector $\vec{n}$. For APT tool defined in Figure 5.2, ω can be represented as a function of the axial height z

$$\omega(z)=\begin{cases}\pi/2-\alpha & \text{lower conic surface}\\ \arccos\left((h_R-z)/R_2\right) & \text{toroidal surface}\\ \beta & \text{upper conic surface}\end{cases} \tag{5.12}$$

5.4 Mechanistic Model of 3-Axis Milling

In the force calculation, the cutting tool is modeled by several helical flutes wrapped around the cutting surface. The helical angle of the cutting flute depends on the cutting surface geometry. For the flat-end mill, the helical angle is usually constant. For ball-end, bull-nose and tapered end mill, the helical angle can be designed arbitrarily and may vary from cutter to cutter. For one flute, the lag angle ψ at axial height z is depended on the cutter's design. Popular designs include constant lead and constant helix angle. The geometric models of these two type designs for general milling cutter can be found in reference [38]. Another way to get the lag angle for a particular cutter is through measurement. By using mathematical formula or measurement, the lag angle along the tool axis can be represented as a function $\psi(z)$.

Assume a cutter with N number of flutes working at spindle speed n (*rev/min*). At time t, the immersion angle $\phi(t)$ of the bottom end of one flute is used as

the reference. For the jth flute, the angular position at axial height z can be calculated as

$$\phi_j(z) = \phi(t) + j\phi_P - \psi(z) \tag{5.13}$$

where the cutter pitch angle $\phi_P = 2\pi/N$ and reference immersion angle $\phi(t) = 2\pi nt/60$. Then, the differential tangential ($dF_{t,j}$), radial ($dF_{r,j}$) and axial ($dF_{a,j}$) cutting forces at a cutting flute element can be calculated using the linear edge force model

$$\begin{aligned} dF_{t,j}(z) &= \left[\mathrm{K_{tc}} h_j(\phi_j(z)) + \mathrm{K_{te}}\right] dz \\ dF_{r,j}(z) &= \left[\mathrm{K_{rc}} h_j(\phi_j(z)) + \mathrm{K_{re}}\right] dz \\ dF_{a,j}(z) &= \left[\mathrm{K_{ac}} h_j(\phi_j(z)) + \mathrm{K_{ae}}\right] dz \end{aligned} \tag{5.14}$$

where $h_j(\phi_j(z))$ is the undeformed chip thickness which can be evaluated from Eq.(5.11), dz is the differential flute element length, K_{tc}, K_{rc}, K_{ac} are the cutting coefficients contributed by the shearing action in tangential, radial and axial directions, respectively, and K_{te}, K_{re}, K_{ae} are the edge constants. These coefficients can be identified for a particular cutter vs. workpiece material by using existing orthogonal to oblique cutting transformation method or mechanistic modeling method. Related methods have been reviewed in Section 2.2. Generally, for flat-end mill with constant helix angle, the cutting coefficients are constant values; for cutter with changing helical angle, such as ball-end mill, the cutting coefficients are considered as a function of the axial location z.

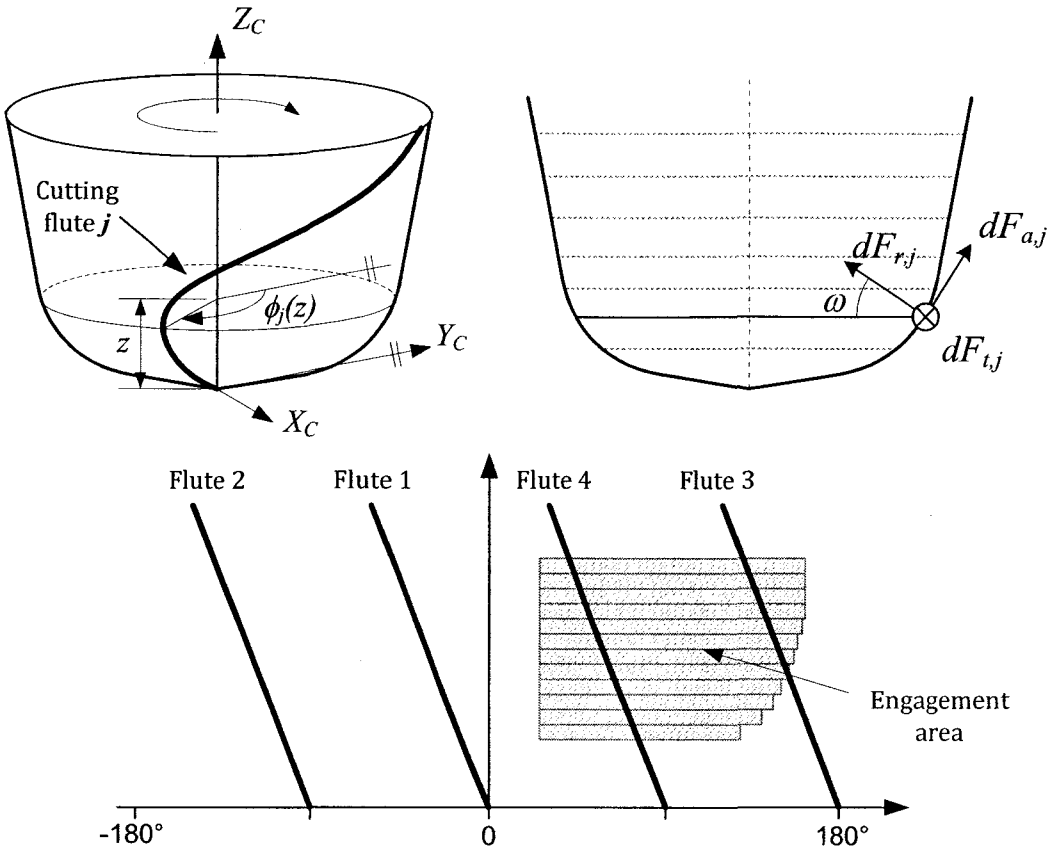


Figure 5.9 Mechanics of APT tool in 3-axis milling

The differential forces are projected to the local Cartesian cutter coordinate system

$$\begin{aligned} dF_{x,j}(z) &= -dF_{t,j}\cos\phi_j(z) - \left(dF_{r,j}\cos\omega(z) - dF_{a,j}\sin\omega(z)\right)\sin\phi_j(z) \\ dF_{y,j}(z) &= dF_{t,j}\sin\phi_j(z) - \left(dF_{r,j}\cos\omega(z) - dF_{a,j}\sin\omega(z)\right)\cos\phi_j(z) \\ dF_{z,j}(z) &= dF_{r,j}\sin\omega(z) + dF_{a,j}\cos\omega(z) \end{aligned} \tag{5.15}$$

The instantaneous cutting torque on the spindle is

$$dT_c(z) = R_c(z)\cdot\sum_{j=0}^{N-1} dF_{t,j}(z) \tag{5.16}$$

The total cutting forces F_x, F_y, F_z and torque T_c can be obtained by integrating the differential elements along the cutter axis. The axial integration limits for each flute

are determined from the cutter workpiece engagement area, which is obtained from the geometric simulation, as shown in Figure 5.9. The total cutting forces can be transferred to the part coordinate system using $T_{C\text{-}P}$ of Eq.(5.2).

The resultant bending force on the cutter is defined as the vector summation of the x, y forces, which has a magnitude

$$F = \sqrt{F_x^2 + F_y^2} \tag{5.17}$$

The cutting power drawn from the spindle is calculated as

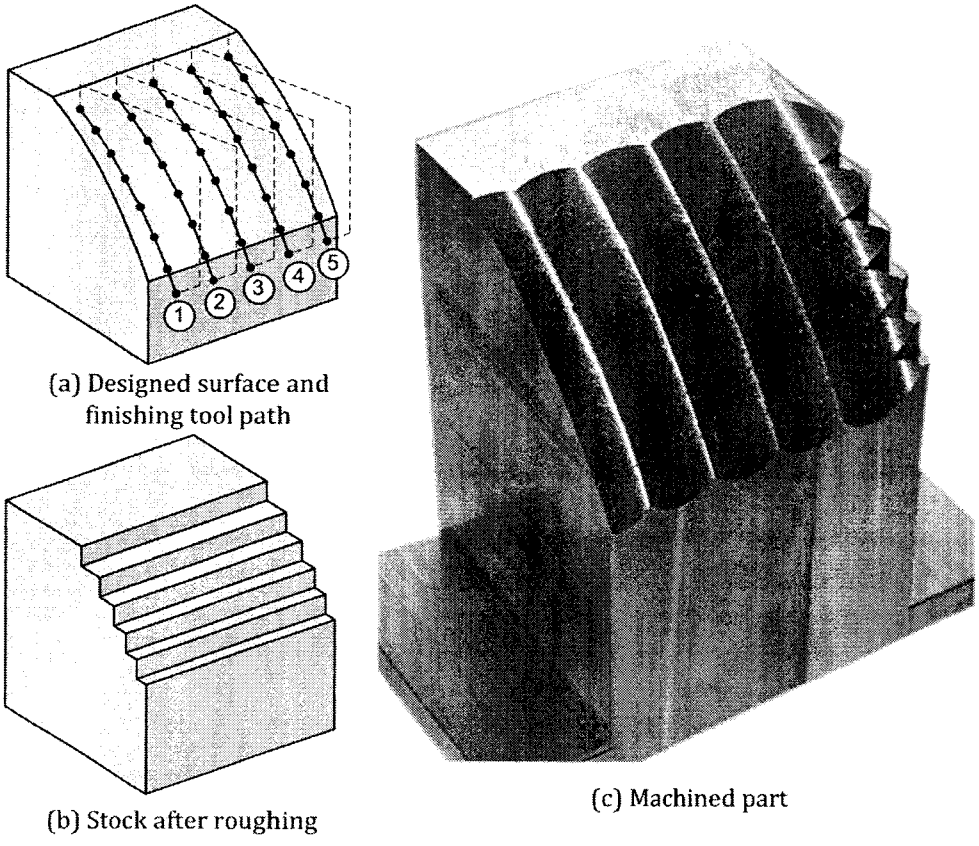
$$P = 2\pi n \cdot T_c / 60 \tag{5.18}$$

5.5 Implementation and Verification

The proposed methodology has been implemented on a personal computer by using the C++ programming language for calculation and OpenGL for visualization. The implementation is a process simulation system for three-axis milling. The inputs to the system include tool paths, workpiece geometry, CNC interpolator parameters, cutter geometry and cutting constants, and the outputs are cutting forces, torque, power, material remove rate, chip and workpiece geometry at each instantaneous moment. By adopting industrial standard file formats, including APT files for exchanging tool paths with CAM software and IGES files for exchange geometry with CAD software, the system can be easily integrated with commercial CAD/CAM software packages.

Experimental verification is performed by using the developed system to predict the cutting forces of a surface milling process and comparing the results with actual forces measured from machining the test part. A solid carbide flat-end mill cutter with *4* flutes, *30 degree* helix angle and *19.05mm* diameter is used to cut aluminum alloy Al7075 workpiece material on a Makino A81M-5XR horizontal machining center. A Kistler 9255B three-axis dynamometer is used for measuring the cutting forces along the x, y and z axis, see Figure 3.13 for experimental setup. The cutting coefficients with cutting fluid on are identified in Table 3.1.

The test part with a sculptured surface is machined using one-way sweeping finishing tool paths. First, a layer based 2½-axis roughing is applied to the rectangular stock. Then, the finishing tool path is applied to the stair-like workpiece shape after roughing, see Figure 5.10. The spindle speed is *1000RPM* and the feed rate is *400 mm/min*. The measured cutting forces of the second path are compared against simulation results in Figure 5.11. The simulated cutting forces are in good agreement with the experiment. The CWE geometry and cutting forces during one revolution at two sample locations are shown in Figure 5.12. The difference in the plots is attributed to the errors in the experimental setup, e.g. tool wear of experimental cutter and cutter runout. For different instantaneous cutter location, the average resultant force during one revolution of the simulation results are within 10% of the experimental results.

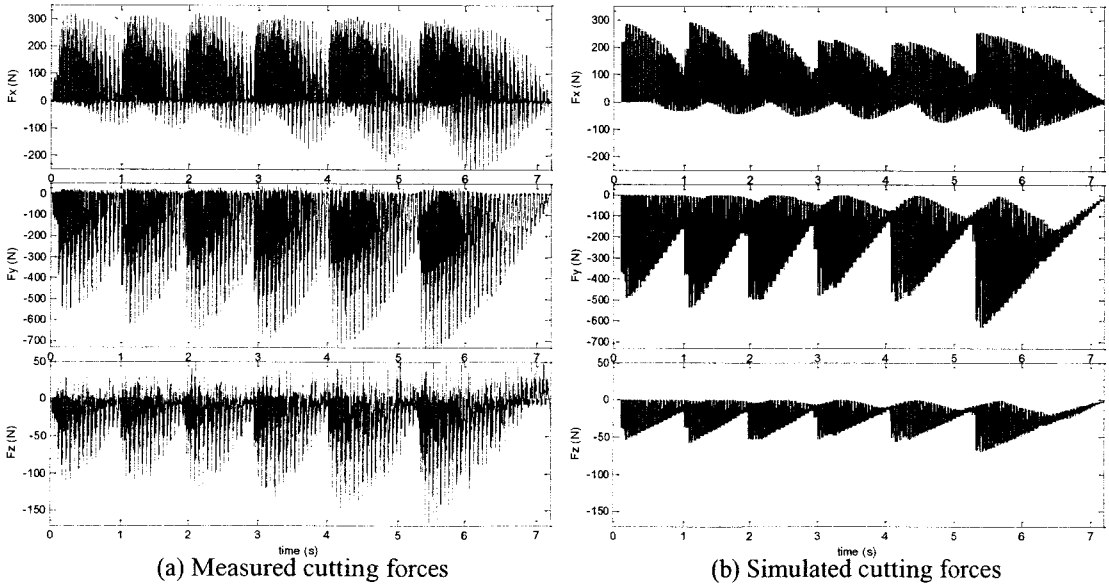


(a) Designed surface and finishing tool path

(b) Stock after roughing

(c) Machined part

Figure 5.10 Test part for 3-axis milling



(a) Measured cutting forces

(b) Simulated cutting forces

Figure 5.11 Cutting forces of tool path #2

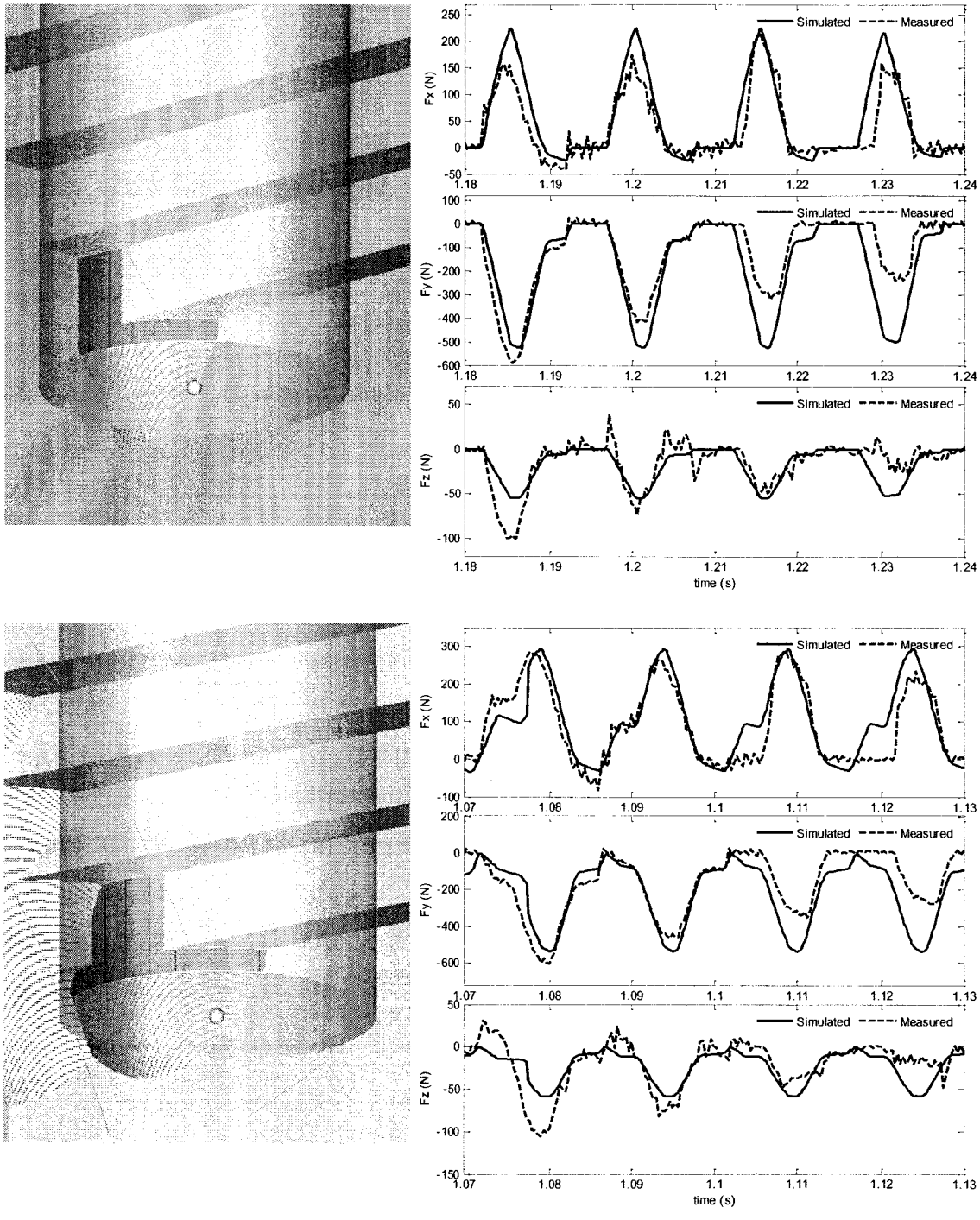


Figure 5.12 Instantaneous CWE geometry and cutting forces of 3-axis milling

5.6 Summary

This chapter presents an efficient, accurate approach to extracting the cutter/workpiece engagement geometry of 3-axis milling of sculptured surfaces. In our research, a basic geometric modeling of chip removal in three-axis milling is investigated, and an effective model is proposed to represent the cutter swept profile. Computationally efficient, closed-form formulations are derived for general APT cutter geometry. A Z-level B-Rep model is adopted to represent the in-process workpiece model, and the CWE geometry is extracted as intersection between the cutter swept profile and the workpiece layer profile. By integrating the CWE geometry with an existing mechanistic force model, instantaneous cutting forces, torque and power can be predicted. A surface milling test is conducted, and measured cutting forces are in good agreement with the simulation results.

Chapter 6 Medial Axis Transforms of Free-form Pockets with Islands

6.1 Introduction

Our interest in MAT stems from our previous work on cutter selection and tool path generation for 2½ axis NC machining. In CAD/CAM systems, part boundaries are normally represented as parametric curves, including straight lines, circular arcs and Non-Uniform Rational B-Spline (NURBS) curves. In this chapter, we present a new, efficient algorithm for calculating the MATs of general pockets whose boundaries are represented as piecewise connected free-form parametric curves. The proposed approach is based on the idea of boundary tracing and decomposition. By studying the geometric properties of contact circles and their center points on MA, a mathematical model of the contact circle is formulated, and a hybrid optimization method, the integration of the particle swarm optimization (PSO) and local optimization methods, is proposed to solve this problem. The contact circle algorithm is used extensively in the boundary tracing process. The contact circle at a boundary point is determined from the perpendicular direction of the tangent vector, which is very sensitive on the curvature perturbation of the boundary. To make our algorithm robust, this instability is handled by inserting compensate points on the other side of the boundary being traced. Our algorithm

starts with the computation of MAT for a pocket without islands, whose boundary is represented as a closed composite curve. While travelling along the boundary, the boundary profile is divided into simpler sub-profiles when a branch circle is reached, and the same boundary travelling scheme is applied to each sub-profile. By following this process recursively, the whole profile can be broken into a number of simple profiles, which contain only one single MA segment. A tree data structure is developed to keep track of the boundary decomposition process. Then we extend our method to deal with pockets with islands by introducing a profile splitting procedure. A pocket profile with islands can always be divided into several simple profiles without islands, and the MAT can be calculated by combining the MAT of each simple profile. Finally, implementation, illustration samples and comparison with existing methods are presented to demonstrate the advantages of the proposed approach.

6.2 Geometric Properties of Medial Axis

This section presents different types of points on the MA and a geometric property that gives an up bound of the contact circle. This geometric property is important to the algorithm introduced in the next section.

6.2.1 Boundary Representation

For the purpose of discussion, the representation of the pocket boundary needs to be defined before studying the properties of its MAT. It is common practice in CAD/CAM software systems to use parametric curve representatives. A

parametric curve in planar domain can be represented with parameter u, which is generally defined as

$$C(u) = \begin{bmatrix} x(u) & y(u) \end{bmatrix}^T \tag{6.1}$$

where $u \in \begin{bmatrix} u_{\min} & u_{\max} \end{bmatrix}$. Different types of parametric curves, including straight lines, circular arcs and Non-Uniform Rational B-Spline (NURBS) curves, are used to design mechanical parts. A complex pocket profile is usually defined by connecting several curve segments together. In order to treat a group of certain individual constituent entities into one logical unit, the *composite curve* data structure is introduced to describe the profile. A *composite curve* is defined as an ordered list of parametric curve entities. It is a directed curve, and each constituent curve entity has the same direction. For a closed profile, the start point of the composite curve coincides with its end point.

In this work, the *composite curve* is defined under the assumption that each constituent curve entity has at least continuous first order derivative with respect to the parameter u ($\geq C^1$ continuity). So the whole composite curve is differentiable except at finite number of connecting points, where both the left and right derivatives can be calculated from the connected parametric curve entities. Practically, this assumption can be easily satisfied in current CAD systems by breaking C^0 continuity curve into several smooth segments ($\geq C^1$) at points with tangent discontinuity.

Since islands may exist in mechanical designs, one or more closed profiles are required to describe boundaries of islands. In order to represent the material side and empty space side, the direction of each profile must be defined properly. Then, a general pocket shape is defined as one outer counter clockwise closed composite curve with zero or several clockwise closed composite curves inside. So when travelling along the outside boundary or inside island boundary, the material is always on the right side. This data structure is inclusive enough to cover all practical pocket shapes.

6.2.2 Medial Axis Points

By definition, the MA is the locus of centers of circles which are locally maximal inside the profile. A circle is locally maximal if there are no other circles inside the profile that contain it. The point on the MA is referred as the *medial axis point*, and the associated maximal circle is called the *contact circle*. A contact circle may touch the profile boundary at one or several points or arcs, which are called *contact components*. Based on the number of contact components of a contact circle, the associated medial axis point can be classified into three types: *2-contact point*, *branch point* and *terminal point*.

The whole MAT can be viewed as a graph, as shown in Figure 1.1, where *branch points* and *terminal points* are connected by simple MA segment curves. A *simple MA segment* only contains *2-contact points*, and is limited by branch points or terminal points. The *2-contact point* is the basic element of the MA segment curve.

Figure 6.1 shows a 2-contact point and the associated contact circle touching the boundary at two points.

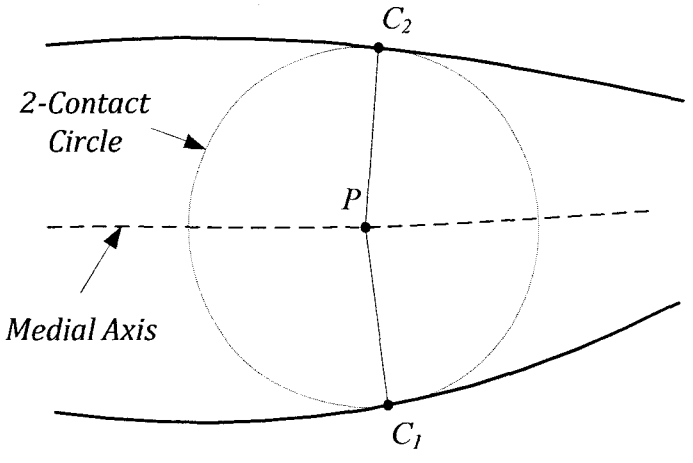


Figure 6.1 Contact circle with 2 contact points

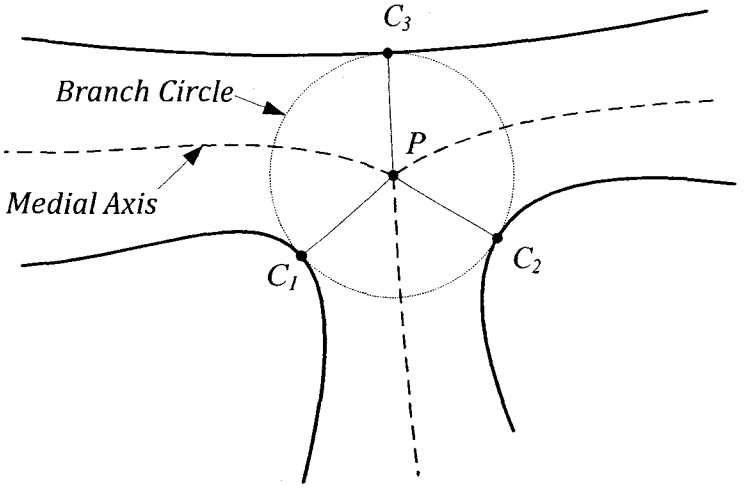


Figure 6.2 Branch circle with 3 contact points

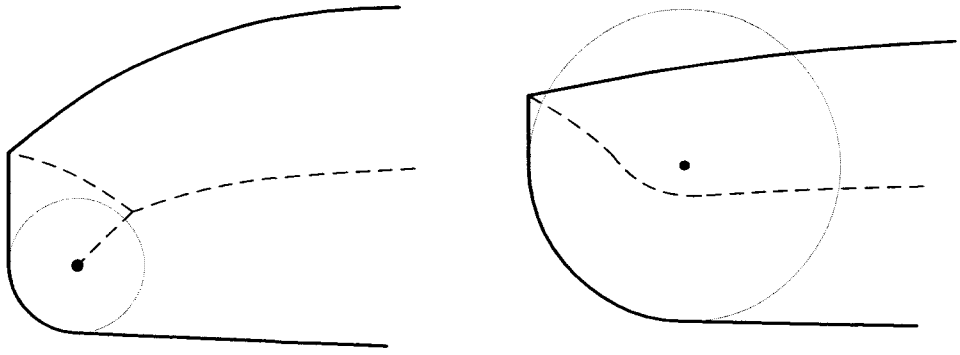
A *branch point* has a contact circle touching three or more separate components on the boundary. Topologically, if a contact circle has n contact components, the branch point is where n medial axis segments connected together, see Figure 6.2. In general case, a branch point has three medial axis segments

connected to it. A branch point with $n > 3$ may be viewed as a form of geometric degeneracy.

The *terminal point* of medial axis has only one contact component. By checking the geometry property of the piecewise connected parametric boundary curves, terminal points can be identified at corner points, circular arc segments and local maximal positive curvature points of free-form curve segments.

A boundary point is a *corner* if the tangent direction discontinues at the point (only *G0* continuity). For the composite curve used in this work, both *sharp corners* and *dull corners* can be identified at connecting points between constituent curve entities. If the interior angle (the angle between the left and right tangent vectors) is strictly less than π, the corner is a *sharp corner*; if the interior angle is larger than π, it is called *dull corner*. The contact circle at a sharp corner is degenerated to a point, and the sharp corner point itself is a terminal point of the MA.

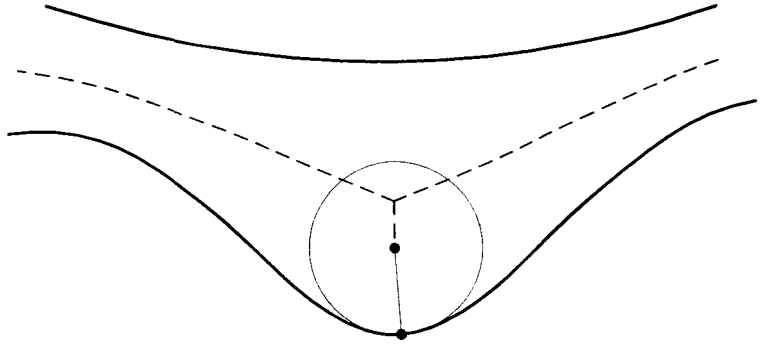
The center point of a circular arc boundary segment is a terminal point if the whole circle of the arc is inside the profile. The arc boundary segment itself is the only one contact component, as shown in Figure 6.3(a). For a free-form curve boundary segment, the osculating circle at point with local maximal curvature can be used to determine the terminal point. If the whole osculating circle is inside the profile, the center of the osculating circle is a terminal point of the medial axis, see Figure 6.4(a).



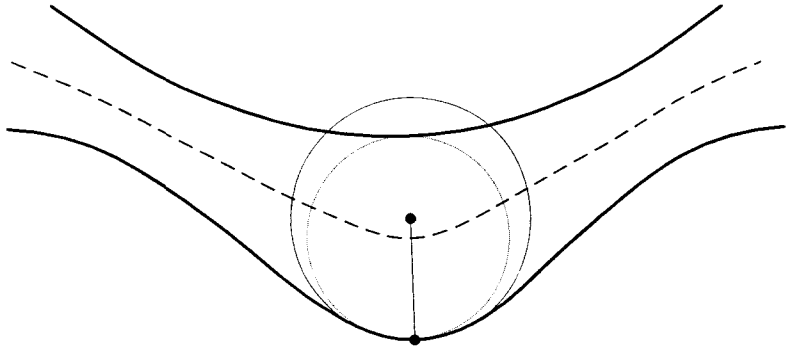
(a) Circle inside the profile
Center = terminal point

(b) Circle outside the profile
Center ≠ terminal point

Figure 6.3 Determine terminal point from circular arc boundary segment



(a) Osculating circle inside the profile
Center = terminal point



(b) Osculating circle outside the profile
Center ≠ terminal point

Figure 6.4 Determine terminal point from osculating circle

6.2.3 Up Bound of a Contact Circle

The diameter of the contact circle at any boundary point is less than the distance between the boundary point and the first "visible" point along the normal

direction. This geometric property is obtained from the observation of the contact circle, and can be proofed by the definition of the MA.

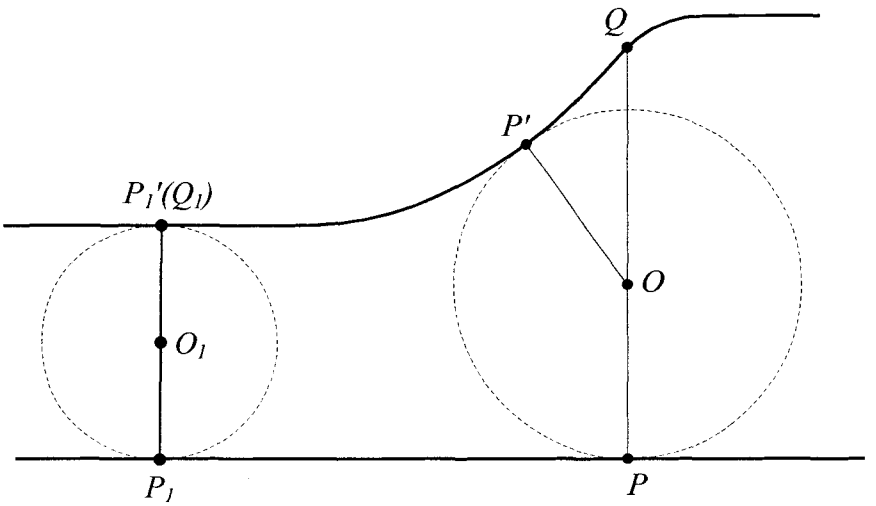


Figure 6.5 The up bound of contact circles at boundary points

Let P be a point on the profile boundary. A ray can be casted from P inwards along the normal direction at P. The ray will intersect with the profile at least once. The first "visible" point Q is the intersection point closest to P. A contact circle can also be drawn at P and touches the other side of profile at point P'. By the definition of medial axis, the contact circle is the largest circle inside the profile, and no boundary point will be inside the maximal circle. Therefore, the point Q, which is a point on the boundary, cannot be inside the contact circle. Since $\overrightarrow{PQ}$ is along the normal direction, the center point of the contact circle O is on $\overline{PQ}$. Because Q is outside the circle, the length $|OQ|$ is larger than the contact circle radius. Therefore, the length $|PQ|$, which represents the distance between the boundary point and the first "visible" point, is larger than the contact circle diameter. At some boundary

point, such as P_1 in Figure 6.5, the first "visual" point Q_1 is coincident with the other contact point P_1'. The length $|P_1Q_1|$ is equal to the contact circle diameter.

The distance between the boundary point and the first "visible" point along the normal direction gives a very useful up bound, which serves as a constraint to reduce the searching domain and increase the efficiency of our implementation algorithm.

6.3 Contact Circle Algorithm

The most fundamental problem in the MAT calculation is to find the contact circle at any location along the profile. In this section, a mathematical model of the contact circle is formulated, and a hybrid optimization method, the combination of the particle swarm optimization (PSO) and local optimization methods, is proposed to solve this problem.

6.3.1 Mathematical Model of the Contact Circle

Given a free-form profile represented as a parametric curve $C(u)=\begin{bmatrix} x(u) & y(u) \end{bmatrix}^T$, let $P=\begin{bmatrix} x(u_P) & y(u_P) \end{bmatrix}^T$ be a point on the profile, and the corresponding unit normal vector $\overline{n_P}=\begin{bmatrix} n_{x,P} & n_{y,P} \end{bmatrix}^T$ can be calculated. The first "visible" point Q is calculated as the closest intersection between the ray along the direction $\overline{n_P}$ and the profile. Let C_Q denote the circle tangent with the profile at P and passing point Q, and its center point $O_Q=\begin{bmatrix} x_{O_Q} & y_{O_Q} \end{bmatrix}^T$ is the midpoint of line

segment PQ. As discussed in the previous section, the contact circle at point P is smaller than the circle C_Q. In other words, all points of the contact circle are inside the circle C_Q, except the point P, where two circles are tangent at.

Then, select another profile point $T = C(u)$, where $u \neq u_P$ and the point is inside the circle C_Q. The criterion that a point is inside a circle can be expressed as the distance from the point to the center of the circle is less than the radius of the circle. The mid-section line of vector $\overline{PT}$ intersects with the line along the normal vector $\overline{n_P}$ at point O_T. Define a circle centered at O_T and passing through both profile points P and T (see Figure 6.6), and the radius of the circle can be formulated as

$$r(u) = \frac{1}{2} \cdot \frac{\left(x(u) - x_P\right)^2 + \left(y(u) - y_P\right)^2}{\left(x(u) - x_P\right) \cdot n_{x,P} + \left(y(u) - y_P\right) \cdot n_{y,P}} \tag{6.2}$$

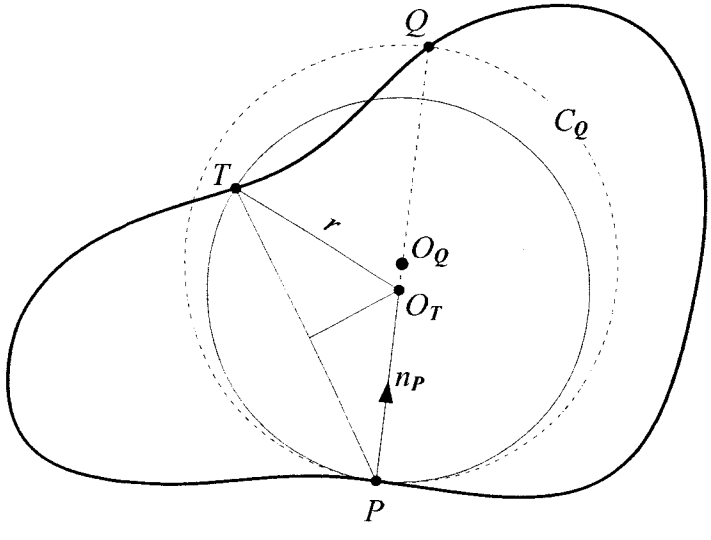


Figure 6.6 Mathematical model of contact circle

By testing all profile points inside the circle C_Q, a serial of testing circles can be obtained using the above equation. At one special testing point, the radius of the testing circle is the minimum, where the testing point T is a tangent point between the profile and the circle. This minimum testing circle is the contact circle at point P. Therefore, the contact circle searching problem can be formulated as

$$\begin{aligned} &\text{Minimize} \quad r(u) = \frac{1}{2} \cdot \frac{\left(x(u)-x_P\right)^2 + \left(y(u)-y_P\right)^2}{\left(x(u)-x_P\right)\cdot n_{x,P} + \left(y(u)-y_P\right)\cdot n_{y,P}} \\ &\text{Subject to} \quad \begin{cases} u \neq u_P \\ \sqrt{\left(x(u)-x_{O_Q}\right)^2 + \left(y(u)-y_{O_Q}\right)^2} \leq \dfrac{|PQ|}{2} \end{cases} \end{aligned} \tag{6.3}$$

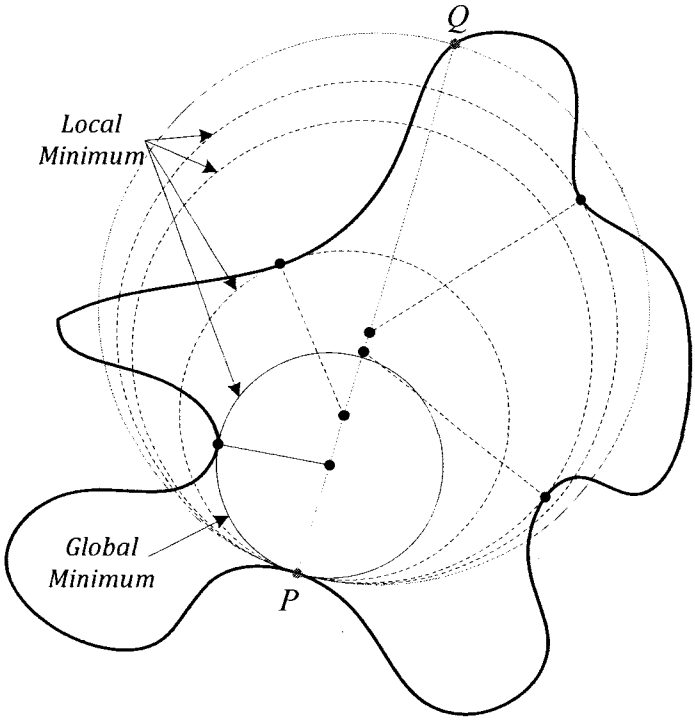


Figure 6.7 Example of the global optimization problem

The above model is a global optimization problem due to the high non-linearity of the free-form curve boundary, as the example illustrated in Figure 6.7. In this work, a hybrid optimization method is developed to solve this problem.

6.3.2 Hybrid Global Optimization Method

The particle swarm optimization (PSO), genetic algorithm, and simulated annealing methods are often used to solve global optimization problems. Since these solvers employ the stochastic technique, they can be slow in finding the global solution within a small tolerance. The well-established local optimization methods are quick to converge at a local solution with a given start point; however, they could not find the global solution. By taking the advantages of these two types of methods and overcome their drawbacks, a hybrid method is proposed, which includes two steps. First, the PSO method is employed to search for the local solution region where the global optimum locates; then, a Newton method is applied to find the global optimum.

The PSO method is featured with population-based search by imitating a swarm of scattered particles exploring the problem domain in order to find the global solution to a complicated optimization problem [91,92]. The problem domain of the contact circle searching problem is the parametric space of the boundary curve, which is a one dimensional continuous domain. Since the goal of the PSO method is to find a start point for the local optimization, a discrete PSO searching method is adopted to increase efficiency. By setting a small step value, the PSO solver only evaluated the object function $r(u)$ at discrete points whose parameter

values are integral times of the step value. Thus the continuous problem domain is represented as a fairly large number of discrete points. The initial swarm of particles are scattered around the discrete problem domain; at each iteration, new locations of the particles are adjusted to the closest discrete parameter values. The population size, the inertia weight, and the acceleration coefficients are crucial to efficiency and convergence of the PSO method; however, these parameters are dependent on different problems, and there is no fixed rule for them. In this work, the population size is determined according to the number of discrete points of the parameter domain, specifically, *20* particles per *1000* discrete points. The initial inertia weight is *0.9*, the final inertia weight is *0.2*, and acceleration constants are set to *2.0*. It has been verified that with above parameters, the PSO method can effectively solve the global optimization problem.

After the PSO method converges to a location which is the global minimum among all points of the discrete parameter domain. This parameter value is used as the start value for the following Newton method to find the accurate local solution.

A function is formed from the dot product of two vectors

$$\begin{aligned} f(u) &= C'(u) \cdot \overrightarrow{TO_T} \\ &= x' \cdot \left(x_P + r \cdot n_{x,P} - x\right) + y' \cdot \left(y_P + r \cdot n_{y,P} - y\right) \end{aligned} \tag{6.4}$$

The object function $r(u)$ is minimum when equation $f(u) = 0$, which can be solved by the Newton method. Let u_i be the parameter obtained at the ith Newton iteration, then

$$u_{i+1} = u_i - \frac{f(u_i)}{f'(u_i)} \tag{6.5}$$

where

$$\begin{aligned} f'(u) = x'' \cdot \left(x_P + r \cdot n_{x,P} - x\right) + x' \cdot \left(r' \cdot n_{x,P} - x'\right) \\ + y'' \cdot \left(y_P + r \cdot n_{y,P} - y\right) + y' \cdot \left(r' \cdot n_{y,P} - y'\right) \end{aligned} \tag{6.6}$$

and

$$\begin{aligned} r'(u) = & \frac{\left(x - x_P\right) \cdot x' + \left(y - y_P\right) y'}{\left(x - x_P\right) \cdot n_{x,P} + \left(y - y_P\right) \cdot n_{y,P}} \\ & - \frac{\left[\left(x - x_P\right)^2 + \left(y - y_P\right)^2\right] \cdot \left(n_{x,P} \cdot x' + n_{y,P} \cdot y'\right)}{2 \cdot \left[\left(x - x_P\right) \cdot n_{x,P} + \left(y - y_P\right) \cdot n_{y,P}\right]^2} \end{aligned} \tag{6.7}$$

The convergence criterion is given by

$$\left|C(u_{i+1}) - C(u_i)\right| \leq \varepsilon \tag{6.8}$$

where ε is the predefined tolerance.

6.3.3 Comparison of Computational Efficiency

The contact circle searching algorithm is developed based on the above hybrid optimization method. To demonstrate its higher computational efficiency, the hybrid method is compared to a previous developed method [93], which adopts a binary searching based method and utilizes the point projection function of an existing geometric kernel software. Both methods are implemented using C++ programming language and are applied to the test profile shown in Figure 6.7. The

total length of the test profile is *995mm*, and the boundary is divided into *2000* discrete locations in the PSO method. For calculating the contact circles at *1000* testing points evenly distributed along the profile, the previous developed method takes *4286ms*, and the hybrid method only takes *1919ms*. The result shows that the hybrid method is more efficient in solving the complex global optimization problem in this work.

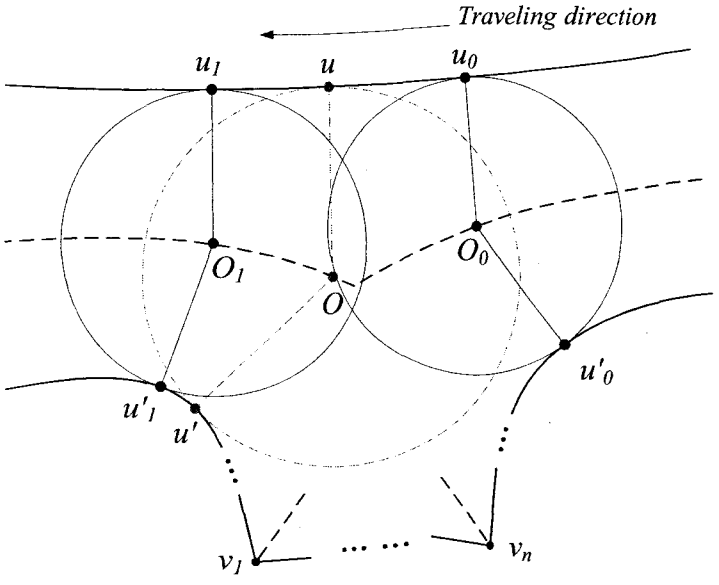
6.4 Branch Circle Algorithm

To find a branch circle while traveling along the boundary, the first step is to detect whether branch circles exist between two successive sample points. Then a searching algorithm is developed to find the number of branch circles and location of each branch circle within the specified tolerance.

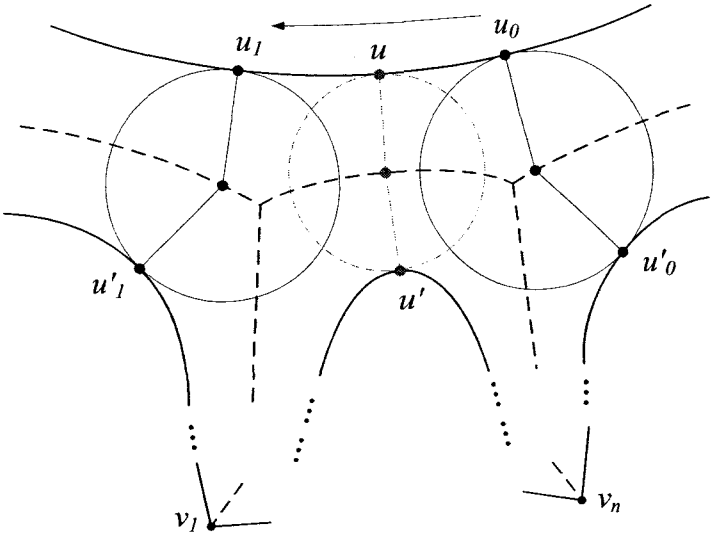
Let u_0 be the parameter value of the first sample point and the next sample point is u_1, as shown in Figure 6.8. By applying the contact circle algorithm on u_0 and u_1, the other two contact points and corresponding parameter values u'_0 and u'_1 can be calculated, respectively. Assume the parameter value increases along the traveling direction. If there is one or more medial axis terminal points between u'_1 and u'_0, there is at least one branch circle between u_0 and u_1. This criterion is derived from the graph topology of the MAT.

It needs to be notices that the above criterion only determines the existence of branch circle(s) between two sample points. The number of branch circles is

determined by recursively dividing the input parameter range. Detail steps are described in the following algorithm.



(a) Only one branch circle between two sample points



(b) More than one branch circles between two sample points

Figure 6.8 Searching for the branch circle

Brach Circle Searching Algorithm

Input: Two successive points on the boundary, parameter values u_0 and u_1

Output: Branch circle center point and radius; parameter values of all contact point on the boundary

Steps:

1) Apply the *contact circle algorithm* to u_0 and u_1. Calculate two other contact points and corresponding parameter values u'_0 and u'_1.

2) Find all terminal points between the parameter range (u'_1, u'_0) on the boundary. Let v_1 be the parameter value of the terminal point nearest to u'_1, and v_n be the parameter value of the terminal point nearest to u'_0.

3) If no terminal points are found between (u'_1, u'_0), terminate the algorithm and return with no branch circle between to u_0 and u_1.

4) Set $u_{\min} = u_0$, and $u_{\max} = u_1$.

5) Set $u = (u_{\max} + u_{\min})/2$.

6) Apply the *contact circle algorithm* to u, calculate the parameter value u' of the other contact point and center point O of the contact circle.

7) Let O_{prev} denote the contact circle center point calculated in previous iteration. If $|O - O_{prev}| < \varepsilon$, where ε is the predefined tolerance, output the contact circle as the result and terminate the algorithm. Otherwise, proceed to the next step.

8) If $u' \in (u'_1, v_1)$, as shown in Figure 6.8(a), set $u_{\max} = u$ and go to Step 5); if $u' \in (v_n, u'_0)$, set $u_{\max} = u$ and go to Step 5).

9) If $u' \in (v_1, v_n)$, as shown in Figure 6.8 (b), there is more than one branch circle between $u_{\min}$ and $u_{\max}$. Call the *branch circle searching algorithm* recursively with input parameters $(u_0 = u_{\min}, u_1 = u)$ and $(u_0 = u, u_1 = u_{\max})$.

One prerequisite of the above algorithm is that parameter values of all the terminal points of the whole boundary need to be calculated and stored beforehand.

6.5 Boundary Tracing Algorithm for a Closed Profile

This section presents the boundary tracing algorithm that extracts the medial axis graph of a closed profile without islands. The basic strategy is to divide the profile into simpler sub-profiles where a branch circle is reached while travelling along the boundary. A tree data structure is developed to keep track of the boundary decomposition process. The developed contact circle searching algorithms are used extensively in the boundary tracing algorithm.

6.5.1 Data Structure

The whole MAT is organized in a tree. Each node contains a profile represented as a closed composite parametric curve. To construct the tree, the input closed profile is taken as the root node. Two operations are applied to the node to introduce new nodes:

- if there is any branch circle inside the node profile, one of the branch circles is used to divide the node into n child nodes, where n is the number of contact components;

- if no branch circle is found inside the profile, the MA segment curve is stored in the node.

Repeat the above operations recursively until all MA segment curves are included in the tree. The result tree has branch circles as internal nodes and MA segment curves as terminal nodes. For a node profile with more than one branch circles, no particular order is specified while selecting a branch circle to divide the node. Consequently, different tree structures can be constructed from the same profile. A sample profile with 5 MA segments and 2 branch circles is shown in Figure 6.9, and two possible tree structures are constructed. The example shows that the tree structure reflects the sequence of the boundary decomposition process.

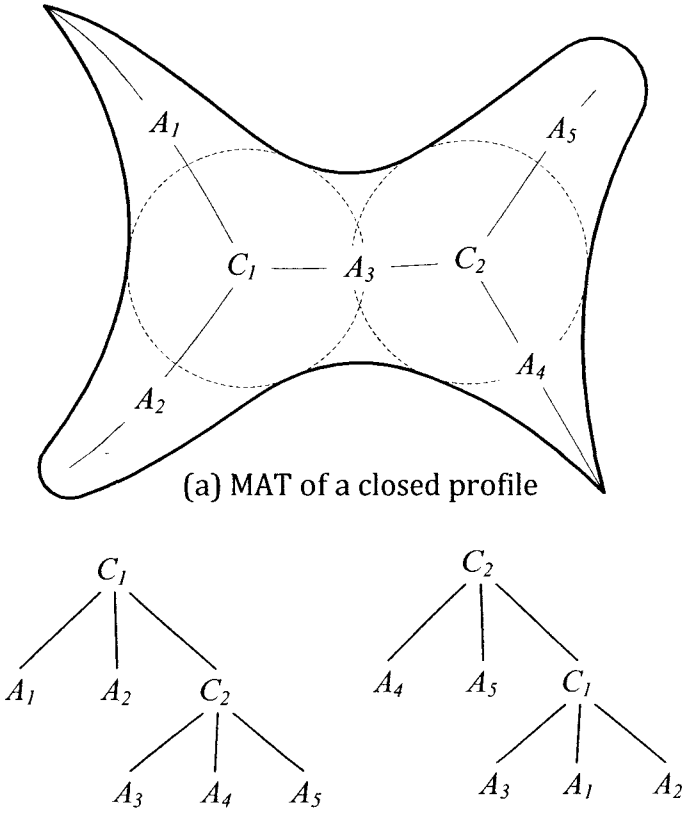


Figure 6.9 Tree data structure

6.5.2 Boundary Tracing Algorithm

Before starting tracing along the boundary, a circular list of all terminal points needs to be constructed. The terminal points are identified according to the geometric properties discussed in Section 6.2. Then, the tracing starts from one of the terminal points and goes along the counter clockwise direction. A user defined sampling distance, usually a small length, is used to evenly distribute the sample points along the boundary. For general free-form parametric curves, numerical methods [94,95] for arc length parameterization can be used to calculate the increment of parameter value according to the arc length increment. The contact circle searching algorithm is used at each sample point to calculate the contact circle

center point. The MA curve is obtained by interpolating the contact circle center points along the sampling sequence with a B-spline curve. Detail steps are described in the following algorithm.

Boundary Tracing Algorithm

Input: A closed composite curve

Output: MAT represented in a tree data structure

Steps:

1) Create the *root* node from the input profile, and the status is *untraveled.*

2) Set the *current node* to the *root.*

3) Start boundary traveling at the *current node.* If it is the *root,* any terminal point can be used as the start point. Otherwise, the terminal point corresponding to the branch circle of its parent node is used as the start point.

4) If a branch circle with n contact points is detected between two successive sample points, break the current profile at the contact points and create n child nodes. One of these child nodes is a terminal node because its MA segment points have already been calculated during the travelling. The rest of the child nodes are *untraveled.* Set *current node* to one of them, then go to Step 3).

5) If a dull corner P is reached, two normal vectors, $\vec{n}_-$ and $\vec{n}_+$, can be calculated from connected two parametric curves. Apply the 2-contact circle algorithm along $\vec{n}_-$ and $\vec{n}_+$, two contact circles (center points O_- and O_+, contact points

P'_- and P'_+) can be calculated. Stop the current travelling at P. A new travelling starts from P'_+ and follows the same decomposition scheme described in the following steps. Eventually, the new travelling stops at P'_-, and the medial axis between O_+ and O_- is calculated. Then, resume the original travelling at P. The above process is illustrated in Figure 6.10.

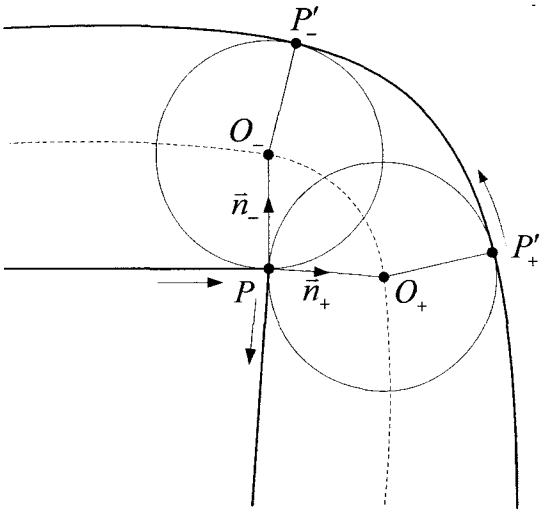


Figure 6.10 Dull corner tracing scheme

6) If a medial axis terminal point is reached and no branch circle is found during the traveling, the *current node* is a terminal node. Save the MA segment curve to the node and set its status to *travelled*. Then, set the *current node* to its parent node and proceed to the next step.

7) Check status of all child nodes of the *current node*. If there is any *untraveled* child node, set the *current node* to one of the *untraveled* child nodes, then go to Step 3).

8) If all child nodes of the *current node* have been travelled and the current node is the *root*, stop traveling and output the MAT tree. Otherwise, set the *current node* to its parent node, then go to Step 7).

An example is shown in Figure 6.11 to illustrate the boundary tracing and decomposition process. Total 5 MA segments and 2 branch circles are found and organized in a tree structure.

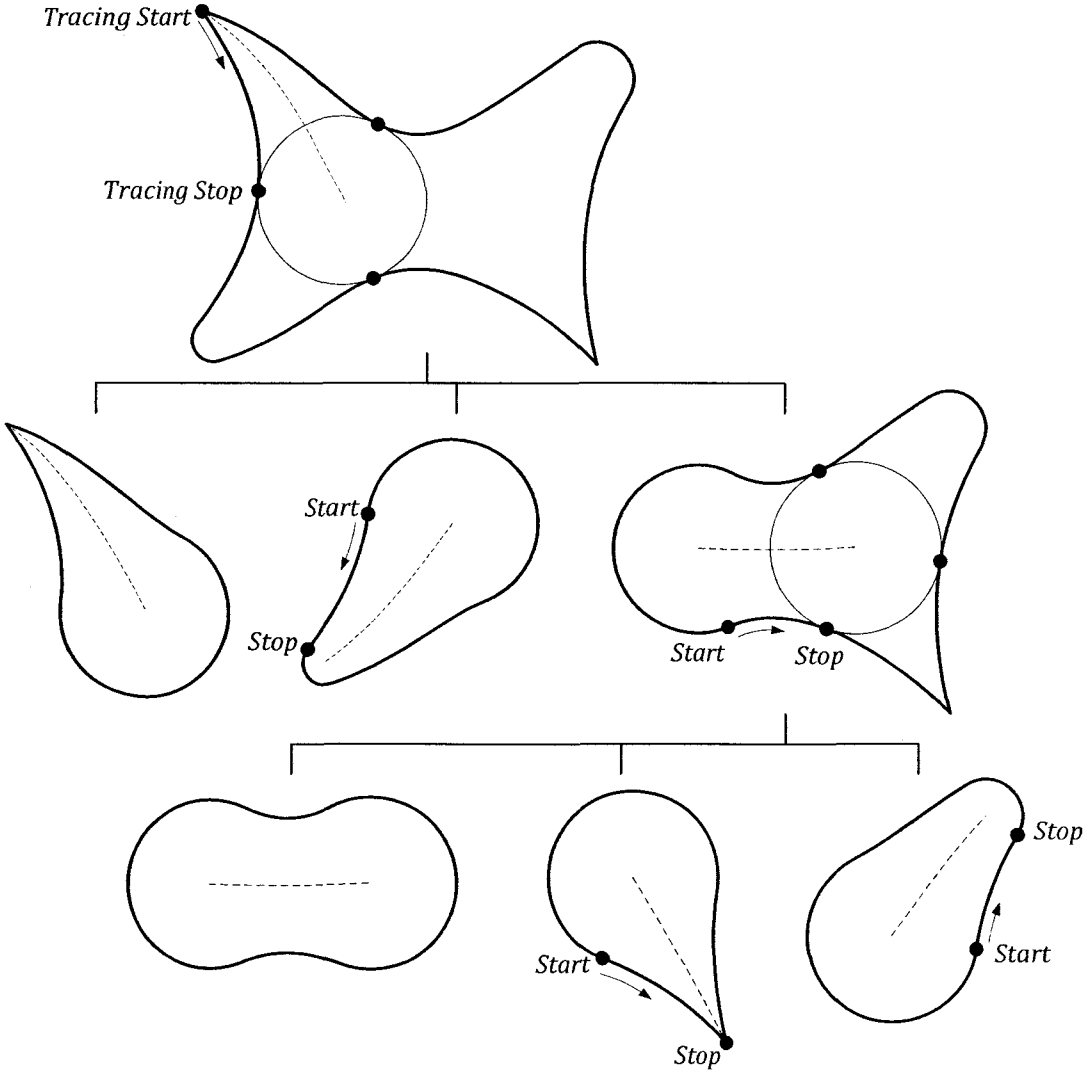


Figure 6.11 Boundary tracing and decomposition process

Because the contact circle is determined from the perpendicular direction of the tangent vector at a boundary point, the result MA curve is very sensitive on the curvature perturbation of the boundary. To overcome this instability, extra compensating points are inserted into the other boundary side of the medial axis being traced. As shown in Figure 6.12, A and B are two successive sample points, and the arc length of AB is equal to the sampling distance Δs. Contact circles can be calculated at O_A and O_B, and the corresponding contact points are A' and B'. If $\left|A'B'\right| \geq 2 \cdot \Delta s$, the number of sample points to be inserted between A' and B' are determined as

$$n = floor\left(\left|A'B'\right| / \Delta s\right) - 1 \tag{6.9}$$

where *floor*() is a function that returns the largest integer that is less than or equal to the input. The compensating points are evenly distributed between A' and B', and points $O_1 \ldots O_n$ are calculated and inserted to the MA segment curve.

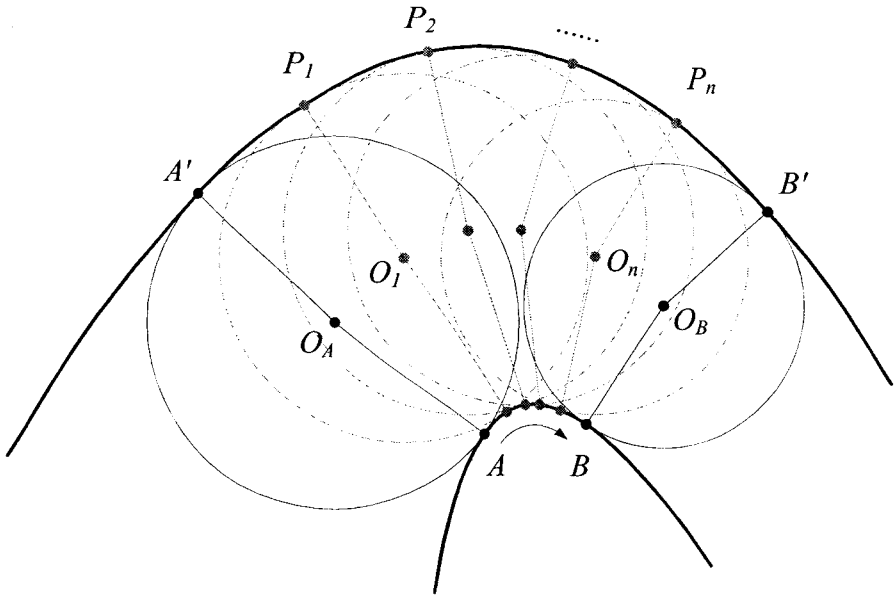


Figure 6.12 Compensating curvature perturbation

6.6 Computing MAT of a Pocket with Islands

The basic strategy to compute the MAT of a pocket with islands is to split the boundary profile into several simpler profiles without islands. For a close profile with one or more islands, the procedure to split the profile is described as the following steps.

1) Select one island of the input profile as the reference.

2) Compute the point with maximum y-coordinate value on the boundary of the reference island, and construct a contact circle at this point. If the contact circle touches the outer boundary, e.g. Figure 6.13(a), proceed to the next step. Otherwise, the contact circle touches another island, e.g. Figure 6.13(b). Then construct another contact circle at the maximum y-coordinate point of

the island being touched. Repeat this operation until the contact circle touches the outer boundary.

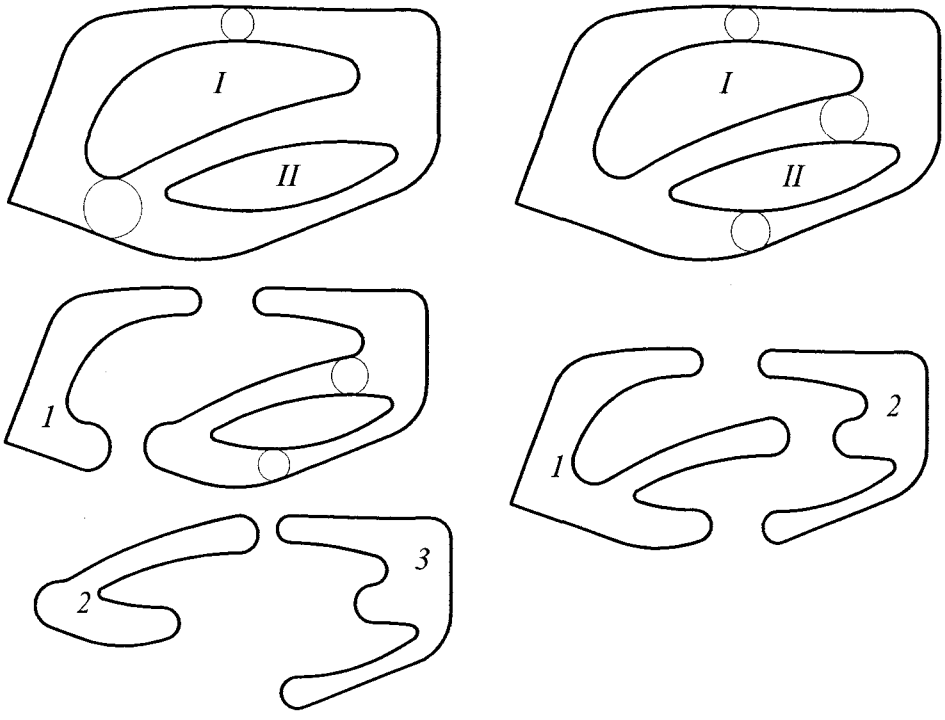
3) Compute the point with minimum y-coordinate value on the reference island, and perform the similar operation as the previous step.

4) Collect all contact circles found in Step 2) and 3), and use these circles to split the profile into two parts.

5) Check both two profiles. If anyone has one or more islands inside, apply Step 1) to 5) to the profile.

6) Repeat the above steps recursively until all profiles contain no island.

Different set of simple profiles can be obtained because no particular order is specified while selecting the reference island, as illustrated in Figure 6.13. After the original profile has been broken down in to a number of simple profiles, the developed boundary tracing algorithm is used to calculate the MAT for each of them. Finally, the complete MAT is obtained by connecting the simple MATs together. For the case that produces different sets of simple profiles, the final MAT results are the same.



(a) First split at hole *I* (b) First split at hole *II*

Figure 6.13 Split pocket profile with islands into a set of simple profiles

6.7 Implementation and Verification

The algorithms described in pervious sections have been implemented on a personal computer by using the C++ programming language and the OpenCASCADE [96] geometric kernel, which provides the implementations of the required general geometric algorithms, such as line/curve intersection, point to curve projection, arc length parameterization and B-Spline interpolation. The IGES interface module of the geometric kernel is utilized to import boundary geometry from commercial CAD software. The developed program has been quantitatively validated against different shapes with varying complexity. Several examples are shown in Figure 6.14 with MAT calculated. The default precision value of the OpenCASCADE geometric kernel for checking coincidence of two points is *1e-7*, which is used in the

developed program for the testing shapes. The computation times shown in Table 6.1 are obtained by setting the sampling distance to 1 and executing the program on a Windows PC with an AMD Althlon 2.7GHz CPU.

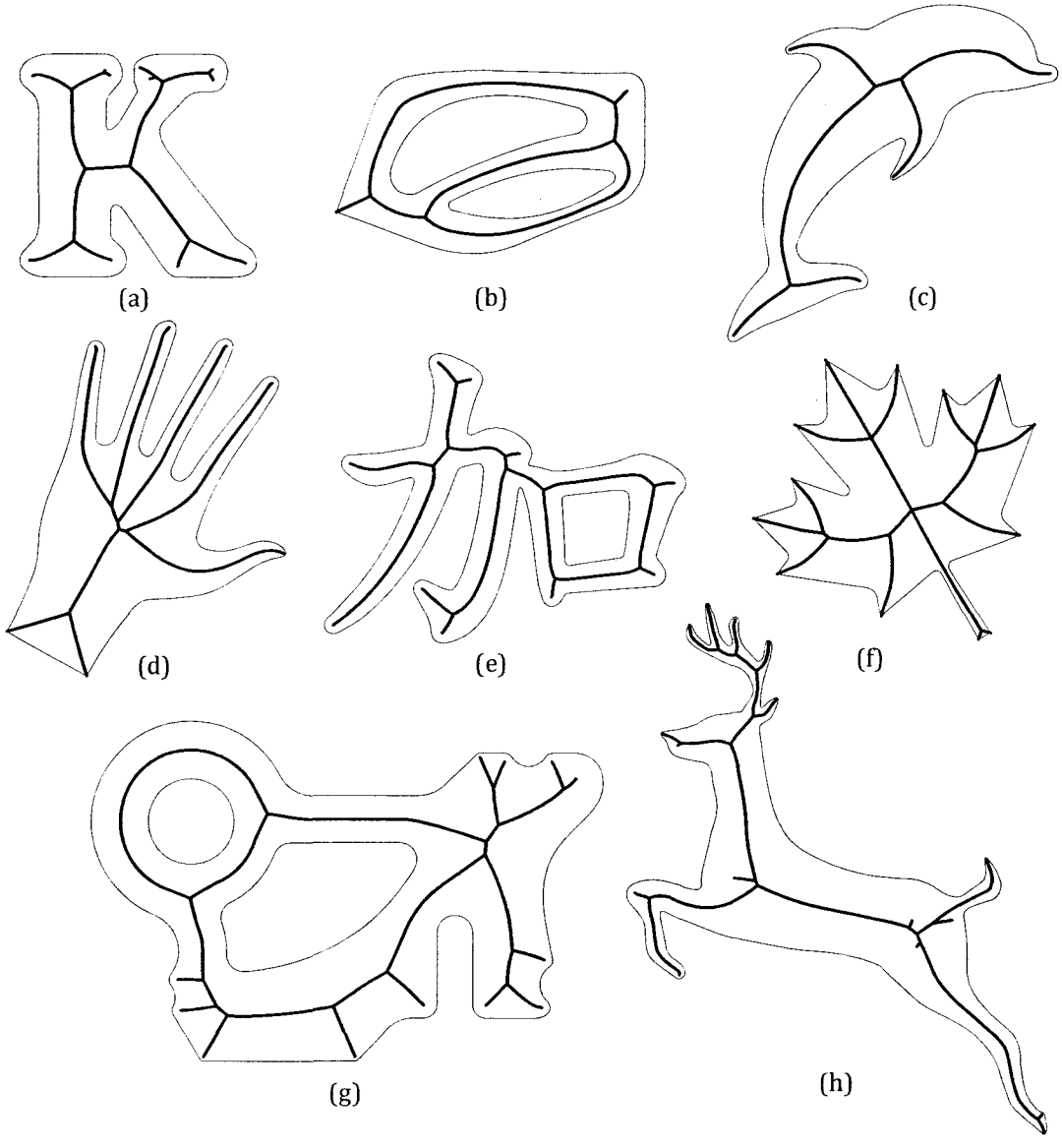


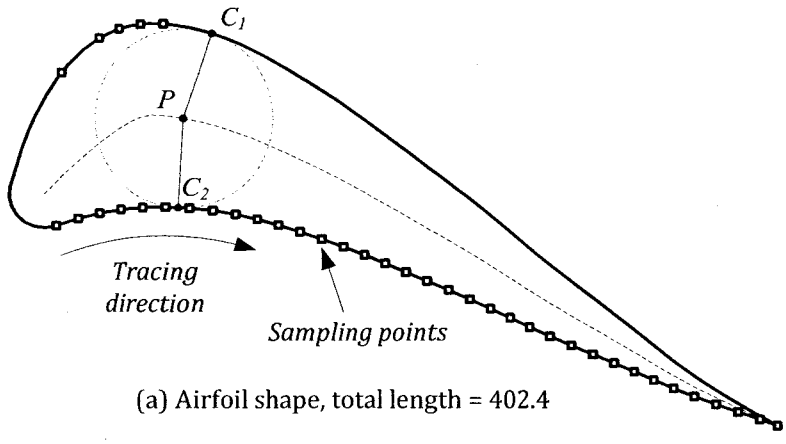
Figure 6.14 MAT of test shapes

Table 6.1 Execution time for the test shapes (sampling distance = 1)

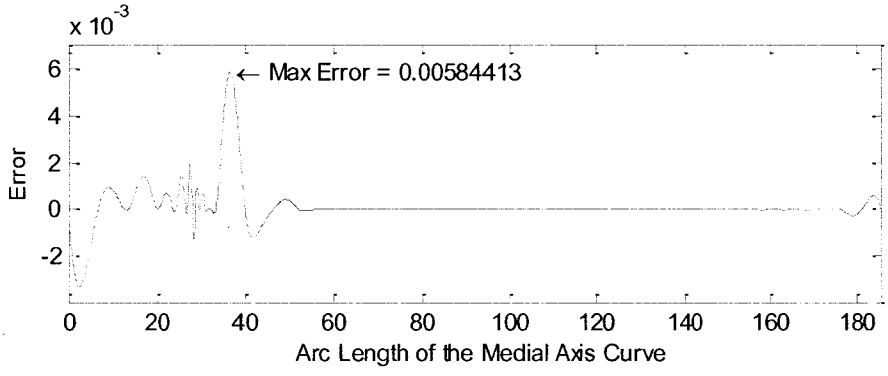
Shape	*Length(mm)*	*Sample points*	*Execution time (ms)*	*MA segments*
(a)	553	322	2028	19
(b)	1119	466	1045	7
(c)	556	316	2246	7
(d)	1068	604	3619	11
(e)	453	279	3135	19
(f)	855	513	517	20
(g)	590	330	2075	28
(h)	1677	895	11903	31

The above examples demonstrate the robustness and effectiveness of the proposed algorithms when dealing with complex shapes. Further study is conducted to verify the accuracy of the proposed algorithms. The accuracy of the MA points calculated at the sampling points is bounded by the precision value (*1e-7*) of the geometric kernel, and the accuracy at other locations is determined by the interpolated curve. At any point P on the MA curve, point C_1 and C_2 are obtained by projecting P onto the two sides of the boundary curve, as shown in Figure 6.15(a). The difference of $|PC_1|$ and $|PC_2|$ is defined as the error for evaluating the accuracy of the calculated MA curve.

An airfoil shape with only one medial axis is examined closely. Figure 6.15 shows the error of the calculated MA curve when the sampling distance is 5. Total 40 sample points (35 points on the side been traced, 4 inserted on the other side to compensate the curvature perturbation) are determined in the boundary tracing algorithm. A number of tests are conducted at different sampling distance, and results are shown in Table 6.2.



(a) Airfoil shape, total length = 402.4



(b) Error along the MA curve, sampling distance = 5

Figure 6.15 Verify the accuracy of the calculated MAT

Table 6.2 Max errors for the airfoil shape

Sampling distance(mm)	*Sample points*	*Execution time (ms)*	*Max error*
20	11	15	3.99
15	15	16	2.08
10	21	31	0.099
5	40	78	0.0058
2	98	171	0.000168
1	196	343	1.08e-5
0.5	390	717	8.11e-7

As been discussed in the literature review that the medial axis of a shape can be computed using the Voronoi diagram of discrete sample points of the shape

boundary and the Voronoi vertices converge to the medial axis for a curve in 2D as the sample density approaches infinity. The Voronoi diagram based method is used to construct the MAT of the testing airfoil shape. Using sampling distance 5, the airfoil shape is discrete into 82 points; the medial axis shown in Figure 6.16 is obtained from the Voronoi diagram of discrete points by removing Voronoi edges that do not belong to the medial axis. The maximum error of the calculated medial axis vertices is 0.2919, and the location is shown in Figure 6.17. The accuracy of the Voronoi diagram based method can be increased by increasing the number of the sample points. A number of tests are conducted at different sampling distance, and results are shown in Table 6.3.

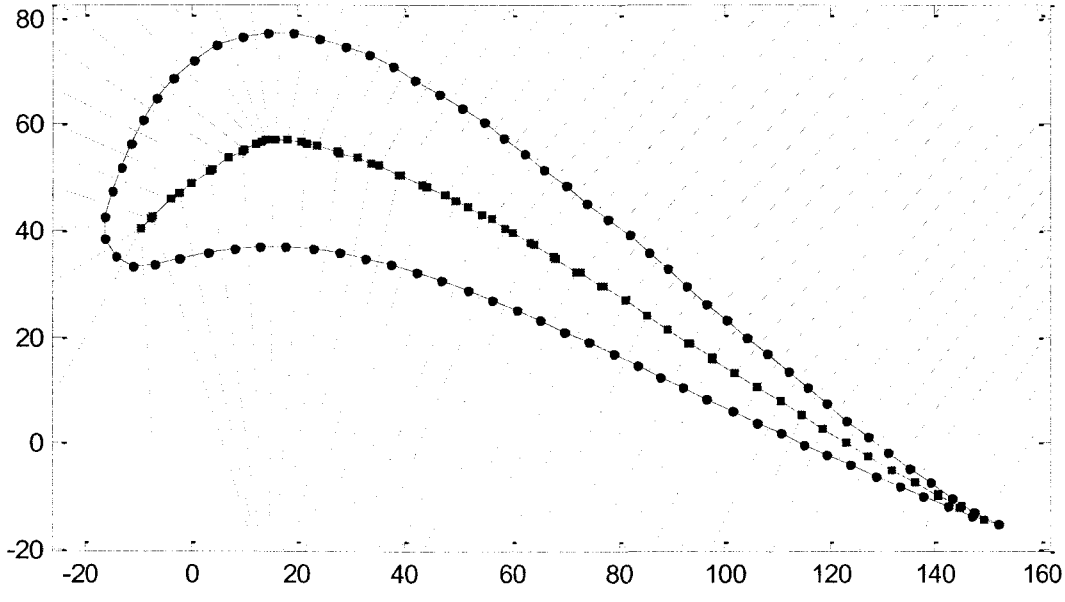


Figure 6.16 MA obtained from Voronoi diagram (82 points)

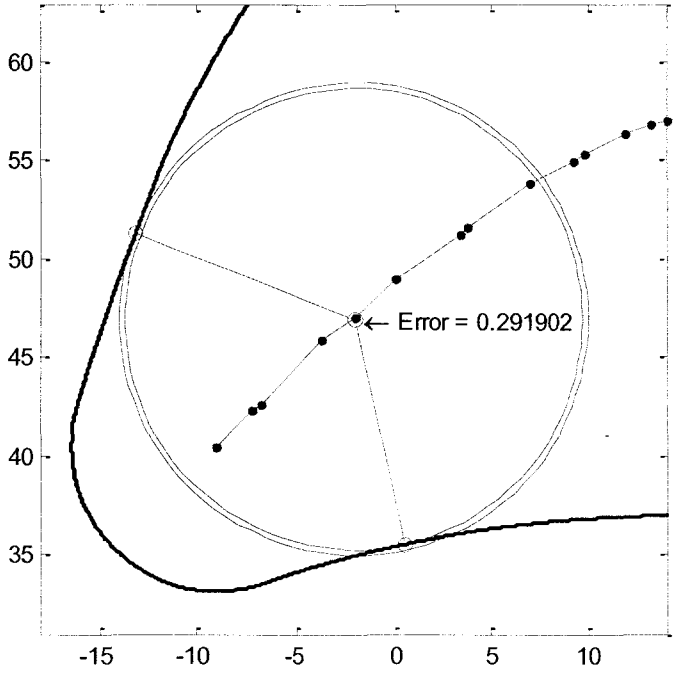


Figure 6.17 Maximum error location of 82 sample points

Table 6.3 Max errors of the Voronoi Diagram based method

Sample distance (mm)	*Sampling points on the profile*	*MA points/Voronoi vertices*	*Max error*
10	41	39/67	0.922371
5	82	80/136	0.291902
2	203	201/343	0.075366
1	405	403/686	0.031823
0.5	807	805/1370	0.012374
0.2	2014	2012/3429	0.004343
0.1	4026	4024/6931	0.001789
0.05	8050	8047/14274	0.000906

By comparing the results of Table 6.2 and Table 6.3, it is shown that our algorithm can achieve higher accuracy with fewer sample points than the method based on the Voronoi diagram of discrete sample points.

Another problem encountered with the Voronoi diagram method is the numerical instability of the float point number when sampling density is very high. In general, computational geometry algorithms are designed with the assumption that all numerical computations are exact. However, the computer represents real numbers using fixed-precision arithmetic by IEEE standard. When sampling distance is very small, the Voronoi vertex calculated using double precision number is not accurate. For example, some portion of the Voronoi diagram of 4026 sample points is shown in Figure 6.18.

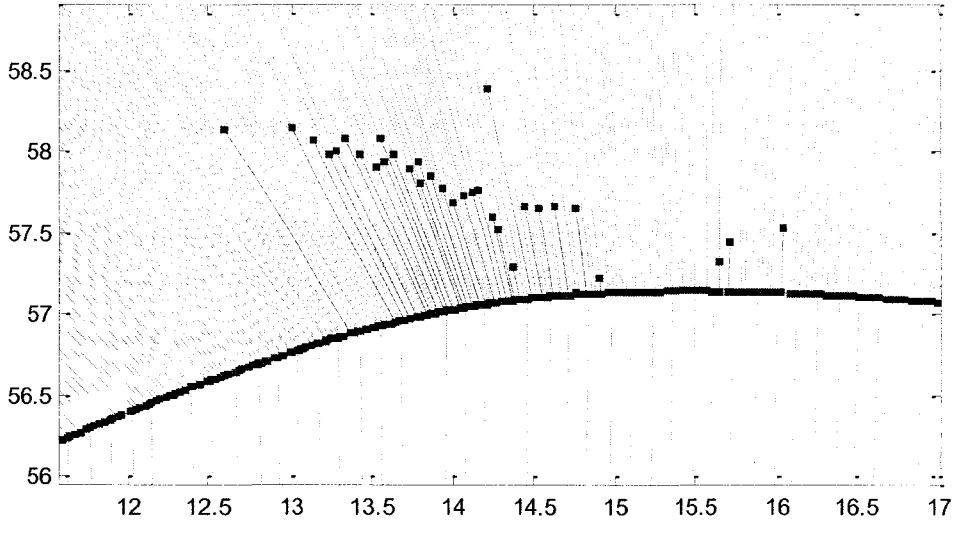


Figure 6.18 Numerical instability of fixed-precision arithmetic

This problem can be solved by using exact geometric computation methods [97,98] at the cost that the computation is much slower than the fixed-precision arithmetic. Since our algorithm employs optimization methods for calculation and fewer sampling points are required, robust computation can be achieved with the fixed-precision arithmetics.

6.8 Summary

This chapter presents a new, efficient geometric approach to approximate the MATs of free-form pockets with islands. The core part of the proposed method is the mathematical model of the contact circle, which is solved using a hybrid optimization method. Based on the boundary tracing and decomposition, the proposed boundary tracing algorithm can handle general pockets with/without islands. The developed algorithms have been quantitatively validated against different shapes with varying complexity. By comparing to the Voronoi diagram based method, it is shown that our approach can achieve higher accuracy with fewer sample points. Since the implementation is based on general geometric algorithms, which are available in popular geometric kernel software, the proposed methodology can be easily integrated with existing CAD/CAM systems.

The MAT can be used in pocketing milling to facilitate tool path generation and cutter selection. Applications of the MAT on roughing tool paths and cutter selection are presented next in Chapter 7.

Chapter 7 Aggressive Roughing Tool Path with Multiple Cutters for Pocket Machining

7.1 Introduction

Pocket is an important feature of mechanical parts. Due to advanced CAD/CAM techniques, pockets with closed free-form boundary curves have been widely adopted in mechanical design and are often cut with 2½-axis CNC milling. Because of the big portion of stock material removed in the pocket machining, roughing is crucial to the productivity, which could simply take more than 60% of the total machining time. Thus, a reduction in roughing time with efficient tool path strategy can considerably increase productivity.

The purpose of roughing is to remove as much material volume as possible and leaving a small amount of un-cut allowance for semi-finishing or finishing. Although the conventional tool path strategies work fine in general pocket machining, they are not the most efficient roughing cuts. In this research, the aggressive pocket roughing is defined as using large standard end-mills to remove stock material inside the pocket with full immerse. The advantages of the aggressive roughing include: (1) the largest cutters are much more rigid and strong and can cut

with greater depths of cut and higher feed rates, and (2) the multiple cutters are employed to access the complex pocket without overcuts. Since it only takes about ten seconds to change a tool in present CNC machine centers, tool change time is not a concern when using multiple cutters in aggressive roughing. Therefore, its material removal rate and cutting efficiency can reach the maximum. However, due to the complicated geometries such as the free-form curve profiles used in the design of mechanical parts, it is quite challenging to calculate the optimum cutter sizes and sequence for efficient machining.

To significantly reduce complex pocket machining time, multiple optimal cutters are selected and their tool paths are generated for aggressive roughing in this chapter. First, the MAT developed in previous chapter is used to generate gouging free tool paths for free-form pockets. This automatically generated tool path is especially suited for narrow and long pocket shapes that are difficult to machine even using conventional tool path strategies. Then, an optimal model of finding multiple largest cutters in terms of the maximum area covered by the tools is established, and the genetic algorithm (GA) is applied to solve this problem. Finally, two examples are presented to demonstrate the advantages of this approach over the conventional method.

7.2 Aggressive Roughing Tool Path Strategy

In geometry, the MA of any pocket is the "middle" paths of the domain constrained by the pocket boundary. The MA of a prismatic pocket is regular;

however, for pockets with free-form boundary, their shapes are more complicated, so their MAs are irregular. Regardless of the pocket shape complexity, more space in the domain is available for objects located along the MA without intersecting the boundary. Thus, if the MA is adopted as tool paths for rough-machining the pocket, larger tools can be used to cut the pocket for higher productivity and without overcutting the pocket. To take advantage of this outstanding feature of the MA, a new strategy of planning tool path is proposed in this work for aggressive rough machining of pockets. The MA curves of the pocket are employed as reference tool paths, which can provide very important information for tool selection and path planning.

7.2.1 Properties of the Reference Tool Path

For any pocket shape, its MAT can be calculated using the algorithm developed in previous chapter. The reference tool paths and their radius functions can be obtained from the MAT as a compact representation of the pocket geometric and topological characteristics. The feature of the reference tool paths is illustrated with an example shown in Figure 7.1. In this example, a pocket is in a shape with three legs and its boundary is composed of several B-spline curves. Its MAT is calculated using the developed algorithm and plotted in the figure. The MA includes 3 terminal points (V_1, V_2, V_3), one branch point (V_B). Three references tool paths (A_1, A_2, A_3) and their radius functions can be attained by retrieving the related information from the tree data structure of the MAT.

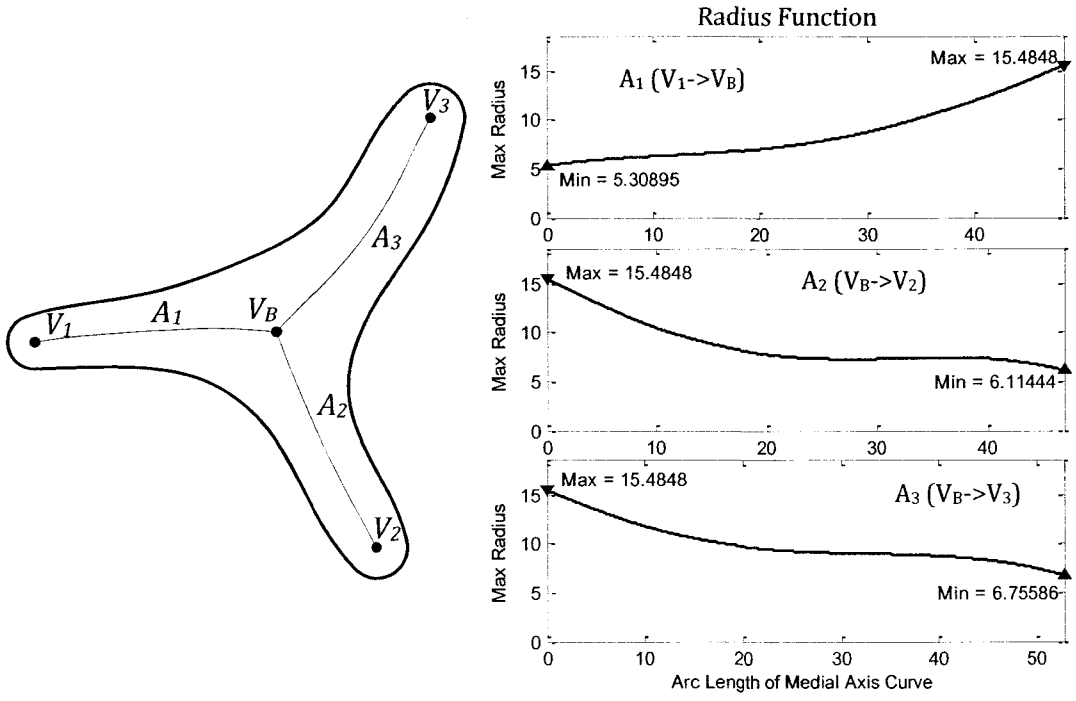


Figure 7.1 A sample pocket shape and its reference tool paths

More specifically, for reference tool path A_1 between V_1 and V_B, the radius function graph is plotted. For a point on the graph, its abscissa value refers to the arc length of a path point from V_1, and its ordinate value refers to the radius of the locally maximal contact circle centered at this path point. The first point of the graph indicates the maximum circle with radius of *5.30985mm* at point V_1, whose arc length is zero. Similarly, the last point of the graph indicates the maximum circle with radius of *15.4848mm* at point V_B, whose arc length is *48.3801mm*. It is evident that this radius function is nonlinear along path A_1. In the same figure, path A_2 between V_B and V_2 and path A_3 between V_B and V_2 are plotted, together with the corresponding radius function graphs. Therefore, the pocket topology is clearly interpreted with the reference paths, and the pocket domain is completely represented by the radius functions.

To conduct aggressive rough machining of a pocket for the highest cutting efficiency, the cutting tools have to be as large as possible and are fully immersed in the stock material along their paths. However, larger cutters are prone to local gouging and global interference with the pocket, and inappropriate tool paths can cause longer machining time. Fortunately, as an automatic geometry interrogation tool, the pocket MAT is able to easily and accurately identify the maximal circle that best fits into the domain at any MA point. Therefore, the reference tool paths and their radius functions provide excellent guide for selecting maximum cutters and their effective paths for full tool-and-stock engagement and without local-and-global overcuts in machining.

7.2.2 Tool Path Generation for a Specific Cutter

According to the above-mentioned properties, it is a principle of planning tool paths for aggressive pocket roughing that the tools cut along the reference paths. This always ensures that each tool has the largest accessible space, thus larger tools can be used and their effective tool paths can be generated accordingly. As an important issue in planning tool paths, detecting local gouging and global interference of tools can be easily conducted with help of the radius functions. In this work, the criterion to identify tool gouging and interference is simply that the radius function value in terms of a cutter location (a MA point) is less than the tool radius. This is easy to understand. If the maximum cutter accessible space, the locally maximal contact circle, at a location is smaller than the tool circle, the tool geometrically intersects with the pocket boundary, which means that the tool

gouges or interferes with the pocket and scrapes the part. In practice, the principle and the criterion can be implemented in planning tool paths in the following way.

Here, tool path planning includes two main steps: (1) to retrieve the reference tool paths and their radius functions from the MAT tree data structure, and (2) to find the intersection points between each radius function graph and a horizontal line representing a specific cutter radius. According to the criterion, compared to this cutter radius, the reference path points with larger values of the radius function define the tool paths for this cutter. To illustrate the path planning process, an example is shown in Figure 7.2. An L-shaped pocket bounded with free-form curves, its reference tool path and the radius function graph are plotted in this figure. Suppose an end-mill with radius of *20mm* is to cut the pocket. To find its tool path, a horizontal line of the cutter radius is drawn on the radius function graph. This line intersects with the function curve at two points s_1 and s_2, which correspond to two points on the reference tool path, O_1 and O_2. Since the function values of the points between s_1 and s_2 are all greater than the cutter radius, their corresponding points on the reference tool path are the valid cutter locations, which is the segment between O_1 and O_2. The shaded area in the pocket is the area covered by the cutter when it moves along the planned tool paths. Furthermore, the horizontal line could intersect with the radius function graph at more than two points (see Figure 7.3); if this case, the cutter has several tool paths, each of which is between two adjacent points and having larger radius function values.

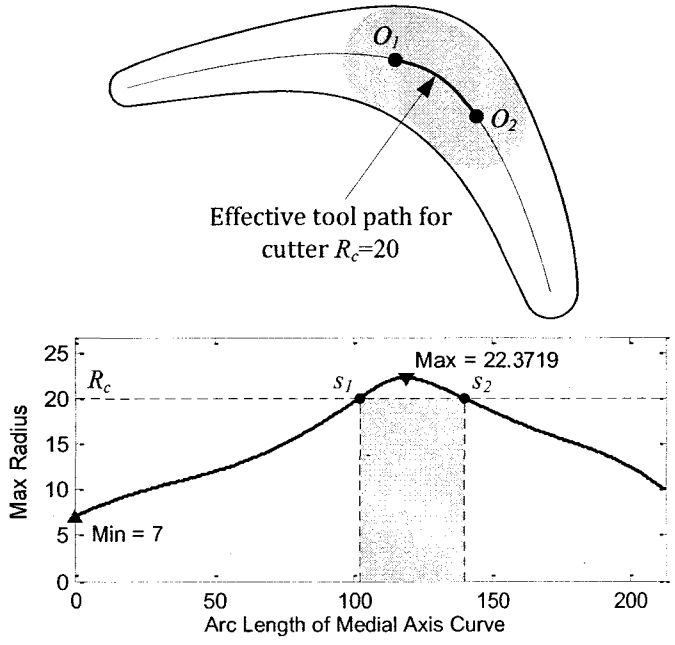


Figure 7.2 One tool path segment calculated from the radius function

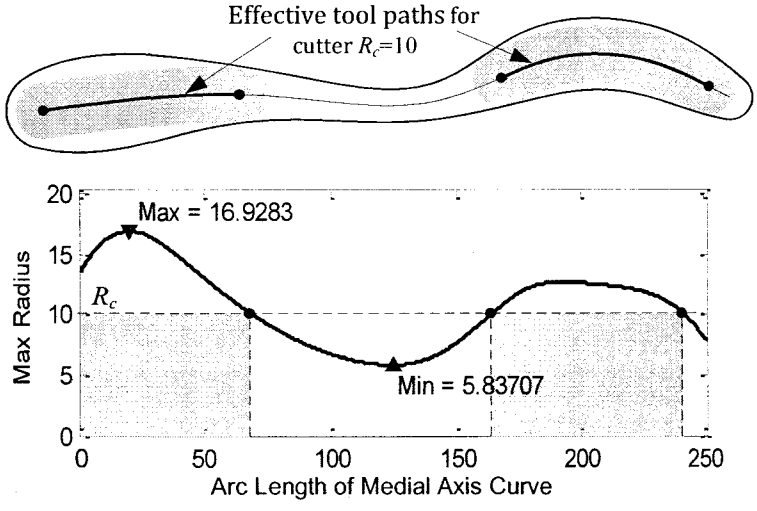


Figure 7.3 Two tool path segments calculated from the radius function

For a sharp corner or a round corner with radius smaller than the cutter radius, the cutter touches the pocket boundary before reaching the terminal point. If the contact circle radius R_t at a terminal point O is larger than the cutter radius R_c, after the cutter reaches this point, more stock material remains and can be cut by

moving the cutter forward, as shown in Figure 7.4. Here, the tool path is extended along its tangent direction at point O to point O', at which the cutter touches the pocket boundary. The length of the extra line segment is $d = R_t - R_c$.

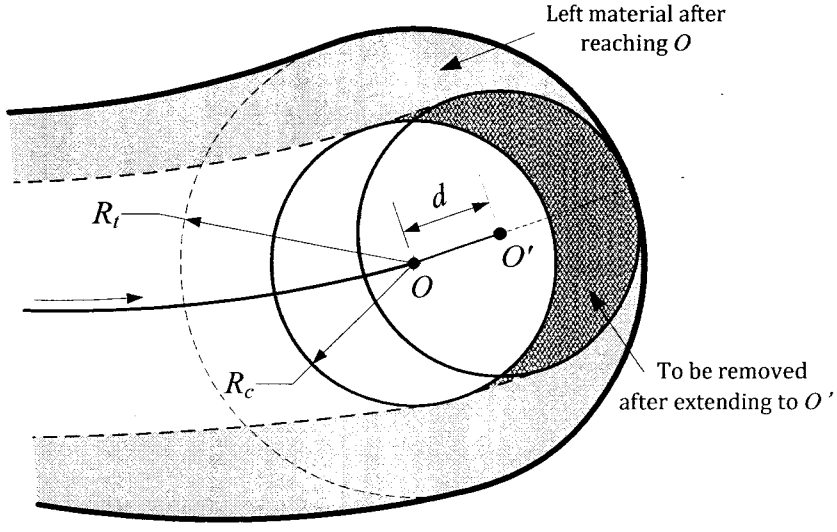


Figure 7.4 The extended tool path at a MA terminal point

In this work, multiple cutters are consecutively used from large to small in roughing a pocket. Usually, the tool paths of a larger cutter R_i directly connect with the tool paths of a smaller cutter R_{i+1}. During machining, the smaller cutter starts at the same point where the larger cutter stops. Due to the larger region covered by the larger cutter, the smaller cutter cannot remove stock material at beginning until it leaves the machined region. To get rid of the non-cutting tool path of the smaller cutter, a piece of the tool path of the smaller cutter is truncated. This method is illustrated in Figure 7.5. At the end O_i of the larger tool path, draw a circle with radius of $r = R_i - R_{i+1}$. This circle intersects the smaller tool path at point O_{i+1}. Thus, the smaller tool path starts at this point, and the path piece O_iO_{i+1} is eliminated.

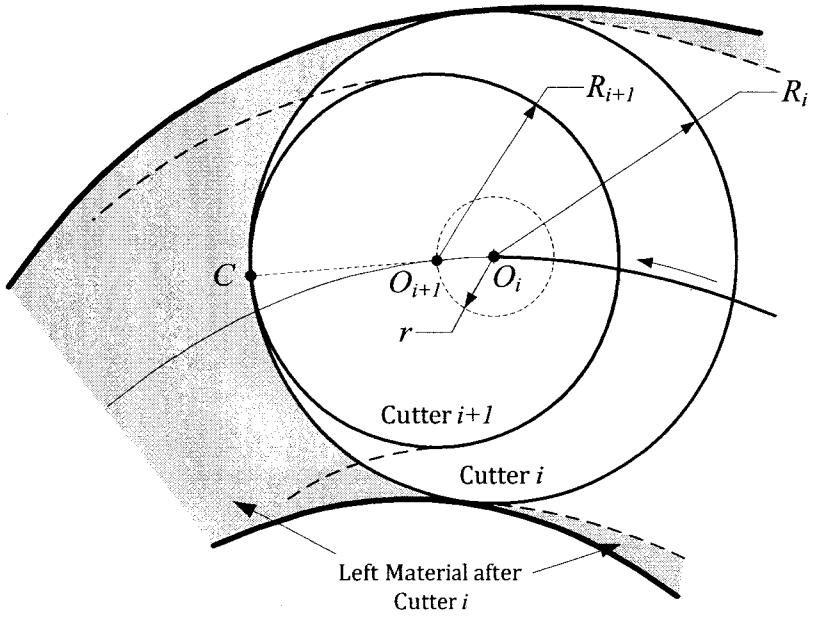


Figure 7.5 Diagram of tool paths adjustment at the starting point

7.2.3 Area of Machined Region and Thickness of Remaining Stock

The main objective of the pocket rough machining is to efficiently remove stock material as much as possible by using a reasonable number of large cutters. Also, it is often required that the thickness of remaining stock inside the pocket is within a specified value. To evaluate the volume of the material removed in roughing, an alternative way is to calculate the area of the region covered by the cutting tools. First, given an end-mill, its tool path can be found by using the approach introduced in the previous section. When the end-mill cuts the pocket along the path, the region covered by the tool is defined with a closed boundary, which comprises the path offsets on both sides by the tool radius and the semi-circles at the path ends and tangent to the offsets. Second, for multiple end-mills, their tool paths can be found. The region(s) covered by each cutter can be identified with their boundaries. Then, all the regions are merged with a Boolean operation -

union - for the whole machined region, thus, its boundary can be found. Finally, a well-established method is adopted to compute the area of a closed region according to its boundary. Meanwhile, the maximum thickness of the remaining excessive material in the pocket can be found by calculating the maximum distance between the boundaries of the pocket and the region covered by the cutters. The above procedure has been implemented on top of the OpenCASCADE geometric kernel, which provides geometric algorithms for computing area of close profiles and distance between curves.

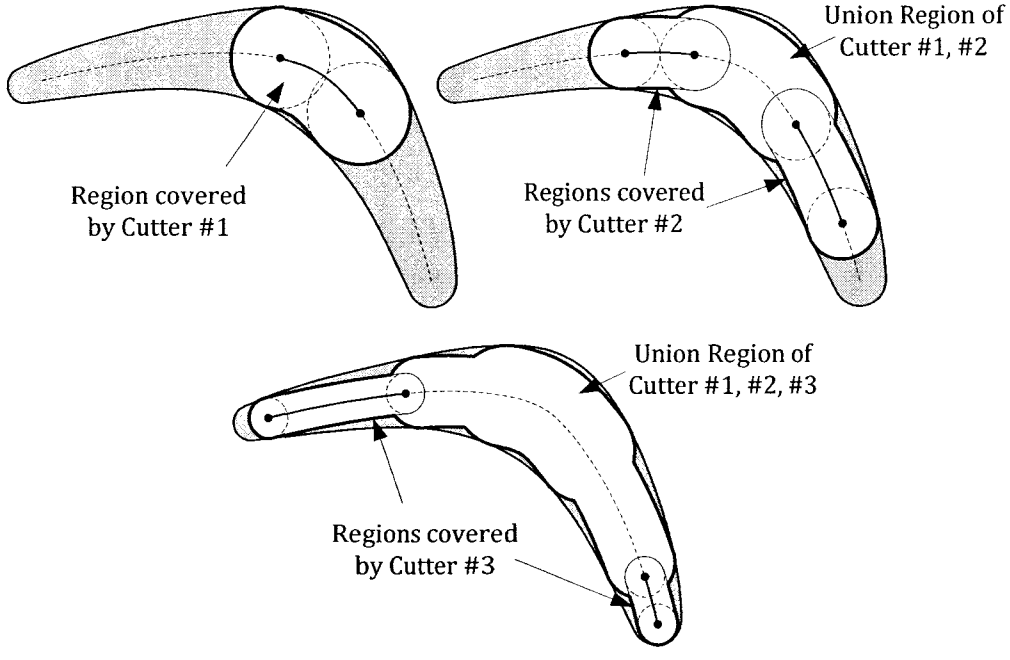


Figure 7.6 Region(s) covered by each cutting tool

7.3 Optimized Multiple Cutters for Aggressive Roughing

7.3.1 Optimization Problem of Multiple Cutters Selection

Due to the complex shape of the free-form pocket, the locally maximal contact circles of different MA points change dramatically, which means the tool accessible spaces at different tool locations on the path vary at large. Consequently, a large cutter cannot uniquely complete the pocket aggressive roughing. Although a small tool can access more regions inside the pocket and roughly cut the pocket, the cutting efficiency is significantly reduced. Therefore, to balance larger tools and various accessible pocket spaces, a good solution is to employ a reasonable number of optimal tools to efficiently rough the pocket without gouging and interference. In this research, the optimization problem of multiple cutters selection is defined in the following.

Suppose we are given a pocket profile and a library with N_T end-mills in different size that are available for machining the pocket. Referred with their radii R_i $(i=1,2,\cdots,N_{\mathrm{T}})$, the end-mills are listed in a decreasing order, i.e., $R_i > R_{i+1}$. From the cutter library, N cutters are selected to rough cut the pocket using the proposed aggressive roughing tool path strategy. The number of selected cutters N is properly determined so as to trade off the tool change time and the material removal volume. The maximal thickness of remaining stock in the pocket is specified, which is usually a small value. For the following finish cut, it is supposed to remove a little stock for high pocket accuracy. Therefore, the optimization problem is to find the best cutter sizes combination that will maximize the volume of stock material removed and

ensure that the thickness of the remaining stock is not greater than the specified maximum value.

The optimization problem of multiple cutters selection is a complicated global problem. The problem variables are the radii of the N end-mills and its objective function is not represented in simple formula. Thanks to all tools in the library being standard end-mills, the radii of the cutters for pocket roughing, which are selected from the library, are discrete variables. To find the optimal multiple cutters, by taking into account the features of this problem, the genetic algorithm method is adopted.

7.3.2 Basic of Genetic Algorithm

The genetic algorithm (GA) became a popular search technique through the work of John Holland [99] in the 1970s. The GA is a class of evolutionary algorithm inspired by natural genetic populations to evolve solutions to problems. There is no requirement to formulate a mathematical equation for the objective function. The basic idea is that a population of chromosomes, which represent candidate solutions to an optimization problem, evolves over time toward better solutions. A *chromosome* is usually represented in binary as strings of zeros or ones with a finite length. Each bit of the chromosome is called a *gene*. A *population* includes a selected number of chromosomes and the population at a given time is a *generation*. A *fitness function* is defined over the chromosome representation and measures the quality of the represented solution. The evolution usually starts from an initial population of randomly generated chromosomes. In each generation, the fitness of every

chromosome in the population is evaluated, and multiple chromosomes are stochastically selected based on their fitness to be modified by genetic operators in order to form a new population for the next iteration. Crossover, inversion and mutation are the three main genetic operators used for global searches. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

7.3.3 Genetic Algorithm Solver to Optimized Cutter Sizes Selection

In this work, a new technique of representing the discrete variables, the radii of the selected tools, is proposed. Since the N tools have to be selected from the standard end-mills in the library, a set of the variables values in the cutters selection model can be simplified and converted into a binary string, which refers to a chromosome in the GA. More specifically, because of the N_T standard end-mills arranged in a size decreasing order in the library, each chromosome is composed of a string of N_T genes. One gene uniquely represents one tool, and all the genes from left to right are mapped to all tools from large to small. Moreover, the gene value of a chromosome string is a binary, i.e., one or zero. For a tool in the library, if selected, its corresponding gene is one; otherwise, it is zero. Therefore, each chromosome consists of a string of N_T genes; N genes are one, and the rest are zero, e.g., $\left[\underbrace{00\cdots1\cdots1\cdots00}_{N_T}\right]^{N_T^R}$. For a given chromosome, by searching for genes with value of one, the selected cutters and their radii can be easily attained.

As a building block of solving the complex optimization problem, the algorithm of evaluating the objective function and the constraints is reiterated here. Based on a chromosome, the radii of the N selected end-mills can be found. After planning tool paths for these tools, the area of machined region and the maximum thickness of the remaining stock can be calculated. The area of the machined region is the value of the objective function, and the constraint is evaluated by comparing the calculated thickness against the specified maximum value. This algorithm is important, for it is repetitively employed in each searching iteration. Following the GA procedure, the eight steps in the process of solving the optimization problem of selecting multiple cutters are listed here.

1) According to the optimization problem, specify an appropriate *population size* of chromosomes, and the chromosomes of a generation are randomly created, each of which represents a set of cutters assumed to be selected and a candidate solution.
2) For each chromosome, the objective function value is calculated and the constraint is checked. Among all the chromosomes, the one with the maximum objective function value and subject to the constraint is the best in the generation.
3) The objective function values of this best chromosome and the best in the prior search process are compared, and if the best in the new generation is better, the best in the prior search is updated with the new best.

4) To create a new generation, based on the objective function values of the chromosomes, two chromosomes are selected with the *roulette wheel selection* method.
5) The crossover of the genes of these two chromosomes is randomly carried out according to the *crossover rate*.
6) The mutation is performed by going through all bits of the chosen chromosomes and flipping bits randomly according to the *mutation rate*.
7) Repeat from step 4) to 6) until a new population has been created.
8) Repeat from step 2) to 7), until meets one of the stopping criteria: a) a maximum number of generations have been produced; or b) best objective function value does not change over a number of generations, which is defined as *stall generations*.

The *population size*, *crossover rate*, and *mutation rate* are crucial to efficiency and convergence of the GA method; however, these parameters are dependent on different problems, and there is no fixed rule for them. In this work, the population size is 100, the crossover rate is 0.7, and the mutation rate is 0.001. It has been verified that with above parameters, the GA method can effectively solve the cutter selection optimization problem.

7.4 Illustrative Examples

The proposed tool path strategy and the GA based searching algorithm have been implemented on top of the developed MAT program using C++ programming

language. This section illustrates the proposed methodology using two examples. The first example demonstrates the roughing tool path strategy's capability of dealing with complex pocket shape; the second example shows the effectiveness of the multiple cutters selection algorithm using a general pocket part with islands.

7.4.1 Aggressive Roughing of a Complex Pocket Shape

The pocket part to be machined is modeled using CATIA V5 (Figure 7.7), which is the first character of the Chinese word of "Canada" or "machining". The pocket is to be carved in a workpiece of material 6061 T5 aluminum and size *101.6mm* by *101.6mm* by *50.8mm*. The machining is carried out on a DECKEL MAHO DMU 60T machining centre.

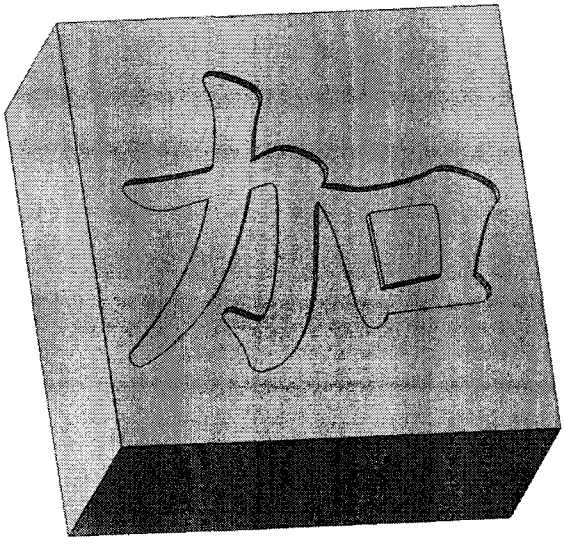
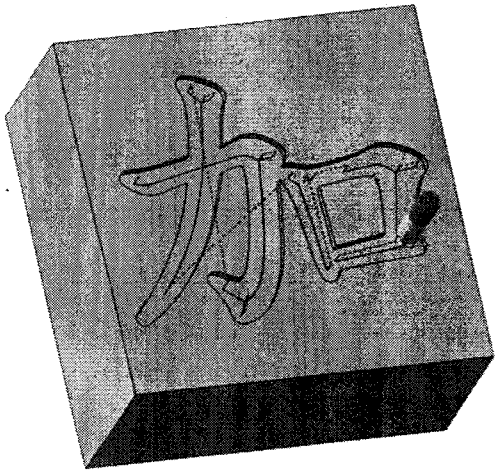


Figure 7.7 The CATIA model of a Chinese character

Due to the small pocket of the character and the small set of end mills available in the university machine shop, by using our approach, an end mill (*R3.086*) of radius *3.086mm* is selected for rough machining this character, and then an end

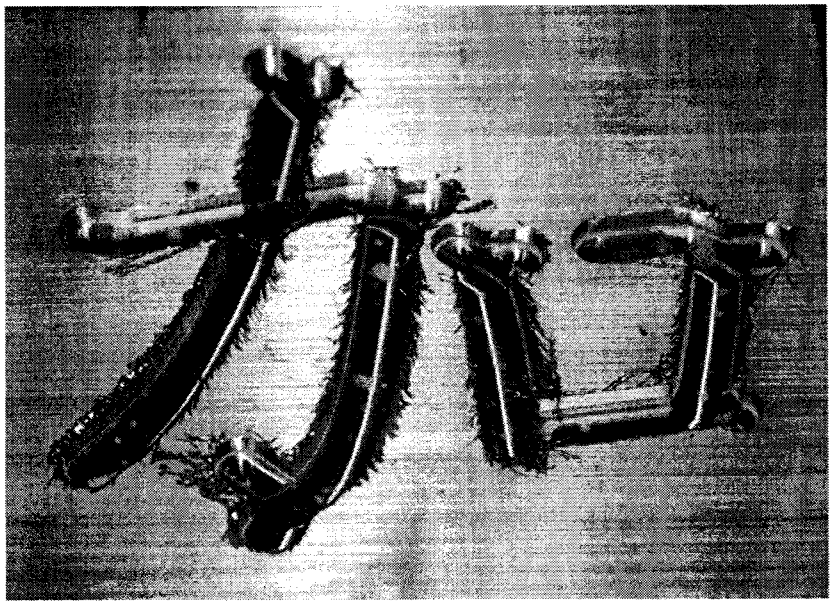
mill (*R1.5875mm*) of radius *1.5875mm* is selected to finish cut it. To conduct aggressive roughing, the medial axis tool paths for the end mill *R3.086* are generated and plotted in Figure 7.8(a), and the machining simulation is conducted with the result shown in Figure 7.8(b). The depth of cut is *2.54mm*, which is the depth of the pocket, and the feed rate is *254mm/min*. The roughing time is *30 seconds*, and the machined part is displayed in Figure 7.8(c), which complies with the simulation result. In finish machining, because of the small tool *R1.5875*, the depth of cut is *1.27mm*, a half of the depth of cut for *R3.086*; and the feed rate is *177.8mm/min*. In Figure 7.9, the paths for the end mill *R1.5875* to finish the pocket are plotted, and the finishing machining simulation result and the actual machined part are displayed. The finish machining time is *3 minutes*. Therefore, the total machining time is *3.5 minutes*.

To compare our approach to the conventional method, using one end mill of radius *1.5875mm* to cut the character, the tool paths of this cutter are planned using CATIA V5. Due to the small tool, the depth of cut is *1.27mm*, and the feed is *177.8mm/min*. In Figure 7.10, the tool paths are plotted, and the machining simulation result and the actual part after machining are displayed. The total machining time is *13 minutes*. It is evident that the machining time is *271%* longer than that using our new approach.



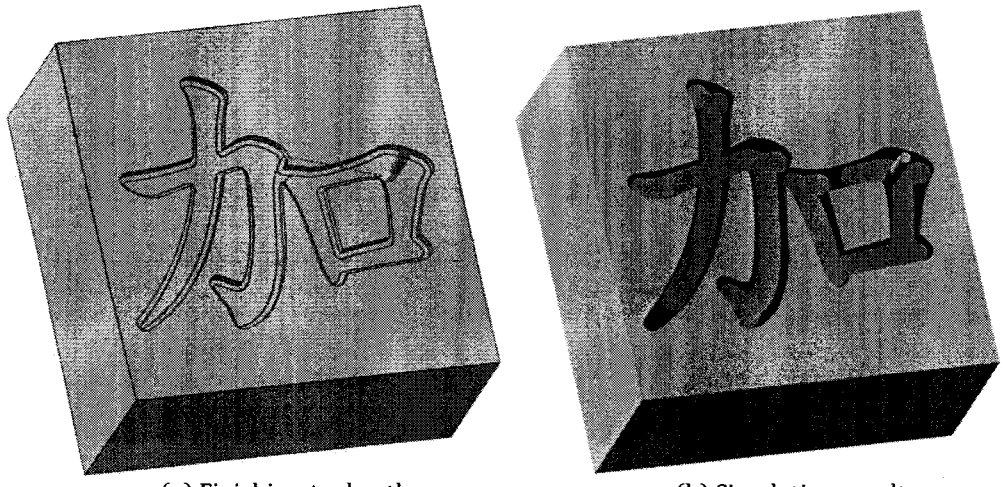
(a) Roughing tool paths

(b) Simulation result

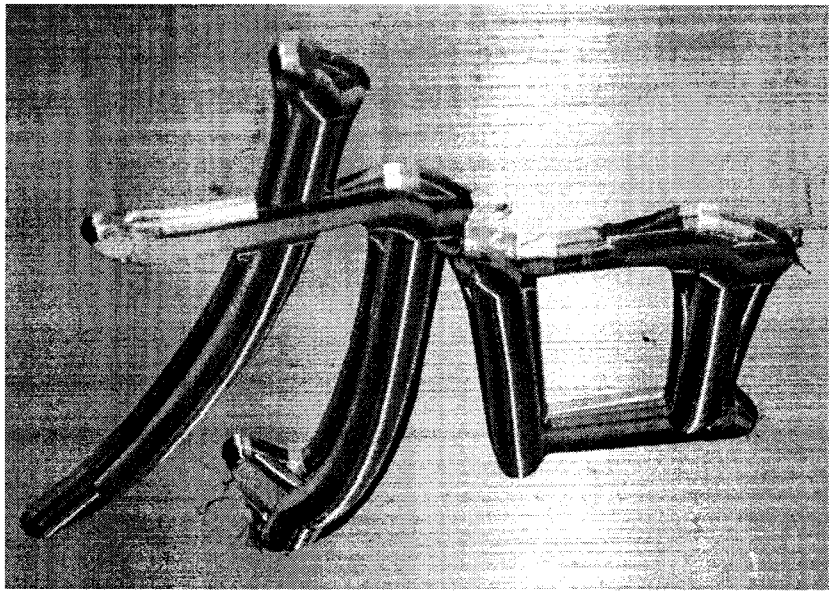


(c) Result of actual rough machining

Figure 7.8 Aggressive roughing with cutter R3.086mm

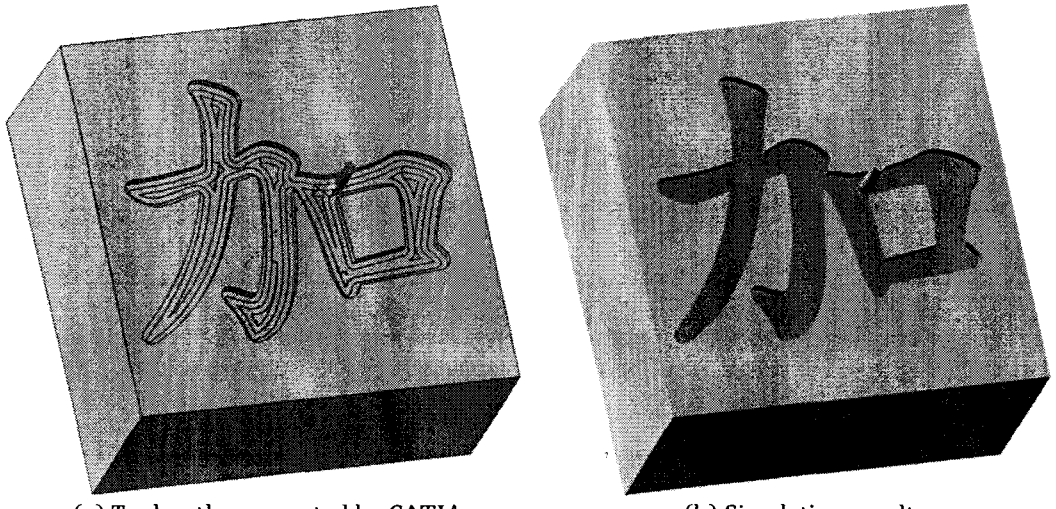


(a) Finishing tool paths (b) Simulation result

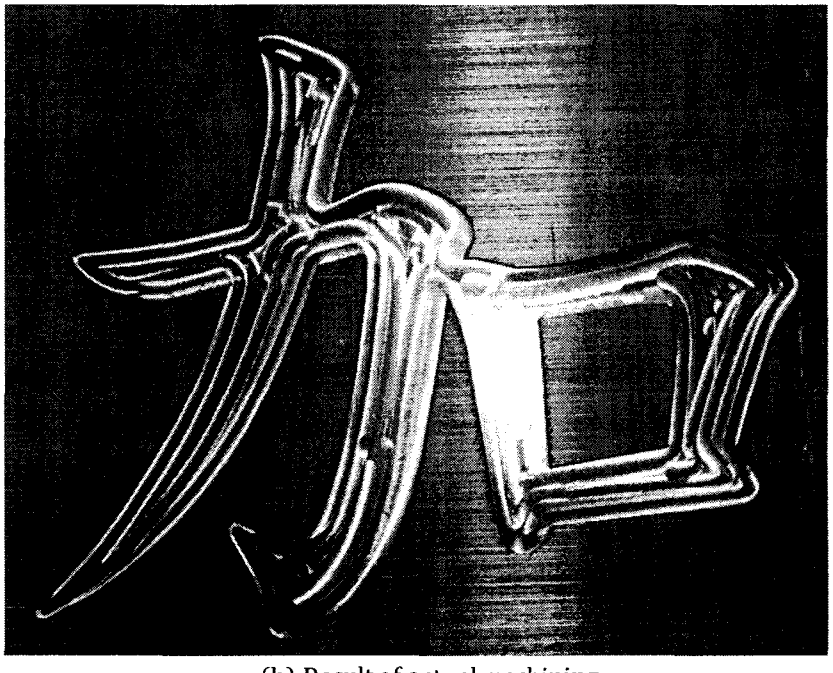


(c) Result of actual finish machining

Figure 7.9 Finish machining with cutter R1.5875mm



(a) Tool paths generated by CATIA (b) Simulation result



(b) Result of actual machining

Figure 7.10 Machining with cutter R1.5875mm using CATIA V5

7.4.2 Multiple Cutter Selection for Aggressive Roughing a Pocket

A typical mechanical part with a free-form pocket with two islands is designed using CATIA V5, see Figure 7.11. The pocket boundary consists of some lines, some circular arcs and a B-Spline curve.

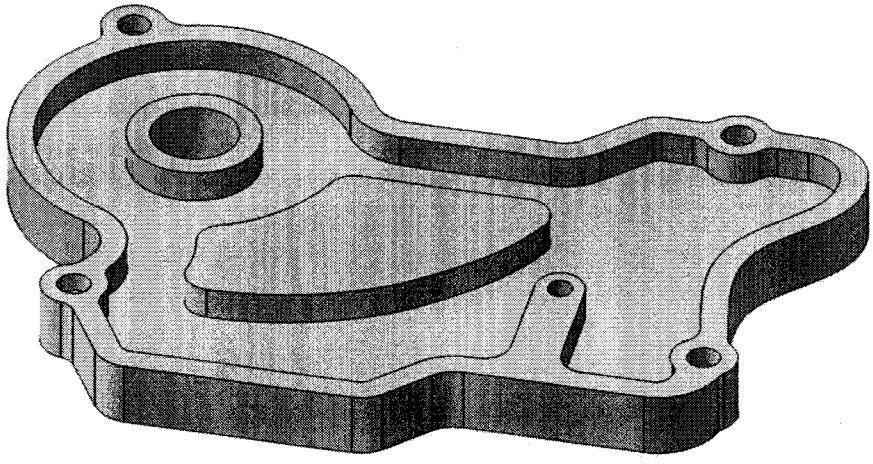


Figure 7.11 Mechanical part with a free-form pocket

By using the developed MAT program, the MAT of the pocket is calculated and shown in Figure 7.12. The reference tool paths and corresponding radius functions are obtained. The maximum and minimum contact circles can be found from the radius functions, and their radii are *30.86mm* and *4mm*, respectively, which are plotted in Figure 7.12.

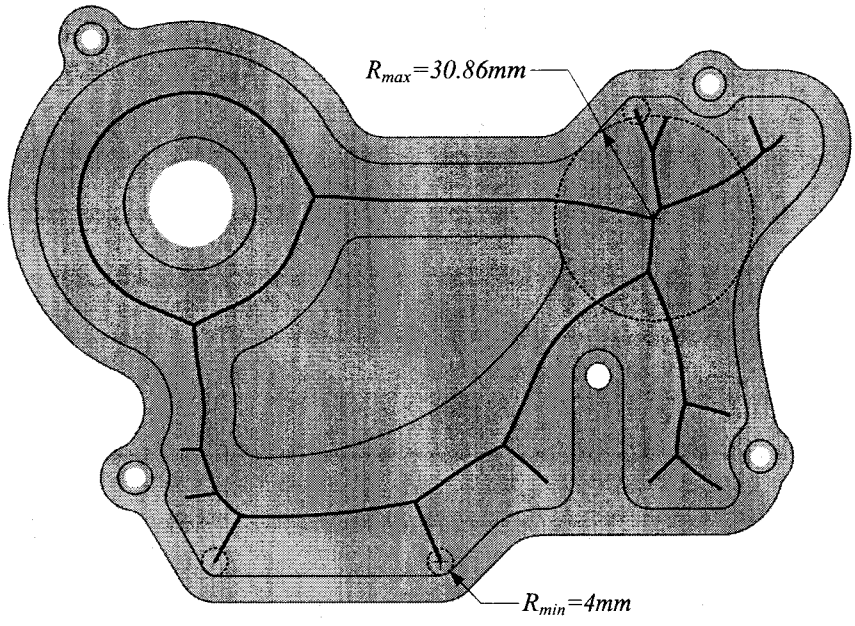


Figure 7.12 Medial axis of the free-form pocket with two islands

Assume the cutter library includes *17* cutters for roughing the pocket, and the radii are arranged in the descending order as *34, 32, 30, 28, 26, 24, 22, 20, 18, 16, 14, 12, 10, 8, 6, 4* and *2mm*. The developed GA based multiple cutters selection program is used to find the optimal set of cutter sizes.

Suppose two end mills will be used to roughly cut the pocket, by using our method, the cutters with radii of *18* and *10 mm* are the optimal solution of using two cutters in aggressive roughing. The tool path lengths of cutters *R18* and *R10* are *205.02* and *420.33 mm*, respectively. The maximum area covered by these cutters is *18821.39* mm^2. The tool paths and the areas covered by the tools are plotted in Figure 7.13. By using the conventional way, two cutters *R20* and *R8* are selected based on a NC programmer's experience. In the rough cut, the pocket area covered by the two cutters is *16846.42* mm^2 and the path length is *660.67 mm*, which is longer

than the path length of the two optimal cutters (see Table 7.1). Thus, the cut by using the optimal two cutters is *11.7%* more efficient, compared to the cut with the manually selected tools.

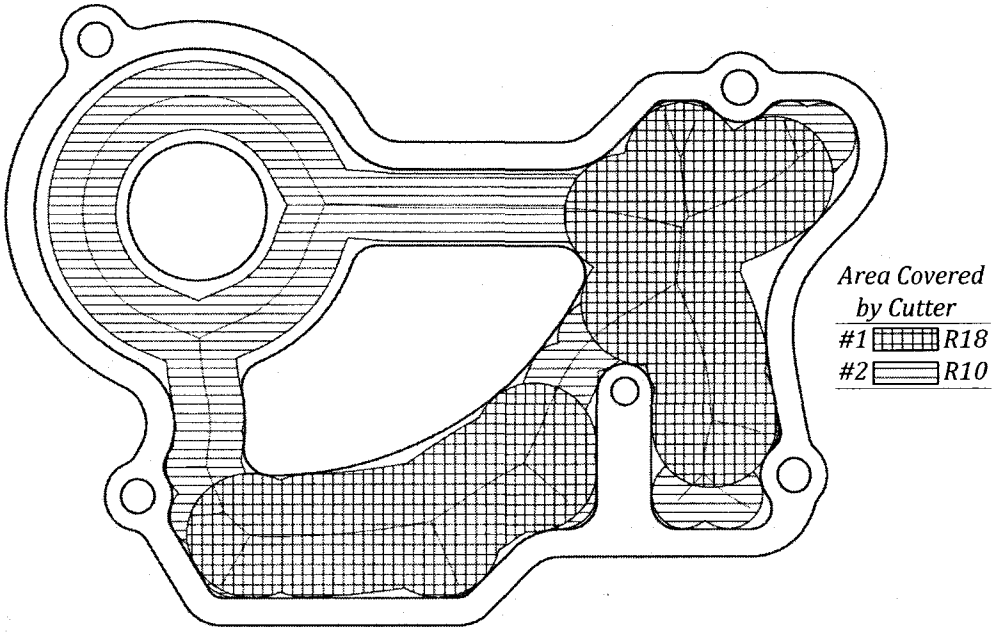


Figure 7.13 Tool paths and areas covered by the optimal two cutters

Table 7.1 Results of two cutters determined with the GA optimizer and the conventional way

Optimal method (2 cutters)	Tool path length (*mm*)	Tool covered area (mm^2)	Conventional way (2 cutters)	Tool path length (*mm*)	Tool covered area (mm^2)
#1 (R18)	205.02	10890.94	#1 (R20)	125.79	8819.92
#2 (R10)	420.33	7930.45	#2 (R8)	534.88	8026.5
Total	625.34	18821.39	Total	660.67	16846.42

If three end mills are to be used for aggressive roughing of the pocket, our method is applied, and the optimal cutters are tools with radii of *18*, *12* and *10 mm*. The tool path lengths of cutters *R18*, *R12* and *R10* are *205.02*, *320.54* and *95.79 mm*, respectively. The maximum area covered by these cutters is *20059.06* mm^2. The tool paths and the areas covered by the tools are plotted in Figure 7.14. In the conventional way, three cutters *R24*, *R16* and *R8* are subjectively selected. In the rough cut, the pocket area covered by the three cutters is *17705.89* mm^2 and the path length is *592.67 mm* (see Table 7.2). Although this path length is *4.89%* shorter than the path length of the three optimal cutters, the cut by using the optimal three cutters is *13.3%* more efficient, compared to the cut with the manually selected tools.

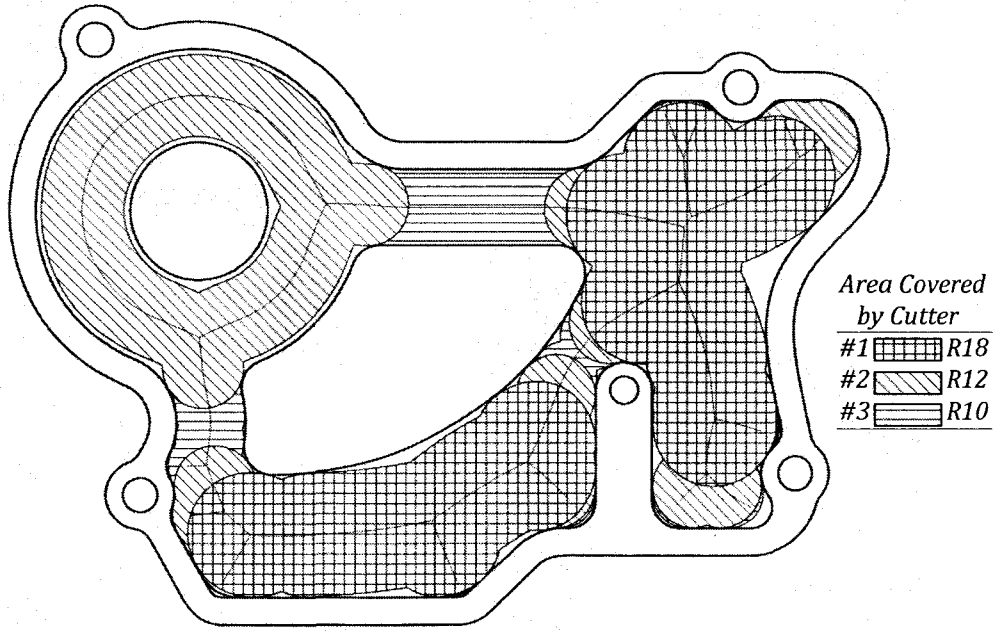


Figure 7.14 Tool paths and areas covered by the optimal three cutters

Table 7.2 Results of three cutters determined with the GA optimizer and the conventional way

Optimal method (3 cutters)	Tool path length (*mm*)	Tool covered area (mm^2)	Conventional way (3 cutters)	Tool path length (*mm*)	Tool covered area (mm^2)
#1 (R18)	205.02	10890.94	#1 (R24)	55.91	4052.67
#2 (R12)	320.54	7902.961	#2 (R16)	204.46	9288.74
#3 (R10)	95.79	1265.153	#2 (R8)	332.30	4364.47
Total	621.34	20059.06	Total	592.67	17705.89

If four end mills are to be used for aggressive roughing of the pocket, our method is applied, and the optimal cutters are tools with radii of *20, 16, 12* and *10 mm*. The tool path lengths of cutters *R20, R16, R12* and *R10* are *125.79, 130.58, 235.19* and *95.79mm*, respectively. The maximum area covered by these cutters is *20539.78mm*^{2}. The tool paths and the regions covered by the tools are plotted in Figure 7.15. If four cutters *R24, R16, R10* and *R8* are selected manually, the pocket region covered by the four cutters is *18905.32mm*^{2} and the total path length is *602.67mm*, which is longer than the path length of the four optimal cutters (see Table 7.3). It is evident that the cut by using the optimal four cutters is 8.6% more efficient, compared to the cut with the manually selected four tools.

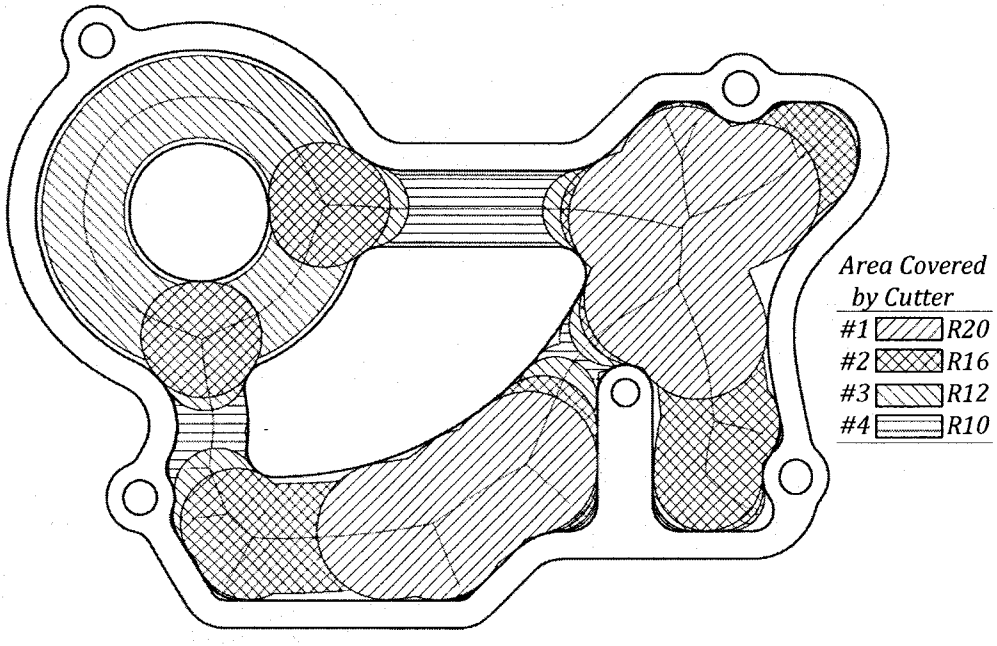


Figure 7.15 Tool paths and areas covered by the optimal four cutters

Table 7.3 Results of four cutters determined with the GA optimizer and the conventional way

Optimal method (4 cutters)	Tool path length (*mm*)	Tool covered area (mm^2)	Conventional way (4 cutters)	Tool path length (*mm*)	Tool covered area (mm^2)
#1 (R20)	125.79	8819.922	#1 (R24)	55.91	4052.67
#2 (R16)	130.58	5201.785	#2 (R16)	204.46	9288.75
#3 (R12)	235.19	5252.924	#3 (R10)	324.98	5520.73
#4 (R10)	95.79	1265.153	#4 (R8)	17.32	44.17
Total	587.34	20539.78	Total	602.67	18905.32

7.5 Summary

In this chapter, a new roughing tool path strategy for aggressive pocket roughing is proposed. The key of the proposed strategy is to use using large standard end-mills to remove stock material inside the pocket with full immerse. The MAT of free-form pocket can be found using the approach introduced in Chapter. Based on the MAT, the gouging free roughing paths are generated along the MA curves. This automatically generated tool path is especially suited for narrow and long pocket shapes. Then, a genetic algorithm based method is developed to optimize multiple cutters selected from a library of standard cutters in terms of the area covered by the cutters. This approach is practical and can be directly implemented in the existing CAD/CAM software in order to reduce roughing machining time in making complex pockets in the manufacturing industry.

The optimization problem discussed in this chapter is to maximize the volume of stock material removed for a fixed number of cutters. The total machining time is not considered. If the optimization goal is to minimize total roughing time while the number of cutter is also a variable, the following information will be required: optimal feed rate for each cutter, rapid motion speed, distance from workpiece to cutter changing position and cutter changing time. The GA based optimization method can still be applied to this new problem. The only modification is to evaluate the fitness value based on the total machining time, which can be calculated with all the input information.

Chapter 8 Conclusions and Future Work

In this research, new methods have been developed for geometric modeling of the 2½- and 3-axis milling process and constructing MAT of planar shapes with free-form curve boundaries. The major contributions of this research are summarized as follows:

- A layer based discrete geometric model is proposed to represent the in-process workpiece. A 2D profile is used to represent the shape of material left on the layer. By approximate the 3D workpiece model with a number of 2D layers, the 3D problem of extracting CWE geometry is solved in a series of 2D problems. The developed Z-layer B-Rep model can deal with complex CWE geometry in both 2½- and 3-axis milling. Both in-process model and extracted chip geometry is represented using the same model.
- Effective and practical algorithms are proposed for steady cut optimization and automatic feed rate selection for pocket machining. The proposed optimization methods are based on the spindle power graph generated by the developed 2½-axis milling process modeling system. The methodology is easy to implement and can be extended to include other machining parameter constraints.

- The geometric model of the 2D cutter swept envelope in 3-axis milling is proposed from the study of the basic mechanism of removing stock material on a horizontal layer. The mathematical formula of the profile is derived for APT cutter geometry and different tool motions. The approach is efficient and accurate to model the CWE in 3-axis milling, compared to the existing swept-volume methods.
- A new, efficient approach to approximate the MATs of free-form pockets with islands is developed. A mathematical model of the contact circle is derived and solved efficiently using a new hybrid optimization method. Based on the boundary tracing and decomposition scheme, the developed boundary tracing algorithm can handle free-form pockets with/without islands.
- A new roughing tool path strategy is proposed on top of the MAT. Gouging free full immersion tool path is generated along the MA curve. And a genetic algorithm based method is developed to optimize multiple cutters selected from a library of standard cutters in terms of the area covered by the cutters.

Computer implementation and experimental tests show the effectiveness of the developed methodologies. The techniques presented in this dissertation can be used to guide the selection of optimal cutting parameters for the CNC milling process.

For future research, following topics are suggested to expand the present research work:

- Developing methods for optimal cutting parameters selection for 3-axis sculptured surface milling;
- Conducting experimental verifications of 3-axis milling with general cutters (bull-nose or ball-end mills);
- Extending the geometric modeling methodology to virtual 5-axis milling with general cutter;
- Developing more applications of the MAT, a promising application is using MAT to generating plunge milling tool path;
- Integration of the proposed methods with CAD/CAM system.

Bibliography

[1] D Gibbs, TM Crandell, An introduction to CNC machining and programming, Industrial Press, New York, N.Y., 1991.

[2] EJA Armarego, RH Brown, The machining of metals, Prentice-Hall, Englewood Cliffs, N.J., 1969.

[3] MC Shaw, Metal cutting principles, Clarendon Press, Oxford, 1984.

[4] Y Altintas, Manufacturing automation :metal cutting mechanics, machine tool vibrations, and CNC design, Cambridge University Press, New York, 2000.

[5] H Zhang, ZC Chen, A new approach to determining optimum tool size for finish milling of nurbs profiles, 2006 (2006).

[6] ZC Chen, H Zhang. Optimal cutter size determination for 2-axis finish machining of NURBS profile parts, Int J Prod Res. 47 (2009) 6279-6293.

[7] H Martin , On the computational geometry of pocket machining, Springer-Verlag, Berlin; New York, 1991.

[8] G Elber, E Cohen, S Drake. MATHSM: Medial axis transform toward high speed machining of pockets, CAD Computer Aided Design. 37 (2005) 241-250.

[9] RR Martin, PC Stephenson. Sweeping of three-dimensional objects, Computer Aided Design. 22 (1990) 223-234.

[10] K Abdel-Malek, H Yeh. Geometric representation of the swept volume using Jacobian rank-deficiency conditions, Computer Aided Design. 29 (1997) 457-468.

[11] K Abdel-Malek, J Yang, D Blackmore. On swept volume formulations: Implicit surfaces, Computer Aided Design. 33 (2001) 113-121.

[12] D Blackmore, MC Leu, F Shih. Analysis and modelling of deformed swept volumes, Computer Aided Design. 26 (1994) 315-326.

[13] D Blackmore, MC Leu, LP Wang. The sweep envelope differential equation algorithm and its application to NC machining verification, Computer Aided Design. 29 (1997) 629-37.

[14] C- Chiou, Y- Lee. A shape-generating approach for multi-axis machining G-buffer models, Computer Aided Design. 31 (1999) 761-76.

[15] C- Chiou, Y- Lee. Swept surface determination for five-axis numerical control machining, Int.J.Mach.Tools Manuf. 42 (2002) 1497-1507.

[16] D Roth, S Bedi, F Ismail, S Mann. Surface swept by a toroidal cutter during 5-axis machining, Computer Aided Design. 33 (2001) 57-63.

[17] S Mann, S Bedi. Generalization of the imprint method to general surfaces of revolution for NC machining, Computer Aided Design. 34 (2002) 373-378.

[18] K Weinert, S Du, P Damm, M Stautner. Swept volume generation for the simulation of machining processes, Int.J.Mach.Tools Manuf. 44 (2004) 617-628.

[19] R Sarma. Flat-ended tool swept sections for five-axis NC machining of sculptured surfaces, Journal of Manufacturing Science and Engineering-Transactions of the Asme. 122 (2000) 158-165.

[20] ELJ Bohez, NTH Minh, B Kiatsrithanakorn, P Natasukon, H Ruei-Yun, LT Son. The stencil buffer sweep plane algorithm for 5-axis CNC tool path verification, Computer Aided Design. 35 (2003) 1129-1142.

[21] SK Gupta, SK Saini, BW Spranklin, Z Yao. Geometric algorithms for computing cutter engagement functions in 2.5D milling operations, Computer Aided Design. 37 (2005) 1469.

[22] AD Spence, Y Altintas. A solid modeller based milling process simulation and planning system, Transactions of the ASME.Journal of Engineering for Industry. 116 (1994) 61-9.

[23] D Yip-Hoi, X Huang. Cutter/workpiece engagement feature extraction from solid models for end milling, J.Manuf.Sci.Eng.Trans.ASME. 128 (2006) 249-260.

[24] H El Mounayri, AD Spence, MA Elbestawi. Milling process simulation - a generic solid modeller based paradigm, J.Manuf.Sci.Eng.Trans.ASME. 120 (1998) 213-221.

[25] BM Imani, MH Sadeghi, MA Elbestawi. Improved process simulation system for ball-end milling of sculptured surfaces, Int.J.Mach.Tools Manuf. 38 (1998) 1089-1107.

[26] HB Voelcker, WA Hunt. Role of solid modelling in machining-process modelling and NC verification, SAE Preprints. (1981).

[27] BK Choi, RB Jerard, Sculptured surface machining :theory and applications, Kluwer Academic, Dordrecht; London, 1998.

[28] W Yun, D Cho. Accurate 3-D cutting force prediction using cutting condition independent coefficients in end milling, Int.J.Mach.Tools Manuf. 41 (2001) 463-478.

[29] GM Kim, PJ Cho, CN Chu. Cutting force prediction of sculptured surface ball-end milling using Z-map, Int.J.Mach.Tools Manuf. 40 (2000) 277-291.

[30] R Zhu, SG Kapoor, RE DeVor, Mechanistic Modeling of the Ball End Milling Process for Multi-Axis Machining of Free-Form Surfaces, J. Manuf. Sci. Eng. 123 (2001) 369-379.

[31] M Martellotti. An analysis of the milling process, Trans.ASME. 63 (1941) 677–700.

[32] SD Merdol, Y Altintas. Virtual simulation and optimization of milling operations - Part I: Process simulation, J.Manuf.Sci.Eng.Trans.ASME. 130 (2008) 0510041-05100412.

[33] SD Merdol, Y Altintas. Virtual simulation and optimization of milling applications - Part II: Optimization and feedrate scheduling, J.Manuf.Sci.Eng.Trans.ASME. 130 (2008) 0510051-05100510.

[34] P Lee, Y Altintas. Prediction of ball-end milling forces from orthogonal cutting data, Int.J.Mach.Tools Manuf. 36 (1996) 1059-1072.

[35] F Koenigsberger, AJP Sabberwal. An investigation into the cutting force pulsations during milling operations, International Journal of Machine Tool Design and Research. 1 (1961) 15-33.

[36] WA Kline, RE DeVor, JR Lindberg. The prediction of cutting forces in end milling with application to cornering cuts, International Journal of Machine Tool Design and Research. 22 (1982) 7-22.

[37] I Yellowley. Observations on the mean values of forces, torque and specific power in the peripheral milling process, International journal of machine tool design & research. 25 (1985) 337-346.

[38] S Engin, Y Altintas. Mechanics and dynamics of general milling cutters. Part I: Helical end mills, Int.J.Mach.Tools Manuf. 41 (2001) 2195-2212.

[39] H Feng, C Menq. Prediction of cutting forces in the ball-end milling process-1. model formulation and model building procedure, Int.J.Mach.Tools Manuf. 34 (1994) 697-710.

[40] H Feng, C Menq. Prediction of cutting forces in the ball-end milling process-II. cut geometry analysis and model verification, Int.J.Mach.Tools Manuf. 34 (1994) 711-719.

[41] A Azeem, H Feng, L Wang. Simplified and efficient calibration of a mechanistic cutting force model for ball-end milling, Int.J.Mach.Tools Manuf. 44 (2004) 291-298.

[42] J Gradiek, M Kalveram, K Weinert. Mechanistic identification of specific force coefficients for a general end mill, Int.J.Mach.Tools Manuf. 44 (2004) 401-414.

[43] M Wan, WH Zhang, GH Qin, G Tan. Efficient calibration of instantaneous cutting force coefficients and runout parameters for general end mills, International Journal of Machine Tools & Manufacture. 47 (2007) 1767-76.

[44] Y Koren, CC Lo, M Shpitalni, CNC interpolators: algorithms and analysis, American Society of Mechanical Engineers, Production Engineering Division (Publication) PED. 64 (1993) 83.

[45] DCH Yang, T Kong. Parametric interpolator versus linear interpolator for precision CNC machining, Computer Aided Design. 26 (1994) 225-34.

[46] Dong-Il Kim, Study on interpolation algorithms of CNC machine tools, 3 (1995) 1930-7.

[47] QG Zhang, RB Greenway. Development and implementation of a NURBS curve motion interpolator, Robot.Comput.Integrated Manuf. 14 (1998) 27.

[48] T Yong, R Narayanaswami. A parametric interpolator with confined chord errors, acceleration and deceleration for NC machining, Computer Aided Design. 35 (2003) 1249-59.

[49] Siemens, Simens 840D Program Manual,.

[50] GE Fanuc, GE Fanuc Series 16i/ 18i/ 160i/ 180i - Operator's Manual,.

[51] H Blum, A transformation for extracting new descriptors of shape. Models for the Perception of Speech and Visual Form, 243 (1967) 246.

[52] EC Sherbrooke, NM Patrikalakis, F Wolter. Differential and topological properties of medial axis transforms, Graphical Models Image Process. 58 (1996) 574-592.

[53] HI Choi, SW Choi, HP Moon. Mathematical Theory Of Medial Axis Transform, Pacific J.Math. 181 (1997) 57-88.

[54] T Culver, J Keyser, D Manocha. Accurate computation of the medial axis of a polyhedron, Proceedings of the Symposium on Solid Modeling and Applications. (1999) 179-190.

[55] T Culver, J Keyser, D Manocha. Exact computation of the medial axis of a polyhedron, Comput.Aided Geom.Des. 21 (2004) 65-98.

[56] F Aurenhammer. Voronoi diagrams-a survey of a fundamental geometric data structure, Computing Surveys. 23 (1991) 345-405.

[57] J O'Rourke, Computational geometry in C, 2nd ed., Cambridge University Press, Cambridge, UK,; New York, NY, USA, 1998.

[58] A Fabri, Voronoi diagrams in CGAL, the Computational Geometry Algorithms Library, (2007) 8-11.

[59] JW Brandt. Convergence and continuity criteria for discrete approximations of the continuous planar skeleton, CVGIP.Image understanding. 59 (1994) 116-124.

[60] TK Dey, W Zhao. Approximating the medial axis from the Voronoi diagram with a convergence guarantee, Algorithmica (New York). 38 (2003) 179-200.

[61] TK Dey, W Zho. Approximate medial axis as a Voronoi subcomplex, CAD Computer Aided Design. 36 (2004) 195-202.

[62] JJ Chou. Voronoi diagrams for planar shapes, IEEE Comput.Graphics Appl. 15 (1995) 52-59.

[63] R Ramamurthy, RT Farouki. Voronoi diagram and medial axis algorithm for planar domains with curved boundaries. I. Theoretical foundations, J.Comput.Appl.Math. 102 (1999) 119-141.

[64] R Ramamurthy, RT Farouki. Voronoi diagram and medial axis algorithm for planar domains with curved boundaries - II: Detailed algorithm description, J.Comput.Appl.Math. 102 (1999) 253-277.

[65] H Alt, O Cheong, A Vigneron. The Voronoi diagram of curved objects, Discrete & Computational Geometry. 34 (2005) 439-53.

[66] H Nebi Gursoy, NM Patrikalakis. An automatic coarse and fine surface mesh generation scheme based on medial axis transform. I. Algorithms, Engineering with Computers. 8 (1992) 121-37.

[67] H Nebi Gursoy, NM Patrikalakis. An automatic coarse and fine surface mesh generation scheme based on medial axis transform. II. Implementation, Engineering with Computers. 8 (1992) 179-96.

[68] M Sabry Hassouna, AA Farag, Robust centerline extraction framework using level sets, I (2005) 458-467.

[69] HI Choi, SW Choi, HP Moon, N Wee. New algorithm for medial axis transform of plane domain, Graphical Models Image Process. 59 (1997) 463-483.

[70] A Hatna, RJ Grieve, P Broomhead. Automatic CNC milling of pockets: geometric and technological issues, Computer Integrated Manufacturing Systems. 11 (1998) 309-330.

[71] D Veeramani, Y- Gau. Selection of an optimal set of cutting-tools for a general triangular pocket, Int J Prod Res. 35 (1997) 2621-2637.

[72] D Veeramani, Y Gau. Selection of an optimal set of cutting-tool sizes for 2 1/2 D pocket machining, Computer Aided Design. 29 (1997) 869-877.

[73] S Hinduja, D Sandiford. An Optimum Two-tool Solution for Milling 2½D Features from Technological and Geometric Viewpoints, CIRP Ann.Manuf.Technol. 53 (2004) 77-80.

[74] I Nadjakova, S McMains, Finding an optimal set of cutter radii for 2D pocket machining, 15 (2004) 351-358.

[75] H Li, Z Dong, GW Vickers. Optimal toolpath pattern identification for single island, sculptured part rough machining using fuzzy pattern analysis, CAD Computer Aided Design. 26 (1994) 787-795.

[76] Z Yang, A Joneja, S Zhu. Recognizing generalized pockets for optimizing machining time in process planning - Part 1, Int J Prod Res. 39 (2001) 3377-3397.

[77] Z Yang, A Jonej, S Zhu. Recognizing generalized pockets for optimizing machining time in process planning - Part 2, Int J Prod Res. 39 (2001) 3601-3621.

[78] A Joneja, Y Weifeng, Y Lee. Greedy tool heuristic approach to rough milling of complex shaped pockets, IIE Transactions (Institute of Industrial Engineers). 35 (2003) 953-963.

[79] T Lim, JR Corney, DER Clark. Exact tool sizing for feature accessibility, Int J Adv Manuf Technol. 16 (2000) 791-802.

[80] T Lim, J Corney, JM Ritchie, DER Clark. Optimizing tool selection, Int J Prod Res. 39 (2001) 1239-1256.

[81] Z Yao, SK Gupta, DS Nau, A geometric algorithm for selecting optimal set of cutters for multi-part milling, Proceedings of the Symposium on Solid Modeling and Applications. (2001) 130.

[82] Z Yao, SK Gupta, DS Nau. Algorithms for selecting cutters in multi-part milling problems, Computer Aided Design. 35 (2003) 825.

[83] Y Zhang, Y Li, New approach to selecting multiple tools for milling 2.5-D pockets, (2007) 2320-2325.

[84] R Narayanaswami, Y Choi. NC machining of freeform pockets with arbitrary wall geometry using a grid-based navigation approach, Int J Adv Manuf Technol. 18 (2001) 708-716.

[85] J- Shih, S-F Chuang. NURBS output based tool path generation for freeform pockets, Int J Adv Manuf Technol. 29 (2006) 714-721.

[86] RM D'Souza, C Sequin, PK Wright. Automated tool sequence selection for 3-axis machining of free-form pockets, CAD Computer Aided Design. 36 (2004) 595-605.

[87] F E Gerald, Curves and surfaces for CAGD :a practical guide, 5th ed., Morgan Kaufmann, San Francisco, CA, 2002.

[88] K Lee, Principles of CADCAMCAE systems, Addison-Wesley, Reading, Mass., 1999.

[89] JJ Childs, Numerical control part programming, Industrial Press, New York, 1973.

[90] JJ Stoker, Differential geometry, Wiley-Interscience, New York, 1969.

[91] J Kennedy, R Eberhart, Particle swarm optimization, 4 (1995) 1942-1948.

[92] D Bratton, J Kennedy, Defining a standard for particle swarm optimization, (2007) 120-127.

[93] Q Fu, ZC Chen, Medial Axis Transform of Planar Shapes with Free-form Curve Boundary, Proceedings of the ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference. 2010 (2010).

[94] RJ Sharpe, RW Thorne. Numerical method for extracting an arc length parameterization from parametric curves, Comput.-Aided Des. 14 (1982) 79-81.

[95] B Guenter, R Parent. Computing the arc length of parametric curves, IEEE Comput.Graphics Appl. 10 (1990) 72-78.

[96] OpenCASCADE, http://www.opencascade.org,.

[97] CM Hoffmann. Problems of accuracy and robustness in geometric computation, Computer. 22 (1989) 31-39, 41.

[98] C Li, S Pion, CK Yap. Recent progress in exact geometric computation, Journal of Logic and Algebraic Programming. 64 (2005) 85-111.

[99] JH Holland, Adaptation in natural and artificial systems :an introductory analysis with applications to biology, control, and artificial intelligence, University of Michigan Press, Ann Arbor, 1975.