Multi-Point Machining of Sculptured Surfaces

by

Andrew Warkentin

A thesis presented to the
University of Waterloo in
fulfillment of the thesis
requirement for the degree of
Doctor of Philosophy
in
Mechanical Engineering

Waterloo, Ontario, Canada, 1997

0-612-22247-0

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Abstract

Sculptured surfaces are common in a wide variety of products such as automobiles, household appliances, water craft and aircraft components. These surfaces may be required to reduce fluid drag or simply for esthetics. Machining is used in the production of most of these surfaces. Present day surface machining techniques generate the design surface at a single point on the cutting tool. Many closely spaced passes of the cutter are required to machine a surface to the required tolerance. In this thesis, a new 5-axis technique for machining sculptured surfaces with a toroidal or flat bottom end mill is presented. The proposed Multi-Point Machining tool positioning strategy advocates generating a surface at more than one location on the tool. More contact points between the surface and the tool will result in faster machining.

Equations were developed to model multi-point contact between the workpiece surface and the tool. They were used to study the nature of contact between the tool and five quadratic surface forms: planar, parabolic, spherical, elliptic and hyperbolic. Two basic types of multi-point contact were found to exist; a circle of contact for planer and spherical surfaces and 2-point contact for parabolic, elliptic and hyperbolic surfaces. These results formed the basis of an exact multi-point tool positioning strategy for quadratic surfaces. Two algorithms were also developed to modify a multi-point tool position for non-quadratic surfaces.

Cutting tests and simulations were used to investigate the properties of the multi-point tool positioning strategy. Multi-point scallops were found to be low and wide, giving the machined surface a much smoother appearance than those produced by competing techniques. The size of these scallops were controlled by the tool geometry, separation between cutter contact points and feed direction. A big tool with small inserts will produce smaller scallops than a small tool with large inserts. The best feed direction is the direction of minimum curvature and the worst feed direction is the direction of maximum curvature.

Multi-point machining clearly offers the best performance of any tool positioning strategy. The results in this thesis show that the scallop heights produced by multi-point machining are about 0.0025, 0.04 and 0.5 times as large as those produced by the other leading tool positioning strategies: ball nosed, inclined tool and principal axis methods.

Acknowledgment

I would like to thank my supervisors, Professor Sanjeev Bedi and Professor Fathy Ismail for their support, enthusiasm and guidance. I also thank, Paul Hoskins and Neil Rao for their encouragement, friendship and help throughout this project. I thank the Natural Sciences and Engineering Research Council (NSERC), Ontario Graduate Scholarship Program (OGS) and the University of Waterloo for providing financial support. I would also like to thank Marius Van Reenan and John Bolt for their efforts with the retrofitting of the Rambaudi milling machine. I thank Mr. Ian Hunter and Mr. P. Marmion of IRDI, Professor R. Buchal and Mr. Marion of the University of Western Ontario and Professor Elbestawi at McMaster University for the timely use of their 5-axis machines.

Most of all, I wish to thank my family who have made all of this possible: my mother and father, Jacqueline and Jack Warkentin, my sister Janet and my fiancée Joni.

Table of Contents

Chapter 4

Multi-Point Contact With a Toroidal End Mill 68

Chapter 5

Implementation of Multi-Point Machining 89

Chapter 6

Algorithms for Tool Position Adjustment 107

Chapter 7

Chapter 8

List of Figures

List of Tables

Chapter 1

Introduction

The application of 5-axis NC machining has increased dramatically in the past ten years [1]. Many companies are switching to 5-axis technology to reduce manufacturing cost, increase accuracy, and decrease machining time.

However, the full benefit of 5-axis technology has yet to be realized in the area of sculptured surface machining. These surfaces are used in the design of car bodies, ship hulls, molds, dies and other application where smooth surfaces are needed to reduce fluid resistance or simply for esthetics. Coons patches, Bézier, B-splines, and other methods have been invented to represent these surfaces. Most commercial Computer-Aided Design, CAD, packages have a sculptured surface design capability. The result is an ever increasing demand to manufacture products featuring sculptured surfaces. These surfaces are typically produced directly by NC machines as in the case of turbine blades, or they are produced by tools manufactured on NC machines. By far the largest demand for surface machining occurs in the mold and die industry where the tools used to manufacture everything from beverage containers to automobile parts are produced. The importance of this industry can not be over-emphasized. In most countries, the mold and die industry consists primarily of small companies with fewer than 100 employees. These small job shops make a substantial contribution to the economy. In the United States alone, approximately 15,000 die shops represent an annual sales volume of about $20 billion [11].

Sculptured surfaces are generally produced in three stages: roughing, finishing and benchwork. Roughing cuts are used to remove most of the material from a workpiece while leaving the part slightly oversized. Finish machining of a sculptured surface removes as much as possible of the remaining material from the roughed out workpiece and attempts to machine the part to its final dimensions. The resulting surface is left with a large number of scallops, as shown in Figure 1.1. Benchwork consisting of grinding and polishing is used to remove these scallops. The time spent on finishing and benchwork is dependent on the size of these scallops. A recent survey by LeBlond Makino of Mason, Ohio [2], stated that a small mold will typically require 57 hours of roughing, 127 hours of finishing and 86 hours of grinding and polishing. Over 78 % of the total production time is spent on finishing, grinding and polishing

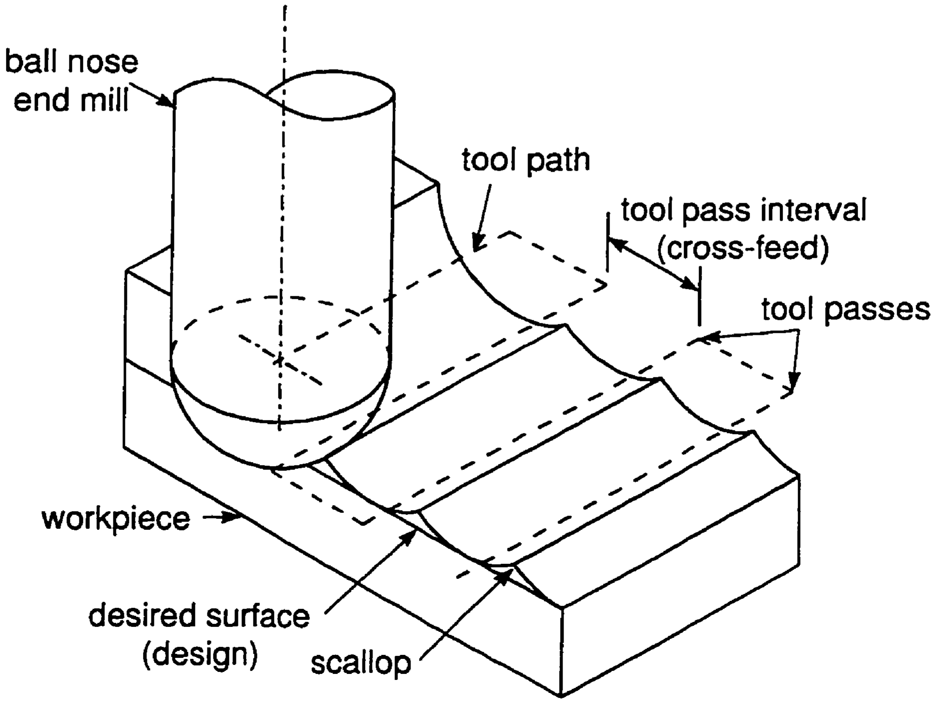


Figure 1.1 Scallops left after machining.

The bottleneck in the production of sculptured surfaces occurs at the finishing stage. Faster techniques for finishing that produce smaller scallops would reduce the amount of time spent on both finishing and benchwork.

The tool path used for finishing a surface is generally produced in three stages. First, tool path planning is used to determine the path the tool will take as it machines a surface. The tool path generally consists of a number of parallel tool passes. The distance between the tool passes is called the tool pass interval or cross-feed. Second, tool positioning strategies are used to determine the cutter position relative to the workpiece at a specific point on the tool path. The objective of a tool positioning strategy is to minimize the material remaining between the tool and the design surface as the tool moves along the tool path. Finally gouge detection and correction algorithms are used to determine if the tool has penetrated the desired surface. The tool position can then be modified to eliminate the undesired penetration.

Scallop size is related to the tool size, the tool pass interval and the tool positioning strategy. In general, bigger tools will produce smaller scallops. It is best to use the largest available tool that will not gouge the workpiece. Reducing the tool pass interval will decrease the scallop size at the expense of increased machining and benchwork time. Only small productivity gains can be expected by further optimization of the time spent on finishing versus the time spent on benchwork. On the other hand, an improved tool positioning strategy offers the possibility of reducing scallop size without increasing machining time. The goal of this work is to develop such a tool positioning strategy.

All present 3-axis and 5-axis tool positioning strategies position the cutting tool such that a single point on the tool contacts the design surface. The proposed strategy called ***Multi-Point Machining***, **MPM**, significantly reduces scallop heights without sacrificing machining time by generating the surface at more than one point on the tool.

1.1 Research Goals and Contributions

The goal of this work was to develop a 5-axis tool positioning strategy that would significantly reduce the time required for finish machining without sacrificing surface finish. In order to achieve this goal, a number of research contributions were made.

The proposed multi-point tool positioning strategy reduced scallops by improving the match between the tool and the surface. Tool-surface matching was accomplished by placing as many points on the tool in contact with the surface as possible. However, before the multi-point tool positioning strategy could be developed, an investigation into the nature of multi-point contact was required. Prior to this work, there had been no attempt to model multi-point contact between a tool and a design surface. The first and most important objective of this work was to develop a system of equations for modeling the contact between a tool and a design surface. These equations did not involve any assumptions on the number of contact points between the tool and the design surface. They were used to study the configuration and number of contact points between the tool and a typical design surface. This information formed the basis of the multi-point tool positioning strategy.

Once the nature of contact between the tool and the design surface was understood the Multi-Point Machining tool positioning strategy could be developed. Theoretically, the new tool positioning strategy could have been developed based entirely on the equations of contact. However, these equations are exceedingly complex and time consuming to solve. A CAM package using these equations would require days to generate a tool path instead of minutes. To circumvent this computational problem, the multi-point tool positioning strategy was developed based on the properties of tool surface contact discovered using the equations of tool surface contact.

Having demonstrated that multi-point tool positions could be generated, a study was undertaken to determine the strengths and weaknesses of the proposed technique and to compare it with the existing tool positioning strategies. This evaluation was unique because it compared the proposed technique to the most advanced 5-axis tool positioning techniques not just the inferior 3-axis techniques used as a benchmark in other studies.

To put Multi-Point Machining in context, a comprehensive review of the issues in 5-axis machining was undertaken. Most papers concentrate on one or two issues in 5-axis machining. This review identified a number of important research questions in 5-axis surface machining. Tool positioning was seen as the area where the most substantial improvements could be made.

Multi-Point Machining was initially applied to the machining of spherical surfaces in order to prove the concept. The special geometry of the spherical surface made Multi-Point Machining relatively easy to implement and exceedingly effective.

The experimental work for this thesis was performed in four different locations in Ontario: The Industrial Research and Development Institute (IRDI) in Midland, McMaster University in Hamilton, The University of Western Ontario in London, and the University of Waterloo. The configuration of the 5-axis machines at each of these locations were different. Each of these machines required a different post-processor to account for the different machine configurations and simulator to test the experimental tool path. Software development was aided by the development of a methodology for modeling 5-axis machine tools.

These experiments were expensive and time consuming. In order to save resources, most of the results in this thesis are computational. Actual cutting tests were performed for a representative sample of the simulations in order to confirm the simulated results. The subject of NC simulators is an area of active research and of great importance to industry. The NC simulator developed in the current work was based on one of the most promising techniques in the literature described in Drysdale et al. [12]. It was fully interactive with a graphical user interface.

1.2 Outline of Thesis

Chapter 2 contains a comprehensive review of the issues in 5-axis machining. Particular attention is given to the kinematics of 5-axis machines and surface machining. In Chapter 3 the Multi-Point Machining concept is introduced using the example of a spherical surface. In Chapter 4, the equations of contact between a tool and a surface are developed. These equations are used to gain insight into multi-point contact. In chapters 5 and 6 the multi-point tool positioning and adjustment algorithms are developed. Chapter 7 discusses the properties of a multi-point tool path and compares Multi-Point Machining to existing 3 and 5-axis tool positioning strategies. The final Chapter contains conclusions and recommendations.

Chapter 2

Background and Literature Review

Computer Numerically Controlled (CNC) machining is utilized extensively in producing parts with complex sculptured surfaces. It is used indirectly to produce these parts, by machining the dies and molds from which they are drawn or cast, or directly in the production of specialized parts such as turbine blades. In all cases the workpiece surface data, generated in a Computer Aided Design (CAD) package is passed to a Computer Aided Manufacturing (CAM) package or module to generate the tool path. Traditionally, these surfaces have been produced on 3-axis machines using ball nose cutters. It has been demonstrated by many researchers, including the present author, that switching from 3-axis to 5-axis technology can result in substantial savings in machining time, coupled with improved surface finish.

The first part of this chapter provides the reader with the essentials to understand 5-axis machining technology. It includes the salient features that distinguish 5-axis from 3-axis machining. The chapter also gives a uniform presentation of the kinematics of the two most common 5-axis machine configurations. The knowledge of machine kinematics is crucial to the proper implementation of the concepts presented in this chapter and those in the literature.

The second part of the chapter reviews some of the current research in 5-axis surface machining. These include: new tool path planning and tool positioning strategies to improve surface finish and reduce production time, techniques for NC simulation and algorithms for gouge detection and avoidance.

2.1 Classification of NC Milling Machines

CNC milling machines are usually described by the number of axes of motion. A 3-axis milling machine is capable of moving a cutting tool in three directions relative the workpiece. Through the use of ball nose end mills and special fixtures these machines are very flexible and can be used for both low-volume and high-volume manufacturing. A 5-axis milling machine can position a tool in three dimensional space and also control its orientation. They are especially useful for low volume, high complexity parts. Moving from 3-axis to 5-axis machine tool technology means much more than adding two rotational axes to a 3-axis machine. The additional cost of a 5-axis machine is substantial. Equally important is the extra training required for the personnel who program and operate these complex machines.

Moving from 3-axis machining technology to 5-axis technology may seem daunting at first. However, 5-axis machining provides flexibility and efficiency that cannot be obtained with 3-axis milling. A 5-axis machine can produce parts with more complex geometry using a single setup without the need for complex and expensive fixtures or the specialized cutters often used in three axis machining. A 5-axis machine can produce many parts which are impossible to produce otherwise, such as the highly twisted impellers found in aircraft turbine engines. Most importantly, 5-axis machines are substantially better at producing free-form surfaces than 3-axis machines.

2.2 The Need for 5-axis Milling Technology

A 5-axis machine gives the cutting tool access to most features of a part without changing the machine setup, as shown in Figure 2.1. This is commonly known as 5-sided machining [2]. For example, a 5-axis machine can drill holes in the sides of a part by simply rotating the part. This eliminates specialized fixtures or setup changes that would be required to produce such parts on a 3-axis machine. In the automotive industry, the extra positioning capability can be used to produce parts with odd geometry, such as cylinder port ducts in high performance engine blocks. 5-axis machines can also produce flat surfaces at arbitrary orientations by tilting a standard flat end mill. A 3-axis machine would require a special form cutter to produce the same geometry.

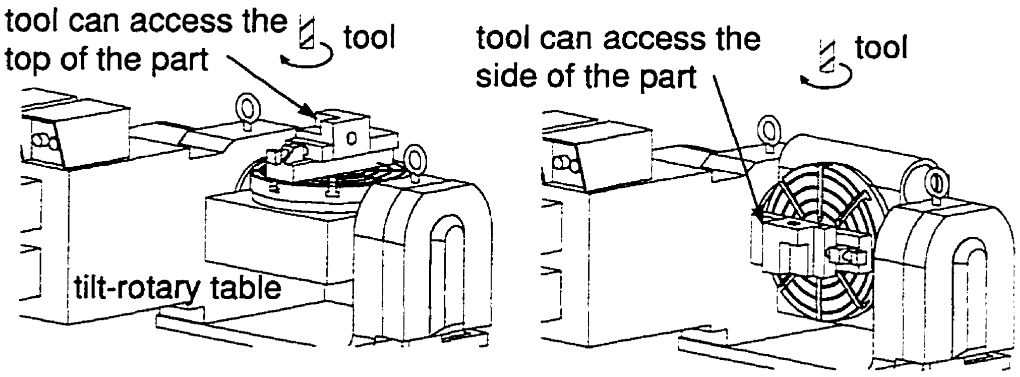

tool can access the top of the part
tool
tilt-rotary table
tool can access the side of the part
tool

Figure 2.1 Five sided machining.

5-axis machining is used in the aerospace industry to produce many of the twisted ruled surfaces that appear in jet engine parts such as impellers. These surfaces can be machined efficiently in one setup on a 5-axis machine using the side of the tool (the flank) rather than the bottom of the tool. Thus it is often called flank milling. A 3-axis machine would have to use the tool tip (point milling) to produce the same surface and may require several different workpiece setups to machine the entire blade. The improved accuracy of the product and substantial decrease in the production costs justify the additional cost associated with 5-axis machining.

5-axis machines provide a significant advantage in free-form (sculptured) surface machining. In traditional 3-axis machining a ball nose end mill machines the surface using many closely spaced passes of the cutting tool. Each pass of the tool leaves a cylindrical impression on the workpiece. These impressions can not reproduce a free-form surface exactly. While machining these impressions, the ball nose end mill performs much of its cutting near the bottom center of the tool where the tangential speed of the cutting surface is lowest, producing a poor surface finish. After machining, hand grinding and polishing are usually required to achieve a smooth surface. A 5-axis machine can tilt a flat end mill with respect to the surface leaving an elliptical shaped impression on the workpiece. By changing the angle of tilt, the dimensions of this impression can be altered to provide a better approximation of the intended surface. In addition, cutting takes place at the edge of the tilted tool where the cutting speed is the highest, producing a better surface finish than a ball nose cutter

2.3 Components of a 5-axis Milling Machine

The four integral components of a 5-axis machining system are: the milling machine, the machine controller, the CAD/CAM software and the personnel. Not all 5-axis machining systems are suitable for all tasks. The selection and proper utilization of equipment is critical to achieving the gains in efficiency available through 5-axis machining.

2.3.1 Configurations of 5-axis Milling Machines

Each 5-axis machine configuration has its own advantages. The most important issues to consider in machine selection are rigidity, work volume and accuracy. In addition, features such as automatic tool changers, and feed and spindle speeds are as important in 5-axis machines as they are in 3-axis machines.

Rigidity is desirable in all milling machines because it increases positioning accuracy and permits higher metal removal rates. Obtaining adequate rigidity can be a problem with 5-axis machines because rotational joints are usually more flexible than linear sliding joints. Generally, a more rigid machine will be more expensive. Determining the required rigidity depends on the types of material being cut and the size of cuts being performed.

The working volume of the machine is also important. It is defined by the range of the joint motions. This range of motion determines the maximum workpiece size and the ability of the tool to access certain features on the workpiece.

Accuracy is critical in a 5-axis machine. Positioning errors in the rotational axes are magnified, depending on distance from the center of rotation. As a result, the error associated with the machine varies according to the position in the work volume. This phenomenon makes it difficult to predict the expected accuracy of a workpiece. Many 5-axis machines are actually 3-axis machines that have been retrofitted with a tilt-rotary table. Misalignment of the additional rotational axes can severely impair the accuracy of the resulting workpieces.

2.3.2 The Machine Controller

A CNC machine controller controls the machine, runs G-Code programs described in the next section and provides the user interface between the machine and the operator. The controller should be capable of full 5-axis simultaneous motion. This means that during a cutting motion, all five axes move at the same time from position to position. For example, at the halfway point of a linear motion, all axes should be halfway. This ensures that the machine will cut in a smooth, predictable fashion. Some controllers are only capable of five-axis positioning. In these cases, the controller will not perform interpolation and the axes will not move in a simultaneous predicable fashion. This makes the controller simpler, but reduces the ability of the machine to perform such operations as surface machining. The controller should also be able to process data as fast as possible. For 5-axis machining of a sculptured surface many closely spaced tool positions are required. This could mean that over a stretch of a few millimeters, the controller may have to change the machine position 100 times. If the controller cannot process these positions fast enough, it will have to reduce the feed rate. This will slow down the machining process and alter the surface finish. Controllers should also be able to mechanically lock the positions of axes for heavy cuts during five sided machining operations.

2.3.3 CAD/CAM Software

CNC machines are programmed with the required tool trajectory using a special command set called G-Codes. These G-Codes are a defacto standard in the CNC machine industry. G-Codes programs can be written manually for simple parts. However, in most case Computer Aided Manufacturing (CAM) software is used to produce G-code programs directly from Computer Aided Design (CAD) models. A CAM package typically produces a G-code program in two stages. First, tool paths consisting of generic cutter locations(CLDATA) are generated. The CLDATA consists of a list of tool position in the workpiece coordinate system. The cutter locations must then be converted into G-Code programs using a post-processor specific to the NC machines that will produce the part.

The selection of a suitable CAD/CAM package for 5-axis machining is important. Many CAM packages are geared around two and a half dimensions. Such packages can perform simultaneous motion of the x and y axes but can perform only positioning motion in the z direction. These packages cannot produce free form surfaces. Many CAM packages claim to be capable of 5-axis machining, although, the available features of these systems vary greatly. Most of these packages can perform 5-axis positioning that is suitable for 5-sided machining. However, they cannot perform 5-axis surface machining. Some CAM packages can machine surfaces in 5-axes by placing the cutter tip on the surface and aligning the tool axis with the surface normal. For concave surface patches this technique will cause the tool to gouge the design surface. These packages may provide offsetting of the tool to limit gouging but at a loss in surface accuracy. The most sophisticated CAM packages can perform tilted end milling, where the tool is tilted at an angle from the surface normal and the tool tip is placed in contact with the surface. This provides more efficient metal removal and higher surface accuracy. However, the current state of 5-axis machining technology available in CAM packages is lagging behind many advanced 5-axis machining techniques that are available in the literature.

In addition to CAD/CAM packages for producing tool paths, numerical verification software plays an important role in 5-axis machining. The complex nature of 5-axis machine tool motions can be simulated to detect possible interference problems between the machine, tool, workpiece and fixtures. Software can also simulate the material removal process, that is particularly useful for detecting gouges. When gouging of the design surface is detected, the tool position must be changed to eliminate the gouge. Material removal simulation can also estimate the scallop size that will be left by the cutting process. This helps in the determination of the optimum cutter size and tool path.

2.3.4 Personnel

The final component of a 5-axis machining system is the personnel who will program and operate the machine. 5-axis machining is more complex than 3-axis machining and requires a good understanding of the machine motion. The production of parts using 5-axis machines requires good three dimensional skills for visualizing and building appropriate tool paths. Thorough training in the operation of the CAM software is important because of the multiple approaches to 5-axis machining, especially in the areas of surface machining. Special care must also be taken when setting up the machine to correctly determine tool lengths and workpiece placement relative to the rotating joints.

2.4 Kinematics of a 5-axis Milling Machine

The benefits of 5-axis machining arise from the ability of the machine to position the cutting tool in an arbitrary orientation with respect to a workpiece. In order to realize a tool position on a 5-axis machine, the kinematics of the target machine must be taken into account. Almost every 5-axis NC machine requires a different post-processor to account for different machine configuration. In fact, the post-processor requires information about the workpiece setup and tooling before it can convert generic cutter location data into specific machine dependent G-Code. Even after post-processing, the same tool path executed on different NC machines will produce noticeably different results. For example, the experimental work for this thesis was performed in four different locations in Ontario: The Industrial Research and Development Institute (IRDI) in Midland, McMaster University in Hamilton, The University of Western Ontario in London, and The University of Waterloo in Waterloo. Each experimental site required a different post-processor. This experience led to a systematic approach for modeling the kinematics of 5-axis machines.

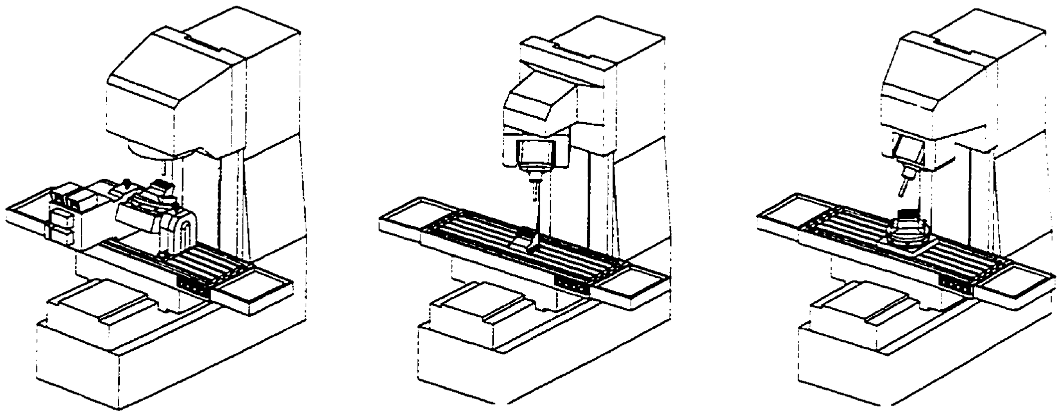
Five-axis milling machines are classified by the combination and order of their linear (T) and rotational (R) axes. For example a machine with three translations and two rotations would be specified as a TTTRR machine. There are many possible combinations of these axes that can be used to produce a 5-axis milling machine. However, as Kiridena [28] points out, there are only three commonly used machine configurations:

a) RRTTT- A tilt-rotary table mounted on three linear axes usually referred as a the tilt-rotary type 5-axis machine.

b) TTTRR - Three linear axes with the cutter oriented by two rotary axes, commonly called a wrist type or Euler type 5-axis machine.

c) RTTTR - A rotary table mounted on three linear axes and a single rotary axis for the tool.

These three types of 5-axis configurations are illustrated in Figure 2.2. The other possible configurations such as TRTTR are generally not used because of the difficulty in designing a machine with a mixture of rotational and linear axes that meets the stiffness requirements for a milling machine. Each of the configurations shown has its own advantages and disadvantages. The wrist type machines are the simplest to program, can be built to accommodate very large workpieces, but tend to be less rigid than the other configurations. They are best suited to surface machining. Tilt-rotary table type machines excel at 5-sided machining and tend to be stiffer than other configurations. However, they are more prone to setup error and may not be able to accept large workpieces.



tilt-rotary table type

wrist type

rotary table type

Figure 2.2 Typical 5-axis machine configurations.

When programming an NC machine, the motion of each joint must be specified in order to achieve the desired position and orientation of the tool relative to the workpiece. This is referred to as the inverse kinematics problem. The programmer will also need to know the resulting position and orientation of the tool for a given set of joint commands in order to verify that the tool path is correct. This is referred to as the forward or direct kinematics problem.

The Denavit-Hartenberg (D-H) convention is a method for modeling machine kinematics made popular in the robotics field. See for example Paul [44] for a detailed explanation of the D-H convention. When using the D-H convention, local coordinate systems are attached to each of the machines' joints. Standardized homogeneous transformations are used to relate the next coordinate system to the previous coordinate system. Each transformation consists of: a rotation about the x-axis, followed by a translation along the x-axis, a translation along z-axis, and a rotation about the z-axis of the previous coordinate system. By blindly following the rules outlined in the D-H convention it is possible to model any machine containing any number of prismatic (translating) and revolute (rotating) joints. This approach is very useful when modeling complex systems whose motion is difficult to visualize. However, the D-H convention is highly abstract. The use of standardized transformations means that there may be a poor correspondence between the physical joint motions and the mathematical transformations. For example, a translation along the y-axis in machine space will require a rotation about the x-axis of 90° followed by a translation along the z-axis. It would have been much easier just to translate along the y-axis. For this reason I have decided to use a single transformation at each joint instead of the D-H convention when modeling a 5-axis milling machine.

2.4.1 Forward Kinematics of a Tilt-Rotary Table Type 5-axis Milling machine

When modeling a tilt rotary table type 5-axis machine it is convenient to consider the coordinate systems illustrated in Figure 2.3. In this figure, the machine coordinate system, $\mathbf{C}_m$, is fixed to the most positive location in the work volume of the NC machine tool. All commands sent to the machine are in terms of the machine coordinate system. All other coordinate systems are for human convenience. The programmed coordinate system, $\mathbf{C}_p$, is located by the vector, $\mathbf{m}$, relative to the machine coordinate system during the workpiece setup prior to machining. It is essential that this vector be set such that the center of rotation of the tilt-rotary table is coincident with the tool tip. After this vector is set, a command sent to the controller to move to position (0, 0, 0) will place the tool tip at the center of rotation. The tilt-rotary table coordinate systems, $\mathbf{C}_{tr}$, and the rotating coordinate systems, $\mathbf{C}_A$ and $\mathbf{C}_C$, are attached to the center of rotation of the tilt-rotary table. Note that there are many different ways to configure a tilt-rotary table depending on the initial position of the rotary axes. The most basic configurations assume that the table is initially horizontal or vertical. This analysis assumes that the table is initially horizontal. The workpiece coordinate system, $\mathbf{C}_{wp}$, moves with the tilt-rotary table. The workpiece offset vector, **wp,** gives the position of the workpiece coordinate system relative to the tilt-rotary table coordinate systems.

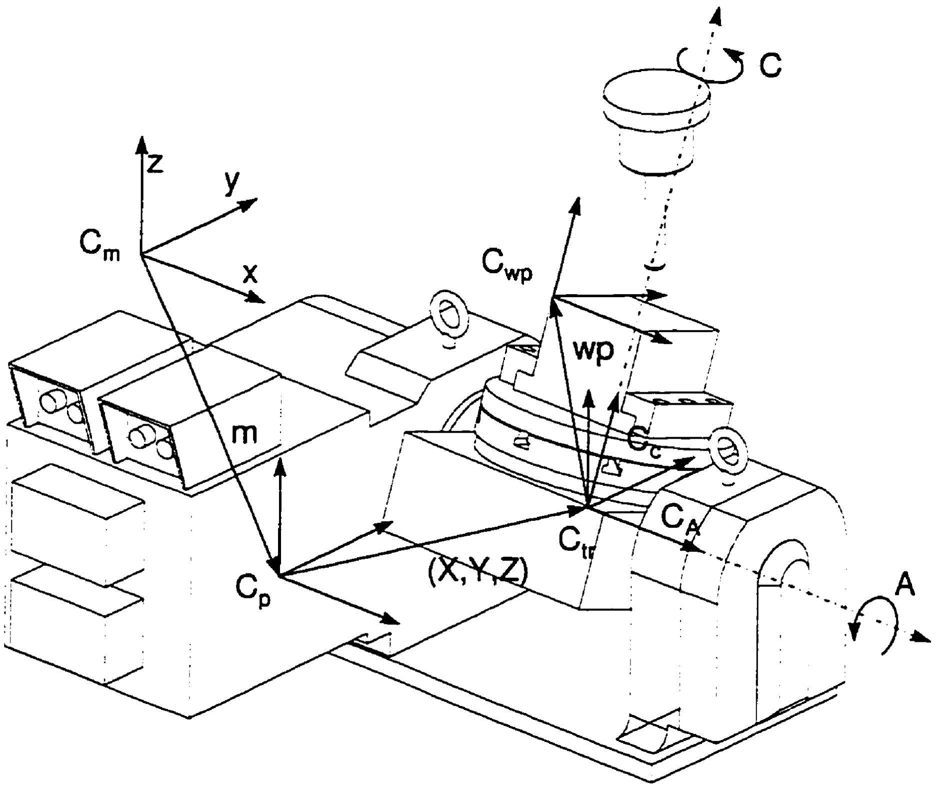


Figure 2.3. Kinematics of a tilt rotary table type 5-axis machine.

When a cutter location file is post-processed, the post-processor uses the workpiece offset vector, **wp**, to convert the cutter location data into G-codes. Each G-code position command consists of X, Y, Z, A, and C components. The tilt-rotary table coordinate system is translated by the X, Y and Z commands relative to the programmed coordinate system and the workpiece coordinate system will be rotated by the A and C commands about the x and z axes in tilt-rotary table coordinate systems. The NC controller converts commands given in the programmed coordinate system to the machine coordinate system using the machine offset vector, **m**.

In order to model the kinematics of the NC machine, homogeneous transformations are used to establish the relationship between the defined coordinate systems. For this exercise a point, $\mathbf{p}^{wp}$, will ultimately be transformed from the workpiece coordinate system into the machine coordinate. The superscript on the point will refer to the coordinate system in which the point is defined and the subscripts indicate a particular component of the vector. The position of a point, $\mathbf{p}^{wp}$, in the workpiece coordinate system, $\mathbf{C}_{wp}$, expressed in the tilt-rotary table coordinate system, $\mathbf{C}_{tr}$, is given by:

$$\begin{bmatrix} p_x^{tr} \\ p_y^{tr} \\ p_z^{tr} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(A) & -\sin(A) & 0 \\ 0 & \sin(A) & \cos(A) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(C) & -\sin(C) & 0 & 0 \\ \sin(C) & \cos(C) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & wp_x \\ 0 & 1 & 0 & wp_y \\ 0 & 0 & 1 & wp_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x^{wp} \\ p_y^{wp} \\ p_z^{wp} \\ 1 \end{bmatrix} \qquad 2.1$$

Next, the point, $\mathbf{p}^{tr}$, now in the tilt-rotary table coordinate system, $\mathbf{C}_{tr}$, is transformed into the programmed coordinate system, $\mathbf{C}_p$, as follows:

$$\begin{bmatrix} p_x^{p} \\ p_y^{p} \\ p_z^{p} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X \\ 0 & 1 & 0 & Y \\ 0 & 0 & 1 & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x^{tr} \\ p_y^{tr} \\ p_z^{tr} \\ 1 \end{bmatrix} \qquad 2.2$$

Finally, the point, $\mathbf{p}^{p}$ in the programmed coordinate system, $\mathbf{C}_p$, is translated into the machine coordinate system, $\mathbf{C}_m$, using the machine offset vector, **m**:

$$\begin{bmatrix} p_x^{m} \\ p_y^{m} \\ p_z^{m} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & m_x \\ 0 & 1 & 0 & m_y \\ 0 & 0 & 1 & m_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x^{p} \\ p_y^{p} \\ p_z^{p} \\ 1 \end{bmatrix}, \qquad 2.3$$

where $\mathbf{p}^{m}$ is the point in the machine coordinate system. By combining all the transformation matrices together, a point in the workpiece coordinate system, $\mathbf{p}^{wp}$, can be expressed in the machine coordinate system by:

$$\begin{bmatrix} p_x^m \\ p_y^m \\ p_z^m \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(C) & -\sin(C) & 0 & X+m_x \\ \cos(A)\sin(C) & \cos(A)\cos(C) & -\sin(A) & Y+m_y \\ \sin(A)\sin(C) & \sin(A)\cos(C) & \cos(A) & Z+m_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x^{wp} \\ p_y^{wp} \\ p_z^{wp} \\ 1 \end{bmatrix} \qquad 2.4$$

This transformation determines the relationship between a point in the workpiece coordinate system and the machine coordinate system. This relationship is required to simulate the metal removal on a specific NC machine. The post-processor needs the inverse of this relationship to determine the joint positions required to place the tool at the correct location in the workpiece coordinate system.

2.4.2 Inverse Kinematics of a 5-axis Tilt-Rotary Table Type 5-axis Milling Machine

The tool path used in 5-axis machining will consist of a set of tool positions, **tpos**, and a corresponding set of tool orientation vectors, **taxis,** in the workpiece coordinate system. The post-processor must convert this information into angular (A, C) and linear (X, Y, Z) to place the tool in the correct orientation and position relative to the workpiece in the programmed coordinate system. Since the tool orientation in a tilt-rotary type machine is fixed on the z-axis in the programmed coordinate system, the correct orientation is achieved by rotating the workpiece about the A and C axes until the tool orientation vector lines up with the z-axis. In other words, the tool orientation vector, **taxis**, is [0, 0, 1] in the tilt-rotary coordinate system. In this way, the rotations A and C, can be found by solving for A and C in the transformation matrix.

$$\begin{bmatrix}0\\0\\1\\1\end{bmatrix}=\begin{bmatrix}\cos(C) & -\sin(C) & 0 & 0\\ \cos(A)\sin(C) & \cos(A)\cos(C) & -\sin(A) & 0\\ \sin(A)\sin(C) & \sin(A)\cos(C) & \cos(A) & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}taxis_x\\ taxis_y\\ taxis_z\\ 1\end{bmatrix} \tag{2.5}$$

However, since these are transcendental equations it is difficult to determine the correct sign of the results. For instance, the first row in the matrix can be used to conveniently solve for the C rotation.

$$C=\pm\tan^{-1}\left(\frac{taxis_x}{taxis_y}\right) \tag{2.6}$$

Care is needed to determine the correct sign of the result. Furthermore, a 5-axis machine can position and orient the tool correctly in two different ways using a negative or positive A rotation. For example, a tool orientation vector, **taxis**, of [0.577,0.577,0.577] can be achieved by A and C rotations of (45°, 54.731°) or (-135°, -54.731°). The selection of which set of angles to use depends on the machine and workpiece setup. For these reasons it is better to calculate the magnitude of the rotations first and then determine the correct signs of the rotations based on the quadrant of the tool orientation vector. The following algorithm determines the correct A and C values assuming the tool orientation vector always points upwards and the A rotation is always negative. Using this approach the angle between the tool orientation vector and the positive z-axis is:

$$a=\cos^{-1}\left(\frac{\sqrt{taxis_x^2+taxis_y^2}}{\sqrt{taxis_x^2+taxis_y^2+taxis_z^2}}\right) \tag{2.7}$$

and the angle between the tool orientation vector and the y-axis is:

$$c=\cos^{-1}\left(\frac{|taxis_y|}{\sqrt{taxis_x^2+taxis_y^2}}\right) \tag{2.8}$$

The correct angles can now be determined based on the quadrant of the tool orientation vector in the workpiece coordinate system as shown in Figure 2.4.

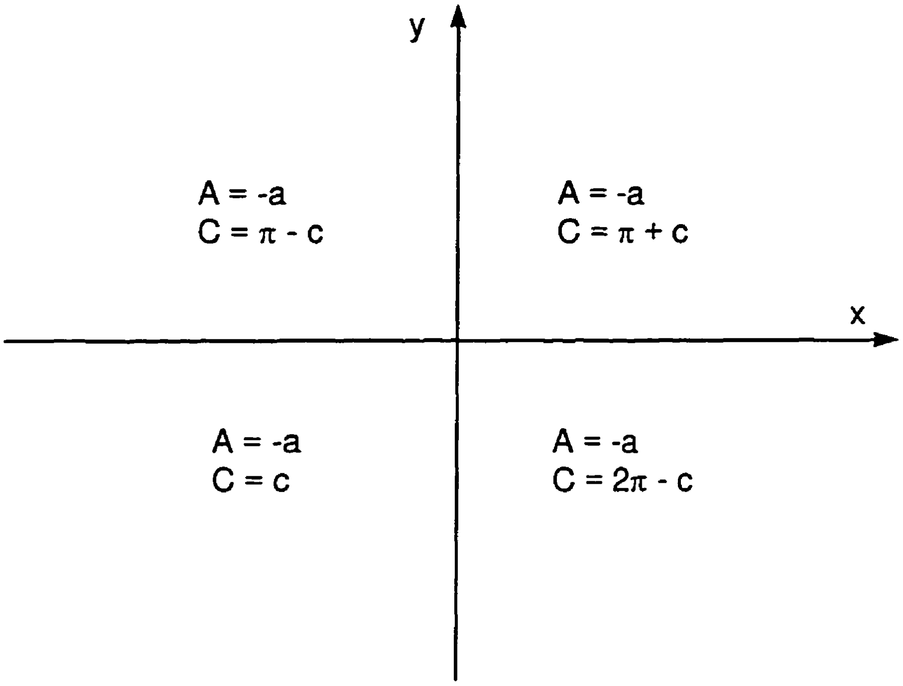


Figure 2.4 Selection of quadrant for rotations.

After the rotations have been determined, the translation can be found by transforming the tool position into the workpiece coordinate system and rotating the resulting vector by A and C. The resulting displacements are the joint commands X, Y, Z needed to place the tool at the correct location on the rotated workpiece and can be determined as follows:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(A) & -\sin(A) & 0 \\ 0 & \sin(A) & \cos(A) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(C) & -\sin(C) & 0 & 0 \\ \sin(C) & \cos(C) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & wp_x \\ 0 & 1 & 0 & wp_y \\ 0 & 0 & 1 & wp_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} tpos_x \\ tpos_y \\ tpos_z \\ 1 \end{bmatrix} \qquad 2.9$$

2.4.3 Forward Kinematics of a Wrist Type 5-axis Milling machine

The kinematics of a wrist type 5-axis machine are easier to model because the workpiece coordinate system is never rotated. Figure 2.5 shows the coordinate systems used when modeling a wrist type 5-axis milling machine. As always, the machine coordinate system $\mathbf{C_m}$ is fixed to the most positive location in the work volume. The programmed coordinate system $\mathbf{C_p}$ is located by the vector **m** such that the center of rotation of the wrist is initially coincident with workpiece coordinate system. The wrist coordinate systems $\mathbf{C_w}$ and the rotating coordinate systems $\mathbf{C_A}$ and $\mathbf{C_C}$, are attached to the wrist's center of rotation. The tool offset vector **t** gives the position of the tool tip relative to the wrist coordinate system. The length of this vector depends on the tool length and the placement of the tool in the tool holder.

Cutter location data will be converted to G-code by the post-processor once the magnitude of tool offset vector is known. Since the magnitude of the tool offset vector changes, a tool path must be post-processed whenever the tool is changed. A G-code position command sent to the 5-axis machine controller will consist of X, Y, Z, A and C components used to command the joints. The wrist is translated by the X, Y, Z position command relative to the programmed coordinate system and the tool is rotated about the wrist by the A and C commands. These commands are converted to the machine coordinate system by the NC controller.

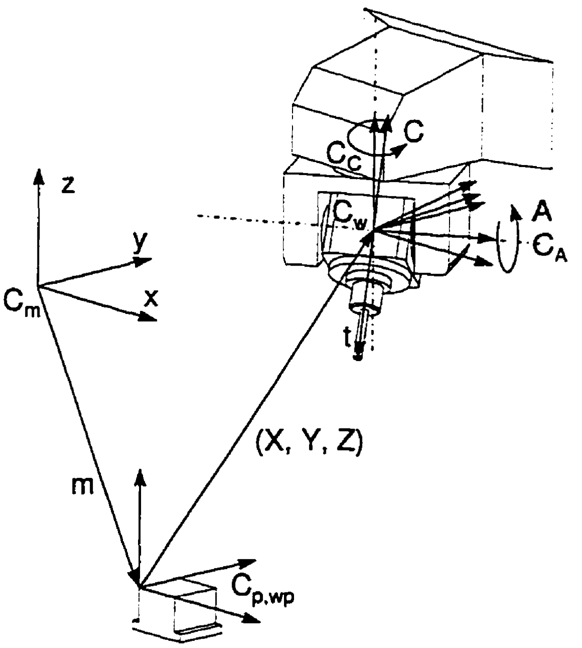


Figure 2.5 Kinematics of a 5-axis wrist type milling machine.

Using vector algebra, the position of the tool tip, **t**, in the wrist coordinate system, $\mathbf{C}_w$, is given by:

$$\begin{bmatrix} t_x^w \\ t_y^w \\ t_z^w \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(C) & -\sin(C) & 0 & 0 \\ \sin(C) & \cos(C) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(A) & -\sin(A) & 0 \\ 0 & \sin(A) & \cos(A) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ t_z \\ 1 \end{bmatrix} \qquad 2.10$$

Similarly, the tool tip, $\mathbf{t}^w$, in the wrist coordinate system, $\mathbf{C}_w$, can be transformed into the programmed coordinate system, $\mathbf{C}_p$, as follows:

$$\begin{bmatrix} t_x^p \\ t_y^p \\ t_z^p \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X \\ 0 & 1 & 0 & Y \\ 0 & 0 & 1 & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t_x^w \\ t_y^w \\ t_z^w \\ 1 \end{bmatrix} \qquad 2.11$$

Finally, the tool tip, $\mathbf{t}^p$, in the programmed coordinate system, $\mathbf{C}_p$, can be transformed into the machine coordinate system, $\mathbf{C}_m$, by:

$$\begin{bmatrix} t_x^m \\ t_y^m \\ t_z^m \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & m_x \\ 0 & 1 & 0 & m_y \\ 0 & 0 & 1 & m_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t_x^p \\ t_y^p \\ t_z^p \\ 1 \end{bmatrix} \qquad 2.12$$

By combining all the transformation matrices together, a tool tip can be expressed in the machine coordinate system by:

$$\begin{bmatrix} t_x^m \\ t_y^m \\ t_z^m \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(C) & -\sin(C)\cos(A) & \sin(C)\sin(A) & X+m_x \\ \sin(C) & \cos(A)\cos(C) & -\cos(C)\sin(A) & Y+m_y \\ 0 & \sin(A) & \cos(A) & Z+m_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ t_z \\ 1 \end{bmatrix} \qquad 2.13$$

This transformation determines the location of the tool tip in the machine coordinate system.

2.4.4 Inverse Kinematics of a Wrist Type 5-axis Milling Machine

In the same manner as for the tilt-rotary table, the generic tool paths must be post-processed into suitable joint commands. Again, the vector **tpos** stores the tool tip position and the vector **taxis** stores the tool orientation. For the wrist type machine, the correct orientation is achieved by rotating the tool about the A and C axes until the tool lines up with the tool orientation vector. These values can be determined by solving for A and C in the set of equations given below.

$$\begin{bmatrix} \text{taxis}_i \\ \text{taxis}_j \\ \text{taxis}_k \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(C) & -\sin(C)\cos(A) & \sin(C)\sin(A) & 0 \\ \sin(C) & \cos(A)\cos(C) & -\cos(C)\sin(A) & 0 \\ 0 & \sin(A) & \cos(A) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{t}_x \\ \hat{t}_y \\ \hat{t}_z \\ 1 \end{bmatrix} \qquad 2.14$$

Where $\hat{t}$ is the normalized tool tip vector in its initial position. However, difficulties arise in this method because the equations are transcendental and two possible sets of rotations can be used to position the tool. Instead, the same approach outlined in the section on tilt-rotary table will be used. The angle between the tool negative z-axis in the workpiece coordinate system is:

$$a = \cos^{-1}\left(\frac{\sqrt{\text{taxis}_x^2 + \text{taxis}_y^2}}{\sqrt{\text{taxis}_x^2 + \text{taxis}_y^2 + \text{taxis}_z^2}} \right) \qquad 2.15$$

and the angle between the tool orientation vector and the y-axis is:

$$c = \cos^{-1}\left(\frac{|\text{taxis}_y|}{\sqrt{\text{taxis}_x^2 + \text{taxis}_y^2}} \right) \qquad 2.16$$

The correct angles can now be determined based on the quadrant of the tool orientation vector in the workpiece coordinate system as shown in Figure 2.6.

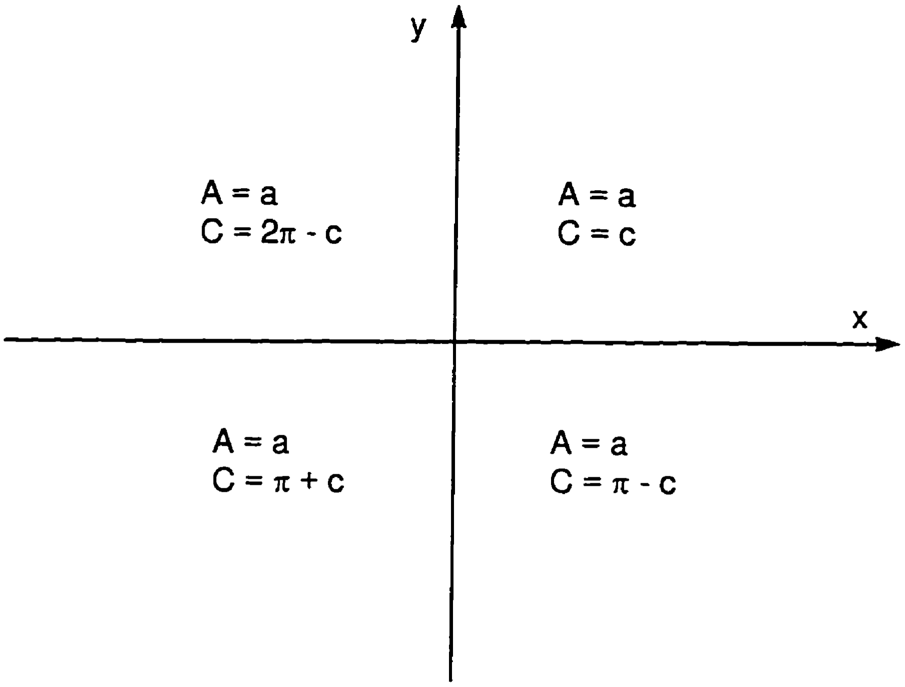


Figure 2.6 Selection of quadrant for rotations.

After the rotations have been determined, the translation can be found by translating the wrist coordinate system along the tool axis by the magnitude of tool offset length from the tool position in the workpiece coordinate system.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} tpos_x \\ tpos_y \\ tpos_z \end{bmatrix} + |t| \begin{bmatrix} taxis_x \\ taxis_y \\ taxis_z \end{bmatrix}$$

2.17

2.5 Surface Machining

Sculptured surfaces are generally produced in three stages: roughing, finishing, and benchwork. Roughing cuts are used to remove most of the material from a workpiece while leaving the part slightly oversized. Finish machining of a sculptured surface removes material from the roughed out workpiece and attempts to machine the part to its final dimensions. The resulting surface is left with a large number of scallops, as shown in Figure

1.1 and reproduced in Figure 2.7. Benchwork consisting of grinding and polishing is used to remove these scallops. The time spent on finishing and benchwork is dependent on the size of these scallops. The scallop size is related to the tool pass interval also known as the cross feed; reducing the tool pass interval will decrease the scallop size at the expense of increased machining time. A recent survey by LeBlond Makino of Mason, Ohio [2], stated that a small mold will typically require 57 hours of roughing, 127 hours of finishing and 86 hours of grinding and polishing. Over 78% of the total production time is spent on finishing, grinding and polishing. Clearly, there is a need for faster machining techniques that produce smaller scallops, and hence require little or ultimately no benchwork.

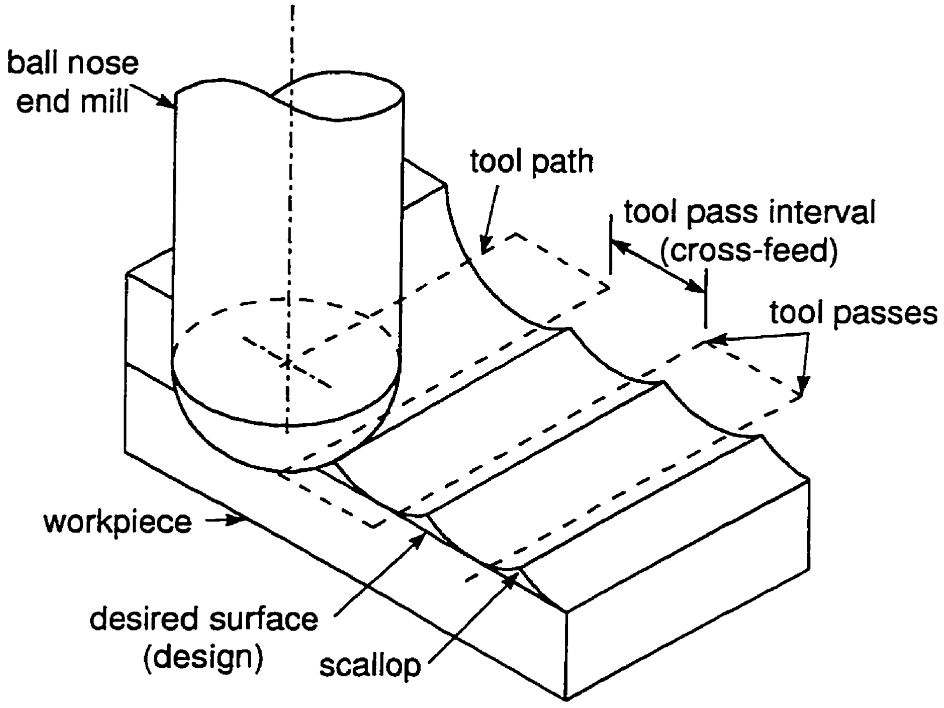


Figure 2.7 Scallops left after machining.

The tool path used to machine a surface is generally produced in three stages. First, tool path planning is used to determine the path the tool will take as it machines a surface. Tool path planning research is primarily concerned with the spacing between points on the tool path and determining the tool pass interval. Second, tool positioning strategies are used to determine the cutter location and orientation at specific points on the tool path. The objective of a tool positioning strategy is to minimize the material remaining between the tool and the design surface as the tool moves along the tool path. Finally gouge detection and correction algorithms are used to determine if the tool has penetrated the desired surface and eliminate this penetration.

2.5.1 Tool Path Planning

An NC tool path used to machine a sculptured surface consists of a set of tool positions. The NC controller interpolates sequentially between these points as the tool moves from point to point. The tool path is usually designed so that the tool makes several tool passes across the surface. The tool pass interval, or cross feed, effects the size of the scallops. Furthermore, the interpolation between individual tool positions on a tool pass may cause the tool to leave the design surface. An ideal tool path will result in a surface with uniform and small sized scallops evenly distributed across the surface. The size of the scallops will have been determined before machining. In the next section the tool pass interval will be considered, followed by a discussion of the spacing between individual tool positions along a tool pass.

2.5.2 Tool Pass Interval Determination

The tool pass interval along with the tool type and surface characteristics determine the size of the scallop left on the surface for a given tool positioning strategy. For the most part, research in this area has focused on tool pass determination for ball nosed end mills. Very little work has been done in tool pass interval calculations for the more complex tool geometry prevalent in 5-axis surface machining. Fortunately, most of the ideas formulated for ball nosed tools can be extended to other types of tools.

The tool path generation algorithms for parts designed with sculptured surfaces can be broadly characterized as either constant or non-constant parameter techniques. Much of the initial work in tool path planning concentrated on constant parameter tool paths; see for example [51]. A constant parameter tool path is usually generated from a parametric surface description of the form shown below.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} f(u,v) \\ g(u,v) \\ h(u,v) \end{bmatrix}, \qquad 2.18$$

where u and v are the parameters of the surface definition. By maintaining one of the parameters constant while varying the other parameter, a tool path consisting of a number of constant parameter curves on the surface can be defined. This approach is computationally efficient because the tool path is easily determined from the surface definition. However, the relationship between the parametric coordinate and the corresponding physical (Cartesian, (x, y, z)) coordinates is not uniform [5]. Therefore, the accuracy and efficiency of a constant parameter tool path will vary with the surface description. Figure 2.8. shows a typical fan shaped example of this type of tool path. At the top of the part the tool paths are far apart and will produce a poor surface finish. At the bottom of the part an unnecessarily large number of tool motions were used to machine the part to a specified surface tolerance.

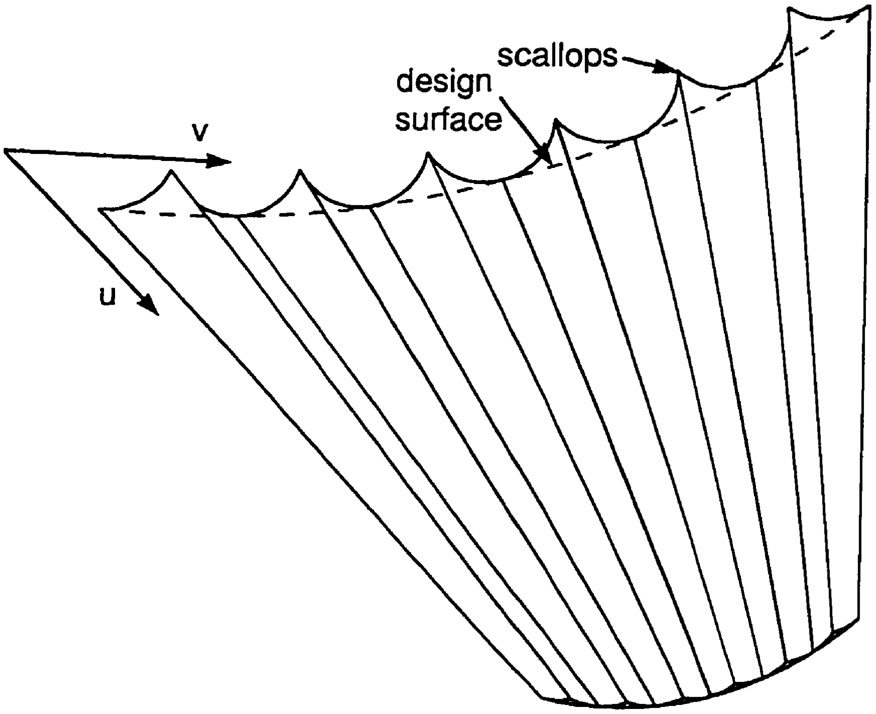


Figure 2.8 Constant parameter tool path.

Most current research in tool path planning is focused on developing non-constant parameter tool paths. These methods determine the tool pass intervals by an estimate of the required scallop height. Choi et al. [7] and Huang and Oliver [19] performed tool path planning in the x-y plane. Cutting curves are defined by intersections of a group of parallel cutter planes. An example of this type of tool path is shown in Figure 2.9. This tool path does not suffer from the problem of divergence.

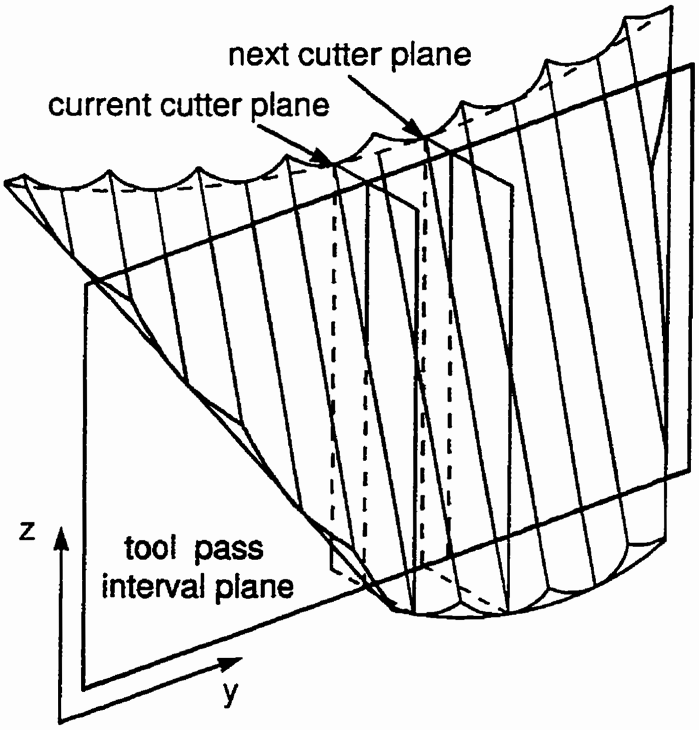


Figure 2.9 Non-constant parameter tool path.

The tool pass interval was calculated by considering a plane called the tool pass interval plane, which is perpendicular to the cutter planes containing two adjacent tool positions as shown in the Figure 2.10. The methodology Huang and Oliver [19] use for calculating the pass interval used for a concave surface is briefly explained here. The intersection curve of the surface with the tool pass interval plane can be approximated as a circular arc [8,52] between the two cutter contact points $\mathbf{cc}_1$, and $\mathbf{cc}_2$ whose radius, ρ, is the radius of curvature of the surface at $\mathbf{cc}_1$ in the tool pass interval plane. From the geometry of the figure it can be deduced that the tool pass interval, l, is:

$$l = \frac{\rho\sqrt{4(\rho+r)^2(\rho+h)^2-\left(\rho^2+2rh+(\rho+h)^2\right)^2}}{(\rho+r)(\rho+h)} \qquad 2.19$$

Where h is the scallop height. The calculation of the tool pass interval is carried out for every cutter contact point on a tool pass. The smallest tool pass interval is used to calculate the next cutter plane. All scallop heights will be within tolerance provided the scallop height approximation is close and the smallest tool pass interval is selected. However, there is no way to guarantee that the scallops are of uniform height across the entire surface. Depending, on the surface curvature some sections of the surface may have scallops that are smaller than necessary.

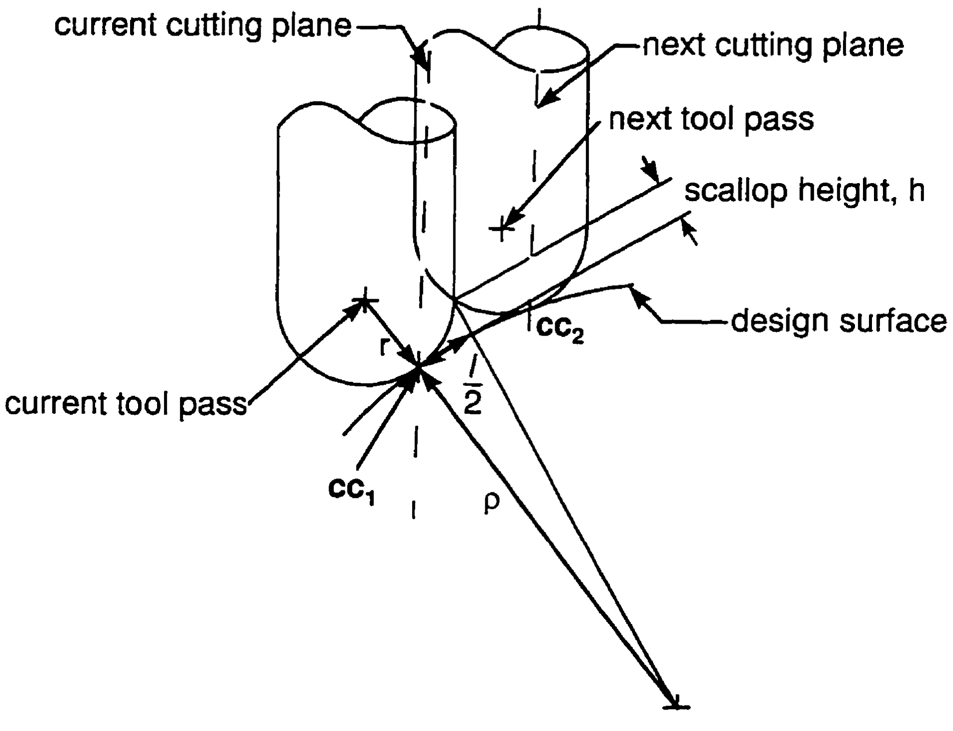


Figure 2.10 Tool pass interval calculation for a concave surface.

Recent work in tool path planning has concentrated on non-constant tool pass interval techniques that maintain a constant scallop height between pass intervals. These tool path planning strategies vary the tool pass interval along the entire length of a tool pass in order to maintain a constant scallop. The approaches advocated by Suresh and Yang [53] and by Lin and Koren [35] start with an initial master tool pass. In most cases the boundary of a surface patch or an iso-parametric curve on the design surface is selected as the master tool pass.

Succeeding tool passes are generated by calculating a pass interval that will result in a specified scallop height for every point on the master tool path. This new set of points forms the next tool pass, which in turn becomes the new master tool pass. The main difference between the two techniques lies in the methods used to calculate the pass interval. Using differential geometry Suresh and Yang [53] have developed a set of equations based on the tool radius, scallop height and surface curvature in the direction perpendicular to the tool pass cutting plane to calculate the pass interval for a specified scallop height. Their method results in a set of complex equations that must be solved numerically. Lin and Koren [35] base their tool pass calculation on the work of Vickers and Quan [57]. The resulting expression approximates the tool pass interval based on the desired scallop height, the tool radius and the curvature at the cutter contact point.

Most research on tool pass interval calculation has focused on surface machining using a ball nosed cutter. However, there is still a need for constant scallop height tool paths for other types of tools for use in 5-axis machining. The main stumbling block is a lack of methods for calculating scallop heights for these more complicated situations. Bedi et al. [4] address the issue of scallop height estimation with a toroidal and flat end mill. The authors note that a toroidal cutter can be approximated by a ball nose end mill of the appropriate radius at the cutter contact point. They then use the scallop height expressions developed by Vickers and Quan [57] for a ball nosed tool to approximate the scallop heights produced by flat and toroidal end mills.

Lee and Chang [31] address the problem of scallop height estimation during 5-axis surface machining with a flat end mill by considering tool positions on adjacent tool passes. The cutting edge of the tool at positions $\mathbf{cc}_1$ and $\mathbf{cc}_2$ is projected onto a plane, $\mathbf{P}_0$, normal to the feed direction containing both cutter contact points, as illustrated in Figure 2.11. Since the cutting edge of an end mill is a circle, the projected shape will be an ellipse. Lee and Chang [31] refer to this ellipse as the effective cutting shape. The intersection point, **M**, between the two effective cutting shapes is first found. The intersection curve, **C**, between the design surface and the plane, P_0, is approximated as a circular arc using the curvature of the surface in the plane. The scallop height, h, is found by taking the distance between the point M, and the intersection curve. Lee and Chang[31] used this methodology to numerically estimate the scallop height between tool passes to ensure that the scallops were within a specified tolerance.

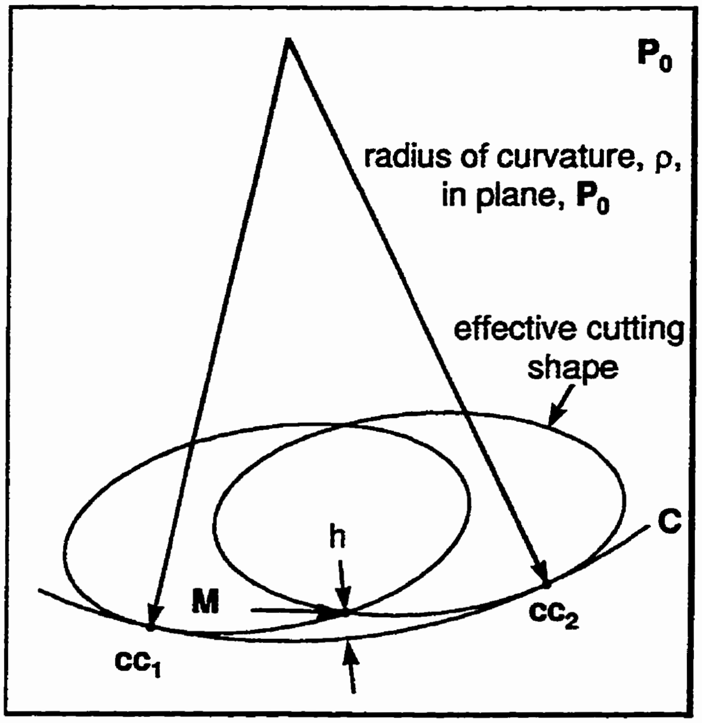


Figure 2.11 Scallop height estimation when surface machining in 5-axis with a flat end mill.

To date, most research on scallop height has considered tool and surface geometry. The missing ingredient is tool motion. Presently all tool pass interval calculations assume the tool travels linearly between static tool positions. This assumption makes it reasonable to project the cutting profile of a tool onto a plane. This approach is reasonable when using a ball nosed tool whose profile is always circular or in 3-axis machining where motions are generally piece-wise linear. However tool paths used to machine surfaces in 5-axis tend to be nonlinear. Effective cutting profiles need to be swept along the tool path rather than projected onto a plane to reflect the swept volume produced by the tool motion.

2.5.3 Tool Position Spacing

The second important issue in tool path planning is to determine the spacing between successive tool positions along a tool pass. The NC machine controller receives the tool path as a sequence of joint commands, for example (X, Y, Z, A, C). The NC controller performs linear interpolation in joint space between two (X, Y, Z, A, C) points to form the actual tool path. The tool path will not be a set of line segments because a 5-axis milling machine contains rotational axes. The tool path planner must ensure that the tool remains within tolerance with the design surface as it moves between cutter location points by selecting a small enough spacing between tool positions. If, however, the tool positions are too closely spaced, the time needed to generate the tool path and the storage requirements for the data may be excessive. More importantly, if the rate of data transfer between the NC controller and NC machine is too slow, the milling machine may be forced to wait for the next positioning command. This data starvation problem can result in reduced machining rates and jerky motion.

A considerable body of research is devoted to determining the best spacing between tool positions in 3-axis surface machining. The objective of that research is to determine the maximum allowable separation between tool positions that will maintain a required tolerance between the tool and the surface as the tool moves from one position to the next. All available work in that area assumes the tool moves linearly between successive tool positions. Many tool path planning techniques use the nominal chordal deviation as a measure of the machining error [17, 19] as shown in Figure 2.12.

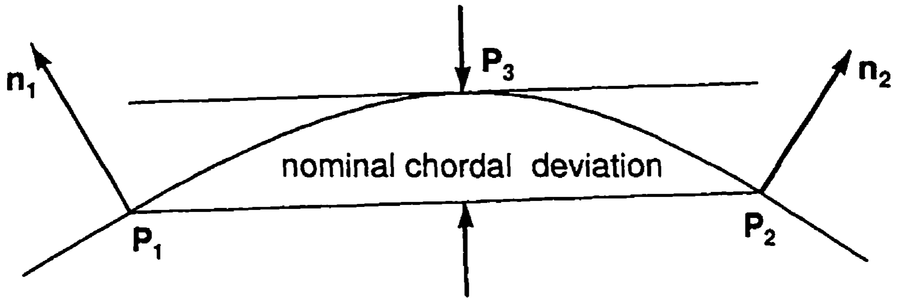


Figure 2.12 Nominal chordal deviation.

A straight forward method of determining the nominal chordal deviation is to calculate $\mathbf{P}_3$ by taking the halfway point between $\mathbf{P}_1$ and $\mathbf{P}_2$ in parametric space and determining the perpendicular distance between $\mathbf{P}_3$ and the line defined by the cord. This method will produce a reasonable approximation of the chordal deviation provided the surface has fairly uniform parametric variation. However, if the parametric variation is non-uniform, the error in this approximation can be significant. Loney and Ozsoy [36] and Wysocki et al. [65] developed numerical techniques for calculating the nominal chordal deviation based on a curve subdivision technique and a cast-and-correct method respectively.

As Huang and Oliver [19] point out, using the nominal chordal deviation as an estimate of machining error can lead to an underestimation of the true machining error if the normal vectors ($\mathbf{n}_1$ and $\mathbf{n}_2$) are not parallel and if both normals are not perpendicular to the chord. The true machining error can only be determined by considering the trajectory of the entire tool as it moves along the path as shown in Figure 2.13.

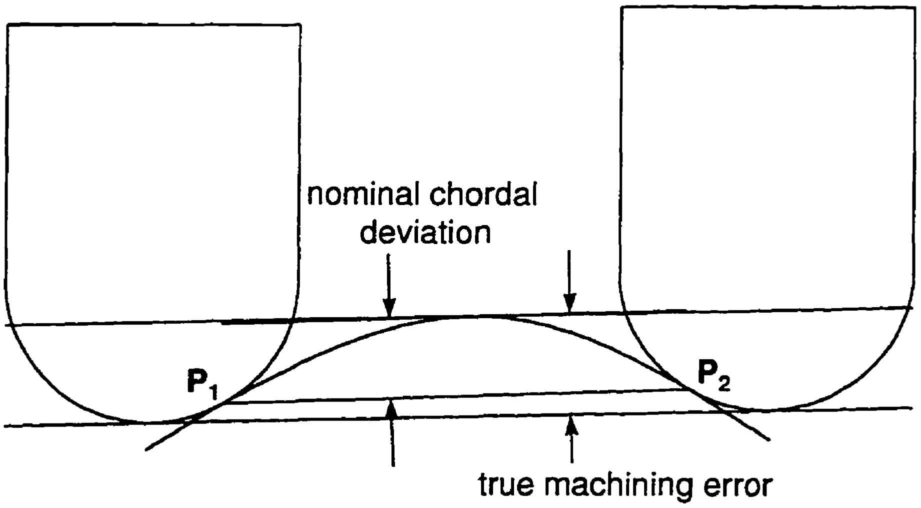


Figure 2.13 True machining error versus nominal chordal deviation.

As the tool moves from $\mathbf{P}_1$ to $\mathbf{P}_2$ the true machining error occurs between the surface and the common tangent joining both tool positions. The technique developed by Huang and Oliver [19], uses the nominal chordal deviation as an initial estimate of the true machining error.

Iterations of the orthogonal projection method [45] are then used to converge to the true machining error.

The trajectory error is the result of the inability of present day NC controllers to do more than linear interpolation between data points. Another approach to solving this problem is to modify the NC controller to perform more sophisticated methods of interpolating between tool positions. Chou and Yang [10] have developed the equations to trace any arbitrary curve in Cartesian space for a wrist type 5-axis milling machine given that the NC programmer can control the velocity and acceleration of each joint continuously in time. They fit the tool path with a polynomial, resulting in a parametric description of the tool path X(u), Y(u), Z(u), A(u), C(u). Expressions for velocity and acceleration in terms of u are then developed to produce the desired tool path for a given feedrate. Chou and Yang have also developed expressions for the jerk that may be of use for predicting vibrations, excessive wear and tracking errors. They used these equations to simulate the position, velocity, acceleration and jerk of a tool path described by a third order parametric space curve.

Other than the work by Chou and Yang [10], this author is not aware of any research into trajectory error that takes into account the non-linear trajectories produced by a 5-axis milling machine as it interpolates between tool positions. At present, 5-axis part programmers must rely on their judgment to determine appropriate spacing between tool positions.

2.5.4 Tool positioning Strategies

Tool positioning strategies are used to determine how a tool will be placed relative to the design surface. The main objective of these strategies is to remove as much material from the workpiece as possible without cutting into the desired surface (gouging) as the tool moves between tool positions.

To a certain extent, the tool positioning strategy is determined by the type of tool. The most commonly used tools in the machining of molds and dies are shown in Figure 2.14; they are: the ball nose cutter with radius, r, the flat bottom end mill of radius R, and the toroidal cutter, which is characterized by two radii, the radius of the insert r, and the radial distance between the center of the tool and the center of the insert, R. As the tools rotate the cutting surfaces of the above cutters are a sphere, a cylinder and a torus corresponding to the ball nose, the flat and the toroidal cutters respectively.

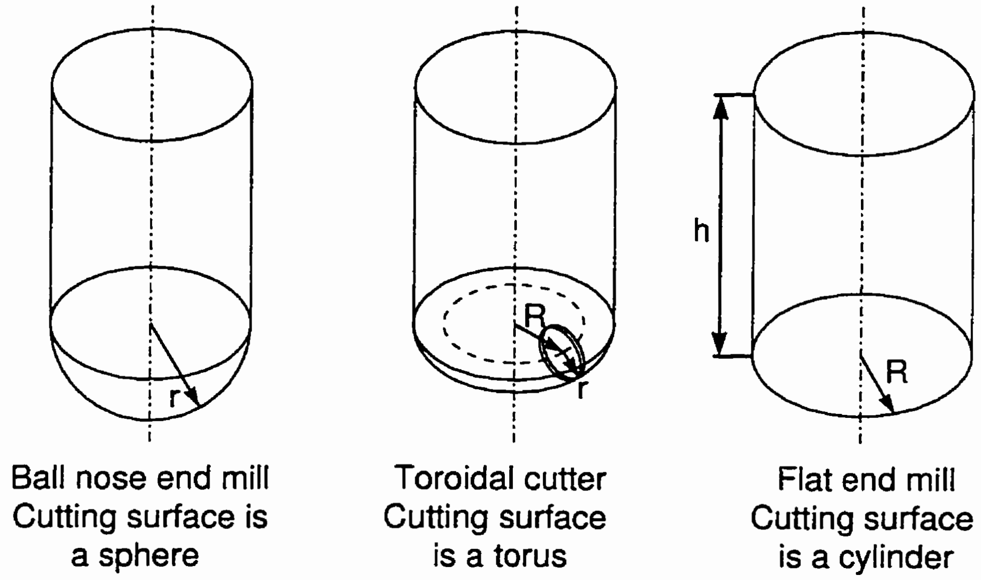


Figure 2.14 Cutting tools typically used for 5-axis surface machining.

The cutting surface of each of these tools can be modeled as a set of parametric equations or as an algebraic expression. The algebraic definitions for a sphere centered on the origin is:

$$x^2 + y^2 + z^2 - r^2 = 0 \qquad 2.20$$

and the parametric definition is:

$$\mathbf{T}(\theta,\phi) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r\sin(\theta)\cos(\phi) \\ r\sin(\theta)\sin(\phi) \\ r\cos(\theta) \end{bmatrix}$$

$$0 \le \theta \le \pi \qquad 2.21$$

$$0 \le \phi \le 2\pi$$

The algebraic definitions for a torus centered on the origin is:

$$(x^2+y^2+z^2+R^2-r^2)^2-4R^2(x^2+y^2) \;=0 \qquad 2.22$$

and the parametric definition is:

$$T(\phi,\theta)=\begin{bmatrix}x\\y\\z\end{bmatrix}=\begin{bmatrix}(R+r\cos(\phi))\cos(\theta)\\(R+r\cos(\phi))\sin(\theta)\\r\sin(\phi)\end{bmatrix}$$
$$0\le\phi\le2\pi \qquad 2.23$$
$$0\le\theta\le2\pi$$

The algebraic definitions for a cylinder centered on the origin is:

$$x^2+y^2-R^2=0$$
$$0\le z\le h \qquad 2.24$$

and the parametric definition is:

$$C(\phi,\theta)=\begin{bmatrix}x\\y\\z\end{bmatrix}=\begin{bmatrix}(R\cos(\theta)\\(R\sin(\theta)\\\phi\end{bmatrix}$$
$$0\le\theta\le2\pi \qquad 2.25$$
$$0\le\phi\le h$$

Generally the parametric definition is used when it is necessary to transform a tool definition into a different coordinate system since this type of definition lends itself to matrix operations. An algebraic definition is used when it is necessary to intersect the tool with another surface.

2.5.4.1 Positioning a Ball Nosed End Mill

Historically, sculptured surfaces have been machined using ball nosed end mills. The easiest way to position a ball nosed end mill is to offset the tool center a distance equal to the cutter's radius along the surface normal, as shown in Figure 2.15. Provided the minimum radius of curvature of the surface is greater than the radius of the ball-nosed end mill no gouging will occur. The tool orientation has no effect on the geometry of the cutting surface of the ball relative to the design surface. In other words scallop size reduction can not be accomplished by changing the orientation of the tool with a 5-axis machine. The only way to reduce scallop size for a given tool path is to select a tool with a larger radius.

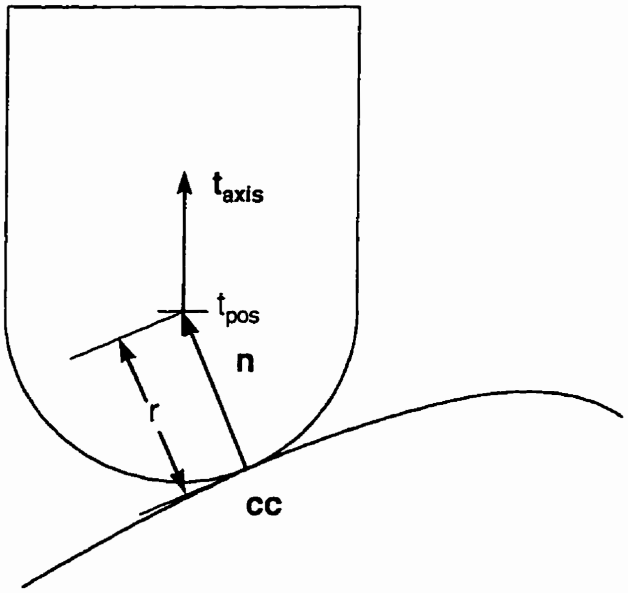


Figure 2.15 Positioning a ball nosed end mill on a surface.

A surface machined in 3-axis or 5-axis with a ball nosed end mill will have the same surface. However, benefits from 5-axis machining with a ball nosed tool are still possible when the problems of center cutting and accessibility are considered. Center cutting occurs when the cutting edge near or at the tool axis is required to cut material. This is a common problem during operations such as plunging and ramping down. Near the tool axis the cutting speed approaches zero resulting in high cutting forces. In 5-axis the tool can simply be inclined in the feed direction to eliminate center cutting. The leading edge of the tool will now be cutting instead of the tool center.

5-axis machining can also solve accessibility problems. In 3-axis machining, the tool may not be able to reach a location in a workpiece because the tool must approach the location along the z-direction. For example, when milling the ports in a high performance engine block. This problem can be overcome by approaching the workpiece from a different direction using a 5-axis machine. For instance, in 1991 Tekeuchi and Idemura[54] addressed the problem of accessibility using a solid modeling approach. Their approach assumed a fixed inclination angle for machining. The machining process was then simulated using solid modeling techniques. At every cutter location, interference checking was performed. If interference was detected, the operator was prompted to reposition the tool in the appropriate manner. This approach allowed the authors to machine exceedingly complex shapes such as an impeller out of a single block of material.

2.5.4.2 Inclining a Flat or Toroidal End Mill

Most current research on 5-axis tool positioning strategies is focused on machining a surface with a flat or toroidal end mill inclined relative to the surface normal as shown in Figure 2.16. This inclination angle is often referred to as the Sturz angle (ϕ). This approach was made popular by Vickers and Bedi [56] and Vickers and Quan [57]. The authors pointed out that the effective cutting shape of a flat end mill is an ellipse when projected into a plane perpendicular to the feed direction containing the surface normal. At the cutter contact point, this ellipse could be approximated by a circle whose radius is referred to as the effective radius. By varying the inclination of the tool, ϕ, the effective radius, r_{eff}, of the tool could be varied as shown in the following equation.

$$r_{eff} = \frac{R}{\sin\phi} \qquad 2.26$$

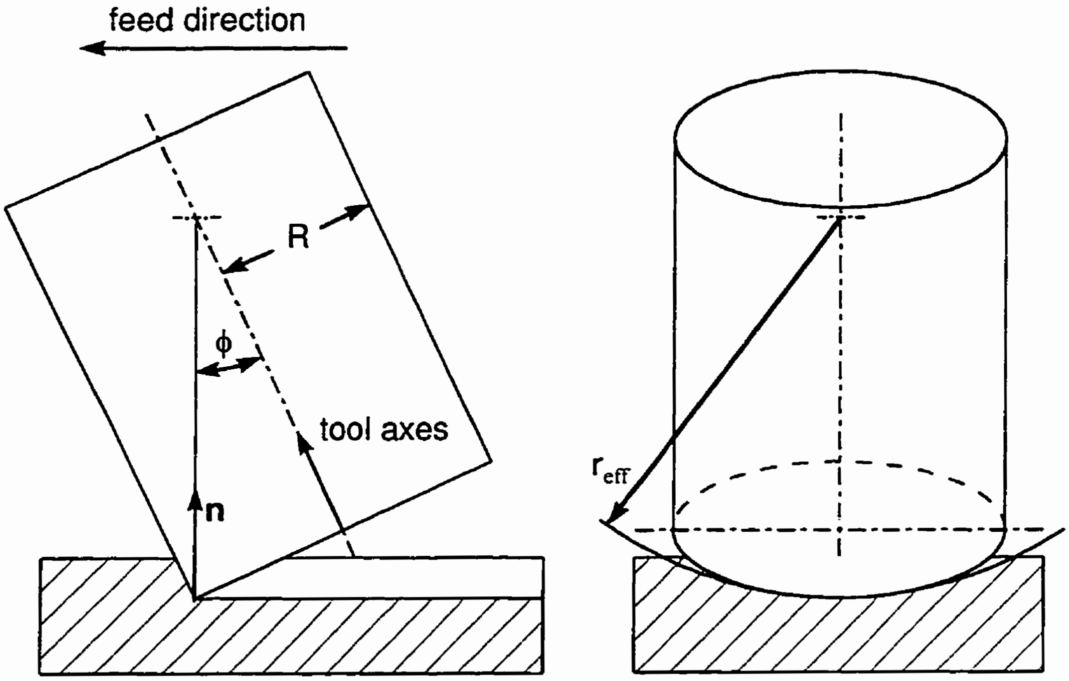


Figure 2.16 Machining with an inclined tool.

A ball nosed end mill, on the other hand, has a constant effective radius. An inclined flat end mill could be used to machine the surface as effectively as a much larger ball nosed tool. The authors performed a number of cutting tests on flat surfaces and ship hull molds. The results demonstrated that machining with an inclined end mill was considerably faster than machining with a ball nosed end mill. This approach has been shown to be highly effective in numerous papers and has been adopted by several high end commercial CAD/CAM systems [30].

The two main drawbacks to this method are the arbitrary method used to select an inclination angle and the use of a constant inclination angle for an entire surface. In a commercial package, the user is typically prompted to select an inclination angle for the entire surface. Experience has shown that an inclination angle between 3° and 6° is reasonable. Too small an angle would produce gouging and too large an angle would produce excessive scallops. Furthermore, a constant inclination angle could not account for local surface variations. Subsequent investigations were primarily concerned with these two issues.

A number of methods have been proposed for adjusting the tool inclination angle in an automatic or semi-automatic fashion. Cho et al.[6] modified the inclination angle in 1993 by using a Z-map technique. In the Z-map technique, the XY plane is represented by a discrete set of (x, y) points. The tool and workpiece are represented by discrete z values at each (x, y) location. Interference was checked by detecting if any of the z values for the desired surface were above the z values of the tool. Tool inclination adjustments were performed in a semi-automatic fashion by rotating the tool about the cutter contact point based on the weighted average of the interfering points. This process was repeated twice at every cutter location point. If interference still occurred the programmer was prompted to adjust the tool orientation manually. In 1994 Li and Jerard [32] pointed out that representing solids as discrete sets of point is very inefficient and that high model resolution was not possible because of data size restrictions. Li and Jerard's approach was to represent the tool and surface accurately as faceted surface models. By considering interference between points, lines and planes, Li and Jerard were able to adjust the tool inclination automatically. The problem with these automatic tool adjustment techniques is that they can only detect and adjust tool positions that produce gouging. There is no mechanism for detecting and improving mediocre tool position. In other words a tool position may produce a large scallop without being detected and improved.

In 1992 Jensen and Anderson [24] proposed a method for calculating an optimal tool angle based on local surface curvature. The local geometry of a surface near a point, **P**, is characterized by its maximum and minimum curvatures, κ_1 and κ_2 respectively, as shown in Figure 2.17. These curvatures describe circles of radii ρ_1, and ρ_1 in two perpendicular planes. The directions of maximum and minimum curvature, λ_1 and λ_2, form a right handed coordinate system with the surface at **P**. See for example Faux and Pratt [14], Farin [13] or Appendix A of this thesis for a complete description of the differential geometry of surfaces.

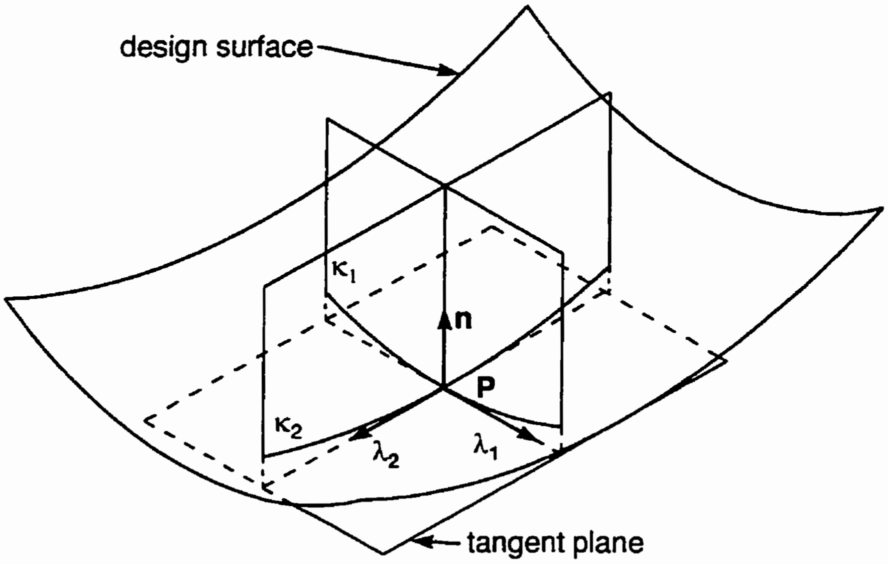


Figure 2.17 Curvatures of a point on a surface.

Curvature matching is shown in Figure 2.18 The tool is placed on the surface such that the feed direction lines up with the direction of minimum curvature on the surface. The tool is inclined in the direction of minimum curvature such that the effective radius of the tool at the cutter location equals the minimum radius of curvature of the surface. Jensen and Anderson [24] also noted that the profile of a torus is a 4th order curve while the profile of an end mill is only a 2nd order curve. Therefore, a better match between the surface and a toroidal end mill should be possible.

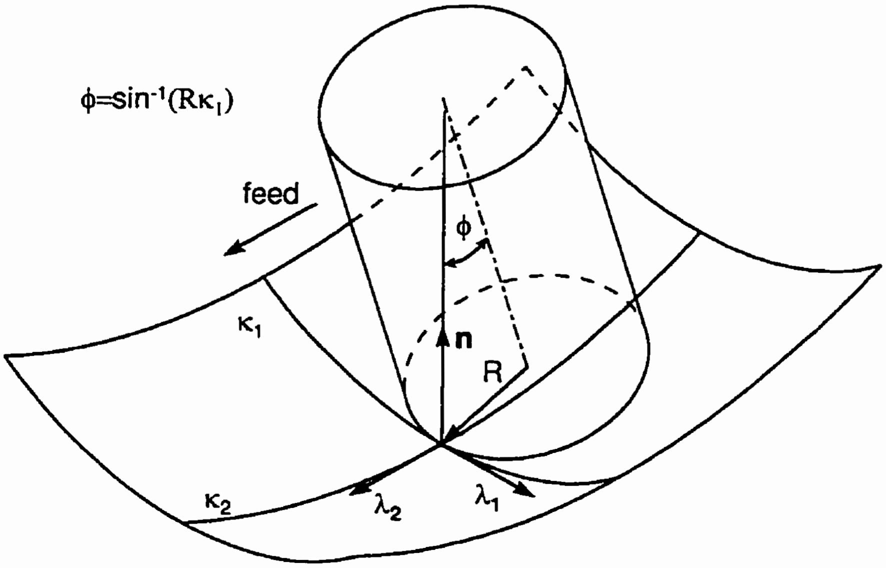


Figure 2.18 Curvature matching with a flat end mill.

In 1993 Jensen et al. [25] extended this work to the toroidal end mill. See Figure 2.19. However, the authors made no attempt to machine an actual workpiece. Had they done so, they would have realized that it is not always practical aline the feed direction with the directions of minimum curvature because these lines tend to follow irregular curves producing impractical tool paths [30]. Rao et al. [47, 48] developed a similar technique in 1994 that they called the Principal Axis Method (PAM). They used their technique to machine various surface patches and investigated the effect of tool path direction on the technique. Rao et al. [48] also compared the technique to 3-axis machining with a ball nosed tool of the same dimensions.

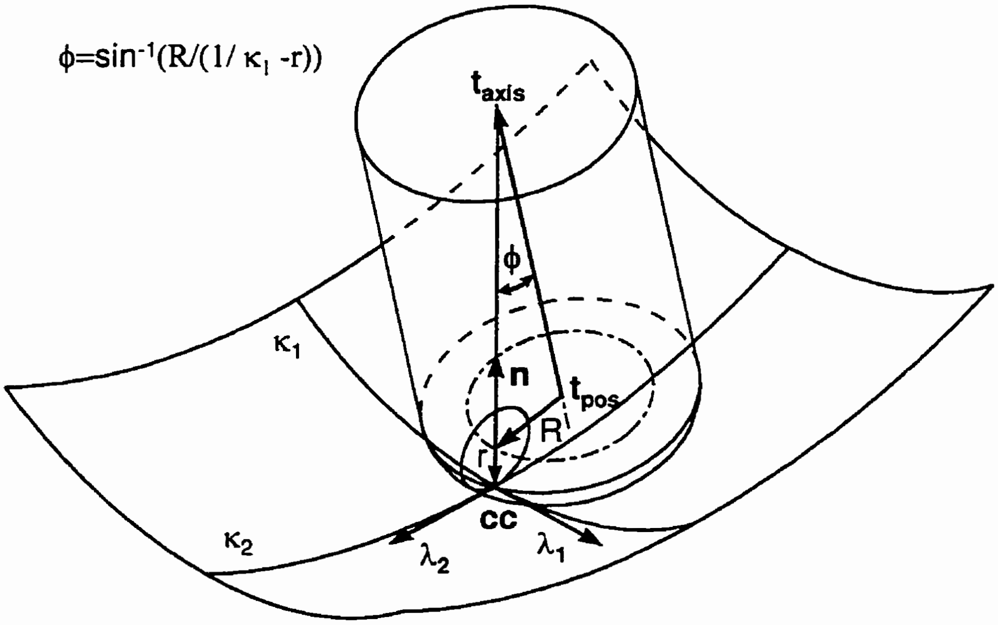


Figure 2.19 Curvature matching with a toroidal end mill.

In 1994 Kruth et al. [30] used curvature matching as a first approximation for their tool inclination calculation. The authors recognized the importance of the workpiece global geometry not just local curvature. Even with curvature matching, gouging may still occur. Kruth et al. checked to see if any portion of the cutting tool was penetrating the desired surface by numerically approximating the distance between the tool and the surface. The tool inclination angle was altered based on the location and depth of gouging.

Another tool positioning strategy called curvature catering was proposed by Wang et al. [61, 62] in 1993 for a cone shaped tool shown in Figure 2.20. The authors derived their theory by intersecting the plane formed by the bottom of the tool and a third order Taylor's series approximation of the surface. The circle formed by the bottom of the tool and the resulting intersection line is matched as closely as possible. The results of this analysis are the same as those obtained by Jensen and Anderson [24, 25] and Rao et al. [47, 48] for a flat end mill. The best inclination angle occurs when the effective radius of the tool is matched to the minimum radius of curvature of the surface. Li et al. [33, 34] deal with the issues of tool path generation, tool pass interval calculation and gouging when using curvature catering.

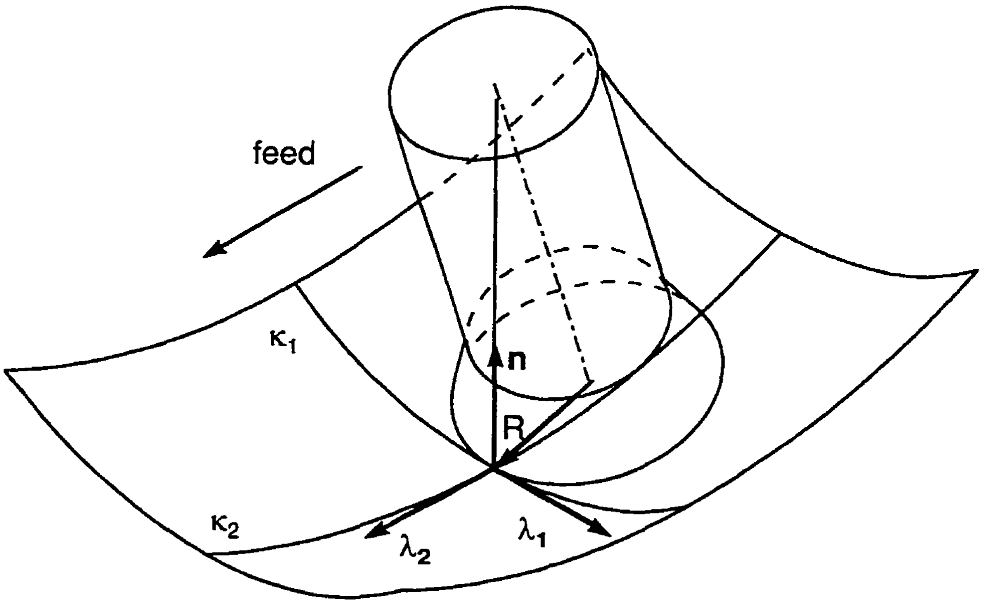


Figure 2.20 Curvature catering technique.

All the tool positioning strategies in the literature can be classed as single point tool positioning strategies. Surface properties at a single point on a design surface are used to calculate a tool position. In a sense, the surface underneath the tool is represented entirely by this single point. The effectiveness of the tool positioning strategy depends on the accuracy of this assumption. For instance, tool positions that consider only the position and surface normal are essentially assuming that the surface is a plane in the vicinity of the tool. These tool positioning strategies will require small tool pass intervals and will be subject to large amounts of gouging for highly curved surfaces. The most sophisticated tool positioning strategies in the literature, Principle Axis Technique and curvature catering, use curvature information in addition to position and normal information. These techniques assume that the surface beneath the tool is second order. The result is tool paths with larger tool pass intervals and less gouging. The Multi-Point machining strategy proposed in this thesis extract more information from the surface by considering the position, normal and curvature at several points on the surface. By combining this information, the tool positioning strategy results in tool paths with even larger tool pass intervals and virtually no gouging.

2.6 NC Simulation, Verification, and Correction

As with any other type of program, NC codes have to be debugged to avoid milling errors such as gouging, undercutting, collision with fixtures, etc. The two traditional methods used to check NC code depend heavily on human observation and intuition. The most common method is to perform trial cuts on a soft, inexpensive material under the observation of a skilled technician. The trial workpiece can then be measured manually or with a coordinate measuring machine CMM to determine the acceptability of the trial piece. Generally, several modifications and trial pieces are needed. This is both a labor and capital intensive process. The second common approach is to visually check a drawing of a tool path. The judgment of the acceptability of a tool path is largely dependent on the skill of the part programmer. A visual inspection of the tool path is generally followed with trial cuts.

Recognizing the need to automate the process of checking NC code, a large body of research has been devoted to this task. Jerard et al. [23] define the terms simulation, verification and correction to describe the important aspects of software designed to validate NC code. To this list I would add gouge detection. Simulation is used to model the geometric aspects of metal removal. Verification is used to determine if the simulated surface meets the requirements of the design. Gouge detection is the process used to determine if the tool will penetrated the design surface. The techniques used for simulation and validation are often used for this purpose. If an error is found in a tool path, a correction strategy is invoked to modify the offending tool position.

2.6.1 Simulation and Verification

Solid modeling is the classic approach used for simulation and verification. Voelcker and Hunt[58] did an exploratory study on the feasibility of using constructive solid modeling for the simulation of NC programs and Fridshal et al.[16] modified the solid modeling package TIPS to perform NC simulations. The general procedure used in most solid modeling based NC simulators can be summarized in three steps. First, solid models representing the workpiece and the swept volume of the tool as it moves between tool positions are constructed. The swept volume of the tool is then subtracted from the workpiece model using a Boolean operation. This "as milled" part is then compared to the model of the desired part using Boolean operations for verification. The difficulties with this process lie in the mathematics of swept volumes and the number of Boolean operations required.

The mathematical description of a swept volume can be calculated by invoking the theory of envelopes [14]. When a tool undergoes simultaneous motion in 5-axis the resulting path of the tool describes a highly non-linear path in space and time, $\mathbf{P}(x, y, z, t) = 0$. Coordinates, x, y, z are used to describe the shape of the tool at any instance and t describes the position of the tool on the tool path. The envelope or swept volume can then be calculated by determining the surface that is tangent to $\mathbf{P}(x, y, z, t) = 0$ at every instant in time. Due to the complexity of the result, these equations are impractical to solve for all but the simplest cases. The problem is further compounded by the number of volumes needed to simulate a tool path that may contain tens of thousands of individual tool positions.

For these reasons researchers have turned to more approximate techniques of simulating and verifying tool paths. View based methods have been proposed by Wang and

Wang [59, 60], Van Hook [55], and Atheron et al. [3]. These methods use a variation of the z-buffer method used in computer graphics as illustrated in Figure 2.21. In these methods a vector normal to the computer graphics screen is drawn through each pixel. The intersection points of these vectors with the workpiece model are stored in a z-buffer. Metal removal is simulated by performing Boolean operations between the tool and the z-buffer.

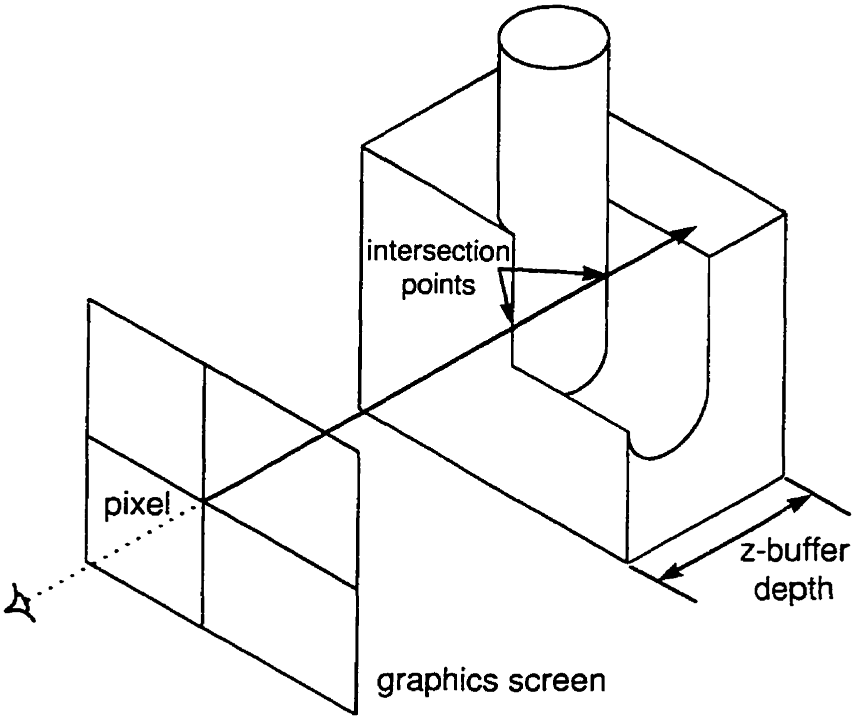


Figure 2.21 View space based simulations.

View based simulations are very fast and interactive. The user can examine the simulation by panning, zooming and rotating the image. However, it is not always easy to detect and quantify errors. Errors not visible in the graphic screen can not be detected. Generating a new view requires rerunning the entire simulation. Further, the user must rely on his/her eye to detect the error and determine its magnitude. Kim et al. [27] suggest a modified z-buffer method they call a z-map for performing simulation and verification that eliminates this problem. In the z-map techniques, the simulation and display of the results are separated. The workpiece is modeled as a set of discrete columns in Cartesian space as shown in Figure 2.22. Each z-column is stored as a number in an array called the z-map. Simulation is performed by intersecting lines defined by the z-columns with the tool motions. After each intersection the value in the z-map is compared with the intersection result. If the intersection result is smaller than the z-map value, the value in the z-map is replaced by the intersection value. When the simulation is completed, the z-map can be displayed to visually inspect the results and compared with the design surface to determine the magnitude of surface deviations.

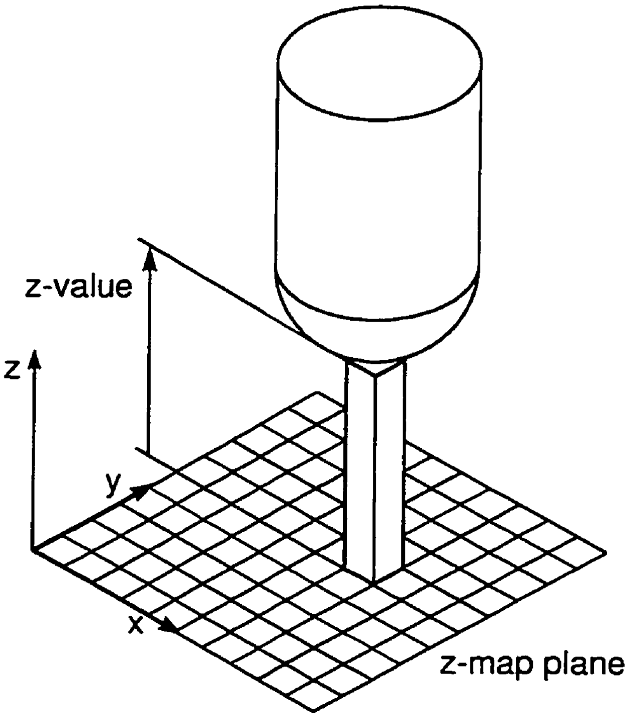


Figure 2.22 The Z-map technique.

The drawback with this type of simulation and any other z-buffer based simulation is that the resolution of the simulation depends on the size of the z-buffer. For instance, a 16 bit z-buffer can hold up to 65536 levels. To overcome this problem, a number of methods based on the "mow the grass" concept have been proposed and were described by Drysdale et al. [12] and Jerard et al. [21-23]. In this method, shown in Figure 2.23, vectors extend (grow) from the desired surface at regular intervals. During the simulation these vectors are intersected with the tool motions and the length of the vector is reduced to the intersection point. An analogy can be made with mowing a field of grass. As the simulation continues the blades of grass represented by the vectors are "mowed down". On completion the amount of material left above the surface or gouged out of the surface can be computed from the final length of the grass. This information can be displayed by color mapping the grass height onto the design surface.

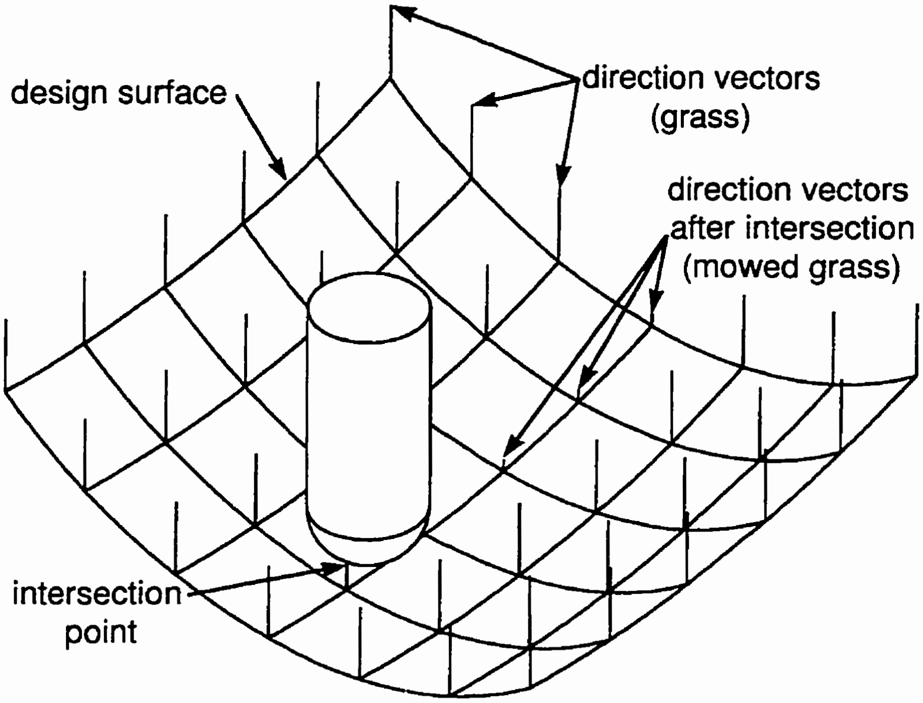


Figure 2.23 The "mow the grass" concept.

Although the approximate methods of simulation and verification are computationally less demanding than solid modeling approaches they still require considerable computer resources. They involve intersecting lines with objects in order to model metal removal. To have a reasonable representation of the final machined surface a large number of vectors, blades of grass, or positions in a z-buffer are required, which can quickly lead to large memory requirements. The number of these vectors depends on the size of the workpiece being modeled and the size of the expected surface deviations. In my experience, if scallop height and gouging are the phenomena of interest, a spacing between vectors of at least 10 times smaller than the tool pass interval is required for a good representation of the machined surface. For example, a workpiece 0.5 m by 0.5 m milled with a 25.4 mm ball nosed end mill with a tool pass interval of 0.5 mm will require a vector every 0.05 mm for a total of one million evenly spaced vectors. In addition to the memory requirements of the model, considerable time must be spent on performing the intersections. The number of intersections depends on the number of tool positions and the number of vectors underneath the tool at each position. Given that that tool positions are typically spaced at least every 1.0 mm along a tool path, the example workpiece will require 500 000 tool positions. Each tool position will have approximately 2000 vectors in its shadow. The resulting simulation will require 10^9 intersection calculations.

2.6.2 Gouge Detection and Correction

Often NC programmers do not require or have the time for a full simulation and verification cycle. They are not concerned with producing a model of the machined surface. Instead they concentrate on checking to see that each tool position is within tolerance of the surface. The result of their simulation will not tell the users the expected size of the scallops but will ensure that the surface is not gouged, undercut or out of tolerance. Examples of research in this area include the works of Takeuchi and Idemura [54], Li and Jerard [32], Rao et al. [47], Kruth and Klewais [30], and McLellan et al. [40]. These authors have all adopted different strategies for 5-axis gouge detection and correction.

Takeuchi and Idemura [54] use a boundary representation (B-rep) for the simulation of the tool motion over the design surface. A set of check points are defined on the tool. At every tool position, an inside/outside test is performed between the check points and the model of the workpiece to detect if gouging has occurred. Automatic correction is accomplished by rotating the tool about the cutter contact point. At the gouging position, the check points are used to determine the direction and magnitude of the rotation. If this process fails, the user is prompted to manually correct the offending tool position. Li and Jerard [32] use a similar approach to detect gouging when generating tool paths for 5-axis machining with a flat bottom end mill.

Rao et al. [48] use a variation of the "mow the grass" technique in the immediate vicinity of a cutter contact point to perform gouge detection when using the principle axis method. This tool positioning strategy relies on the minimum radius of curvature of the surface to calculate a tool inclination angle. A smaller radius of curvature produces a larger tool inclination angle. Theoretically, gouging should not occur when using this tool positioning strategy provided that the curvature of the design surface does not change in the region underneath the tool. A gouge implies that the curvature underneath the tool has increased. Therefore, the algorithm developed by Rao et al. [48] artificially increases the curvature used in the tool positioning calculations incrementally till gouging is eliminated.

McLellan et al. [40] have developed a geometric algorithm for determining how close the tool is to a surface or how far the tool has penetrated into the design surface. The algorithm is loosely based on the theory of intersections discussed in Markot and Magedson [38]. Surfaces in close proximity contain so called characteristic points. These points always occur in pairs; one point on each surface. The surface normals at these points are collinear. In a local sense, the closest points between two non-intersecting surfaces are the characteristic points. Similarly, in a region were two surfaces intersect, the characteristic points are the points where maximum penetration occurs. It should be noted that surfaces can have many different pairs of characteristic points. The algorithm developed by McLellan et al. [40] searches the tool's cutting surface and the design surface for characteristic points. If a characteristic point results in gouging, the location is stored for correction purposes. When a gouge is detected, the user has the option of using three different gouge avoidance strategies. The preferred method is to incline the tool. Based on the position and depth of the gouge, an optimal tilting direction and angle are calculated. If this strategy fails, the tool may also be lifted in the direction of the tool axis or the normal to the surface. Although lifting the tool is the surest way to eliminate gouging, McLellan et al. [40] note that unwanted material will be left on the surface in the region of the cutter contact point.

The difficulties with this gouge detection algorithm based on characteristic points arise when there are more than one characteristic point. Such a situation arises when the tool is touching a surface tangentially at a cutter contact point and gouging the surface at another point. The algorithm may not converge to the point of interest (gouge point). Instead the algorithm may converge on the cutter contact point and the presence of gouging will not be detected. McLellan et al. [40] have solved this potential problem by restarting the algorithm at different points on the tool. If the algorithm converges to the cutter contact point from every start point there is assumed to be no gouging.

2.7 Summary

5-axis machining has been demonstrated to be an efficient way of producing sculptured surfaces; it reduces the machining time and improves the surface finish. Tool path planning focuses on determining the tool pass interval and spacing between tool positions such that predictable scallops are evenly distributed across the machined surface. The main challenge of this research is to develop methods of predicting scallop height based on tool pass interval. For the most part this issue has been resolved for ball nosed end mills but is still a problem when using a flat or toroidal end mill. Research efforts are underway to develop new strategies for tool positioning that matches the tool profile closer to the design surface and thus reduce the need for further finishing operations. Tool positioning strategies that use curvature information such as curvature matching, the Principle Axis Method and Multi-Point Machining method, in particular were found to be superior to the currently used strategies. A final verdict, however, will require extensive testing on a large number of surfaces. Research efforts are also underway to develop faster techniques for tool path simulation and verification. The ultimate objective of these techniques is to help detect and avoid gouging and interference; a major concern that casts skepticism on 5-axis machining. The current techniques require a considerable investment in computer hardware to be of practical use in industry. Even with today's fast computers a simulation and verification cycle can take several hours. More research is needed to develop and implement techniques for simulation and automatic tool path validation and correction.

Industrial acceptance of 5-axis machining will materialize only if the above research efforts lead to satisfactory solutions that can be amalgamated into existing CAM packages. In addition to software, acceptability will also depend on building more rigid machines with controllers capable of more sophisticated methods of interpolating between points on a tool path.

Chapter 3
Multi-Point Machining of Spherical Surfaces

As noted in Chapter 2, all previous 5-axis tool positioning strategies have attempted to maximize metal removal by considering only a single point on the surface. It is argued here that a tool positioning strategy that considered more points of contact between the tool and the surface would be more effective. In this chapter, this argument is substantiated by applying the multi-point machining concept to spherical surfaces [64] and comparing it to 3-axis machining with a ball nosed end mill. The simple geometry of the sphere makes it relatively easy to achieve multi-point contact. This idealized case will be used to demonstrate the concept. Latter in this thesis, the sphere example will be used to confirm the validity of the general approach.

Spherical surfaces are often used in the design of joints, molds, valves, bearings, etc. These surfaces are easily produced on a lathe. However, if the workpiece is large and complex, or the spherical surface is not centered on the workpiece, it may be more desirable to produce the surface on a milling machine. For instance, hemi-spherical combustion chambers for a high performance engine block could be produced this way.

Multi-Point Machining of spherical surfaces is best explained by the "drop the coin" concept, illustrated in Figure 3.1. If you place a coin in a spherical dish, every point on the coin's circumference will contact the surface. In geometric terms, a sphere and a plane defining the bottom of the coin will intersect in a circle. As the coin moves around the surface, every point on the coin's circumference remains in contact with the surface. In the proposed technique, the coin is replaced by the flat bottom of an end mill. The concept can be extended to toroidal end mills by identifying circles of tangency between a torus and a spherical surface. In this chapter, the theory is developed for the general case of the toroidal end mill. The flat end mill is considered to be a special case of a toroidal end mill with an insert radius, r, of zero.

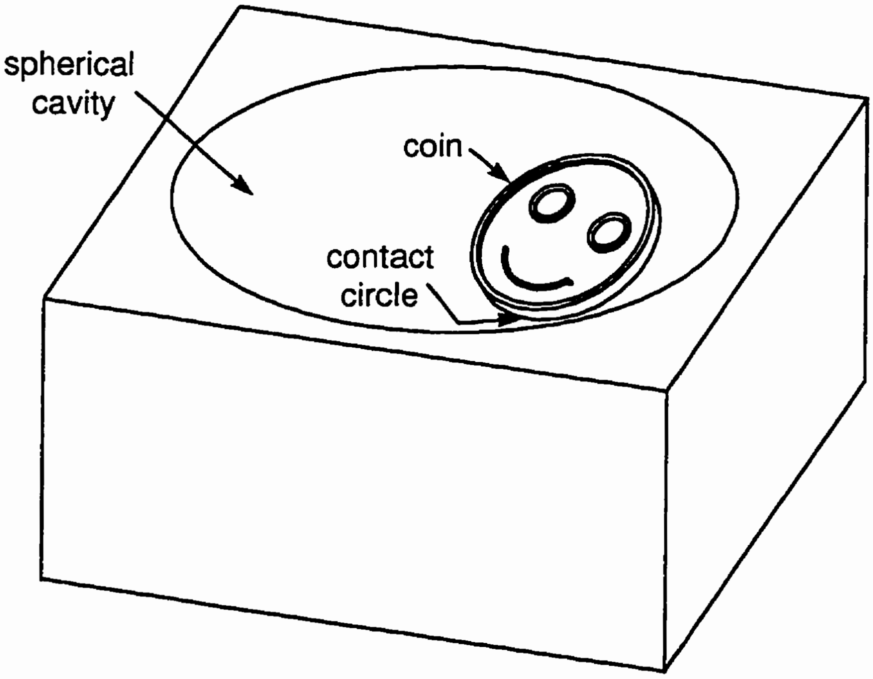


Figure 3.1 The "drop the coin" concept.

3.1 Multi-Point Machining of Spherical Cavities

In the "drop the coin" concept, a coin placed in a spherical dish will contact the surface at every point on its circumference. This circle will be referred to as the contact circle and has a radius of R_c. As the coin moves on the surface, the radius of the contact circle and the distance between the center of the coin and the surface is constant. In other words the center of the coin will travel on a sphere offset from the surface by a constant radial distance. To convert the "drop the coin" concept into a useful tool positioning strategy requires calculating the offset distance and the contact circle radius. The tool will move on a sphere offset from the design surface. Provided the contact circles overlap, there should be no scallops. In other words, the tool pass interval should be less than or equal to the diameter of the contact circle.

Figure 3.2 shows a cross section of a toroidal end mill placed on a spherical cavity. In order to achieve a circle of contact, the tool axis is aligned with the surface normal **n** at point **P** and the center of the tool is offset from the surface along the normal by a distance, d. When these conditions are met, the center of the sphere forms a right circular cone with the contact circle whose cone angle is β/2. This geometry can be used to calculate the correct position of the tool center.

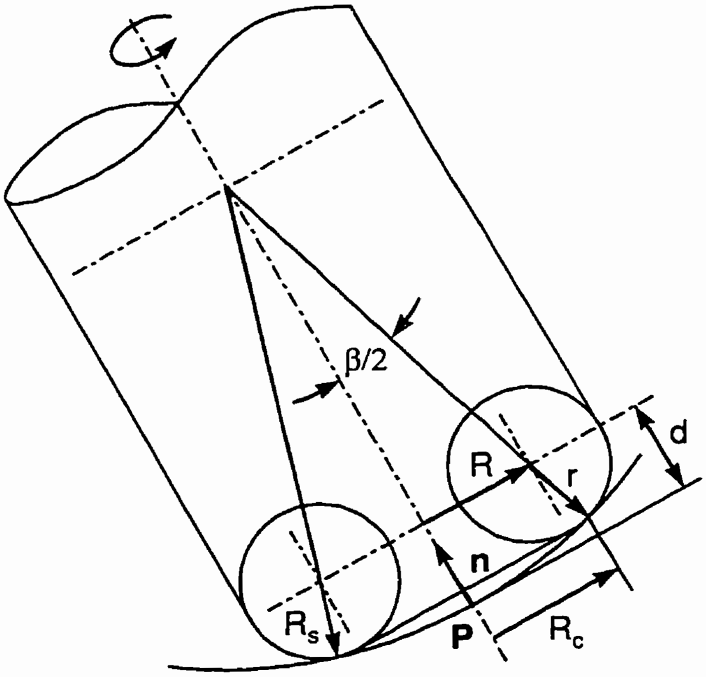


Figure 3.2 Multi-point tool positioning for a spherical cavity.

The contact circle radius can be determined using similar triangles.

$$R_c = R\left(\frac{R_s}{R_s - r}\right) \qquad 3.1$$

Notice that the contact circle radius increases as the tool parameters, R and r, increase. This implies that larger tool pass intervals and reduced machining time can be achieved by increasing the tool size. Once the contact circle radius is known the cone angle can be calculated by:

$$\beta/2 = \sin^{-1}\left(\frac{R_c}{R_s}\right) \qquad 3.2$$

and the offset distance is:

$$d = R_s - (R_s - r)\cos(\beta/2) \qquad 3.3$$

3.2 Multi-Point Machining of Spherical Domes

The corollary of the "drop the coin" concept is the "drop the donut" concept. Instead of placing a coin in a dish, a donut is placed on a dome. A circle of contact occurs on the interior portion of the donut. This concept can be used to machine convex surfaces by replacing the donut with a toroidal cutter. The desired circle of tangency between the tool and the sphere can be achieved by aligning the tool axis with the surface normal **n** at point **P,** and offsetting the tool along the surface normal by d as shown in Figure 3.3. When these conditions are met, the center of the sphere forms a right circular cone with the contact circle whose cone angle is β/2. This geometry can be used to calculate the correct position of the tool center. From similar triangles, the radius of the circle of contact is:

$$R_c = R\left(\frac{R_s}{R_s + r}\right) \qquad 3.4$$

Once the contact circle radius is known the cone angle can be calculated by:

$$\beta/2 = \sin^{-1}\left(\frac{R_c}{R_s}\right) \qquad 3.5$$

and the offset is:

$$d = R_s - (R_s + r)\cos(\beta/2) \qquad 3.6$$

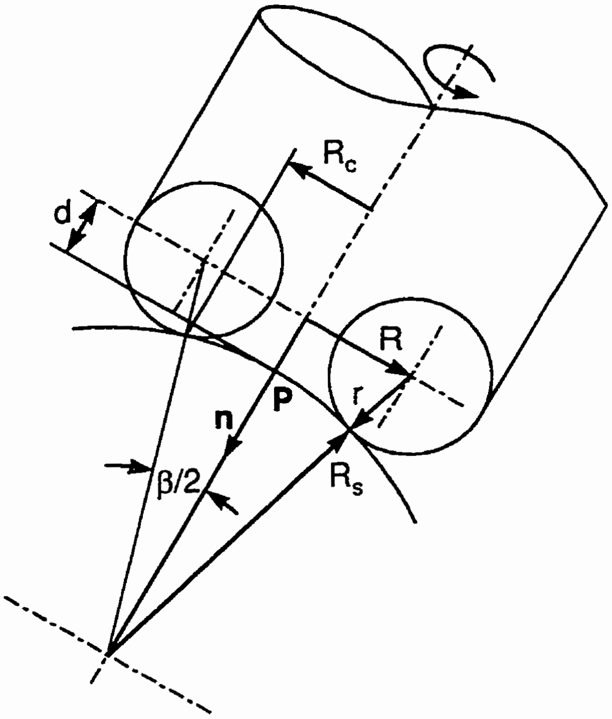


Figure 3.3 Multi-point tool positioning for a spherical dome.

By examining the equation of the contact circle radius, a few conclusions can be made about tool selection. As the separation between the two inserts increases the contact circle radius increases until the contact circle radius equals the sphere radius. At this point the sphere would slip between the inserts. In practice the tool size is limited by the clearance between the sphere and the bottom of the tool holder. Provided this criterion is met, the tool pass interval can be maximized and machining time minimized by selecting a tool with the largest torus radius and smallest inserts available.

3.3 Evaluation of Proposed Method

In this section the proposed technique will be evaluated by comparing it to machining with a ball nose end mill, which is the most commonly used surface machining technique. The most important consideration in this comparison is the amount of time required to machine the surface to a specified surface finish. In order to find a basis for comparison, a measure of time and a reasonable surface finish must be determined.

The time required to machine a surface depends on the feed rate, the rigidity of the milling machine, and the length of the tool path. This comparison is only concerned with the effect of the tool path length. The other factors are machine dependent and can only be improved by building a better milling machine. On the other hand, the tool pass interval is specified during tool path planning. A tool path with large tool pass intervals will require fewer tool passes. Since the lengths of the tool passes are all about the same, a larger tool pass interval will result in a tool path whose overall length is shorter.

Surface finish is dependent on the tool positioning strategy and the mechanics of machining. In the proposed approach scallops are eliminated provided that the tool pass interval is smaller than the diameter of the contact circle. However, the surface finish is still dependent on the cutting conditions, machine rigidity, and the controllers ability to interpolate between tool positions. Experience has shown that these factors generally result in surface deviations of at most 0.025 mm in 5-axis machining. Therefore a comparable technique should produce a scallop height within this tolerance.

To complete this comparison expressions are needed to relate tool pass interval to scallop height. It has already been established that the proposed techniques does not produce any scallop provided that the tool pass interval is equal to the diameter of the contact circle for the proposed technique. In the next sections the tool pass interval expressions will be developed for the ball nosed tool.

3.3.1 Tool Pass Interval Calculation for a Ball Nosed Tool Machining a Spherical Cavity

Figure 3.4 shows the geometry necessary to calculate the tool pass interval when machining a spherical cavity of radius, R_s, with a ball nosed end mill of radius, r. The angular separation between the two adjacent cutter locations **cl_i** and **cl_{i+1}** is α. A scallop with a height, h, is left between the cutter contact points **cc_i** and **cc_{i+1}**.

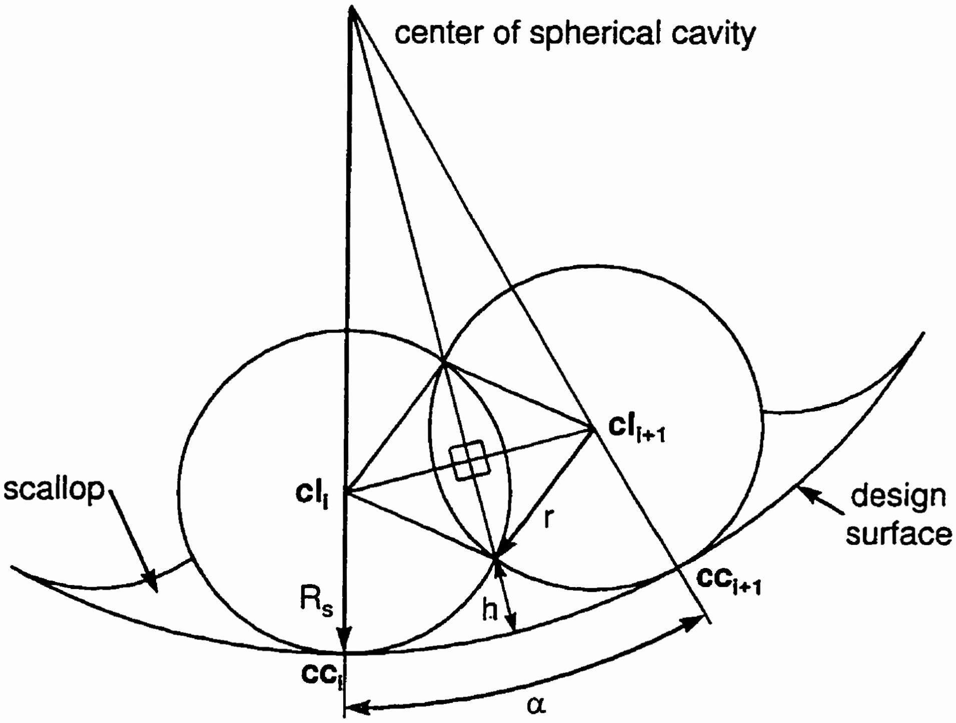


Figure 3.4 Tool pass interval calculation for ball nosed end mill machining a spherical cavity.

Using the geometry in Figure 3.4, the scallop height, h, can be expressed by:

$$h = R_s - (R_s - r)\cos\left(\frac{\alpha}{2}\right) - \sqrt{r^2 - (R_s - r)^2\sin^2\left(\frac{\alpha}{2}\right)} \qquad 3.7$$

Rearranging equation 3.7 and solving α we get:

$$\alpha = \cos^{-1}\left(\frac{h^2 - 2hR_s - 2rR_s + 2R_s^2}{2(R_s^2 - R_s r - R_s h + rh)}\right) \qquad 3.8$$

The tool pass interval as measured along the surface is then:

$$\chi = cc_{i+1} - cc_i = 2R_s \sin\left(\frac{\alpha}{2}\right) \qquad 3.9$$

3.3.2 Tool Pass Interval Calculation for a Ball Nosed Tool Machining a Spherical Dome

The tool pass interval calculation for the spherical dome is similar to that of the spherical cavity. Figure 3.5 shows the geometry necessary to calculate the tool pass interval when machining a spherical dome of radius R_s with a ball nosed end mill of radius r. The angular separation between the two adjacent cutter locations **cl**$_i$ and **cl**$_{i+1}$ is α. A scallop with a height of h is left between the cutter contact points **cc**$_i$ and **cc**$_{i+1}$.

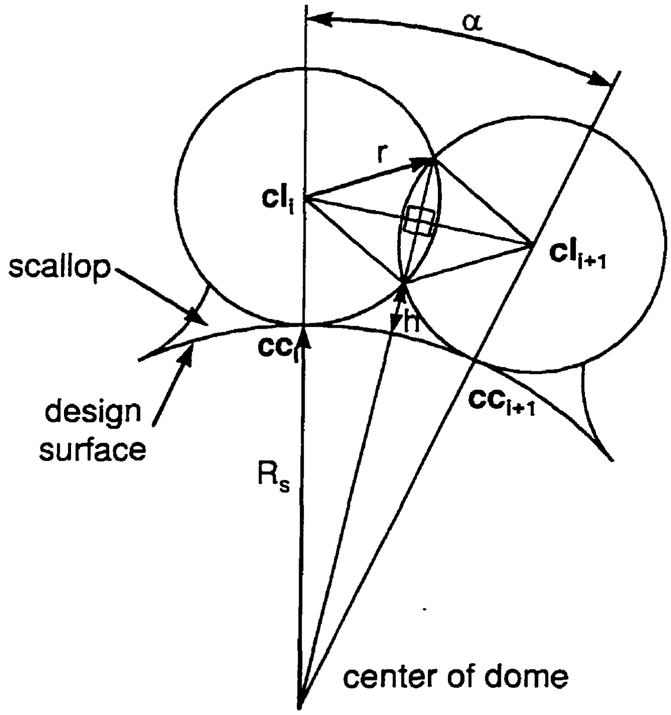


Figure 3.5 Tool pass interval calculation for ball nosed end mill machining a spherical dome.

Using the geometry in Figure 3.5, the scallop height, h, can be expressed by:

$$h = (R_s + r)\cos\left(\frac{\alpha}{2}\right) - R_s - \sqrt{r^2 - (R_s + r)^2\sin^2\left(\frac{\alpha}{2}\right)} \qquad 3.10$$

Rearranging equation 3.10 and solving for α yields.

$$\alpha = \cos^{-1}\left(\frac{2R_s^2 + 2R_s h + h^2 + 2R_2 r}{2(R_s^2 + R_s r + R_s h + rh)}\right) \qquad 3.11$$

The tool pass interval as measured along the surface of a spherical dome is then:

$$\chi = cc_{i+1} - cc_i = 2R_s \sin\left(\frac{\alpha}{2}\right) \qquad 3.12$$

3.3.3 Results of Numerical Evaluation

Table 3.1 compares the tool pass intervals required by the two techniques when milling a spherical cavity with a 50 mm radius given a surface tolerance of 0.025 mm. The parameters, χ_{ball} and χ_{mpm} refer to the tool pass interval for the ball nosed tool and multi-point technique, respectively. Each row corresponds to toroidal end mills and ball nosed end mills with the same diameter. The insert radius r of the toroidal cutter was arbitrarily selected to be the same size as the torus radius, R. Table 3.2 shows similar results for flat end mills.

Toroidal end mill			Ball nose end mill		Ratio
R	r	χ_{mpm}	r	χ_{ball}	χ_{mpm}/χ_{ball}
1.5875	1.5875	3.3	3.175	0.4	8.0
3.175	3.175	6.8	6.35	0.6	11.3
4.7625	4.7625	10.5	9.525	0.8	13.7
6.35	6.35	14.5	12.7	0.9	15.8
9.525	9.525	23.5	19.05	1.2	19.0
12.7	12.7	34.0	25.4	1.6	21.2

Table 3.1 Comparison of tool pass interval required to machine a spherical cavity to a surface tolerance of 0.025 mm using toroidal and ball nosed end mills. Note that all values are in mm.

Flat end mill		Ball nose end mill		Ratio
R	χ_{mpm}	r	χ_{ball}	χ_{mpm}/χ_{ball}
3.175	6.4	3.175	0.4	15.4
6.35	12.7	6.35	0.6	21.1
9.525	19.1	9.525	0.8	24.8
12.7	25.4	12.7	0.9	27.5
19.05	38.1	19.05	1.2	30.7
25.4	50.8	25.4	1.6	31.6

Table 3.2 Comparison of tool pass interval required to machine a spherical cavity to a surface tolerance of 0.025 mm using flat and ball nosed end mills. Note that all values are in mm.

These tables clearly show the speed advantage of the new technique over the conventional method. For the same diameter tool, a larger tool pass interval can be used that would result in approximately an order of magnitude reduction in tool path length and machining time. In addition to the speed improvement, a better finish should be achieved, which implies a reduction in bench work and improved dimensional accuracy.

Table 3.3 compares the tool pass intervals required when machining a spherical dome with a 50 mm radius to a tolerance of 0.025 mm. Each row corresponds to toroidal and ball nose end mills with the same diameter. The table shows the new technique should produce a spherical surface significantly faster then the old technique.

Toroidal end mill			Ball nose end mill		Ratio
R	r	χ_{mpm}	r	χ_{ball}	χ_{mpm}/χ_{ball}
1.5875	1.5875	3.1	3.175	0.4	8.0
3.175	3.175	6.0	6.35	0.5	11.3
4.7625	4.7625	8.7	9.525	0.6	13.8
6.35	6.35	11.3	12.7	0.7	15.8
9.525	9.525	16.0	19.05	0.8	19.3
12.7	12.7	20.3	25.4	0.9	22.1

Table 3.3 Comparison of tool pass interval required to machine a spherical dome to a surface tolerance of 0.025 mm using toroidal and ball nosed end mills. Note that all values are in mm.

3.3.4 Experimental Verification

In order to verify the proposed technique, it was implemented for spherical cavities and domes. The surfaces were defined parametrically as:

$$S(\theta,\phi)=\begin{bmatrix}x\\y\\z\end{bmatrix}=\begin{bmatrix}r\sin(\theta)\cos(\phi)\\r\sin(\theta)\sin(\phi)\\r\cos(\theta)\end{bmatrix}\qquad 3.13$$
$$0\le\theta\le\pi$$
$$0\le\phi\le2\pi$$

The tool path was developed on a sphere offset from the design surface by a distance d, along the surface normal. Circular tool passes on the offset sphere were developed by maintaining constant θ intervals between tool passes while varying ϕ between 0 and 2π. The θ interval was set to 90% of the cone angle β. The resulting cutter location files were post-processed using the theory developed in Section 2.4 of this thesis.

An aluminum dome and a dish were machined on two different types of 5-axis milling machines to confirm the validity of the proposed technique. Each piece was roughed using a three axis tool path with a ball nosed end mill. Each surface had a radius of 50 mm and a depth of 20 mm. The dish was cut at the Industrial Research and Development Institute (IRDI) in Mideland, Canada, on a JOTECH 5-axis wrist type CNC milling machine (JTH 50). The following cutting conditions were used; spindle speed = 20,000 RPM, feed = 0.0325 mm/tooth, 0.5 mm depth of cut, using a 2 flute 10 mm diameter end mill. The resulting tool path consisted of 1260 points. Linear interpolation was used to move between the points. The cutting time was 112 seconds. A photograph of the dish is shown in Figure 3.6. The surface finish of the dish was similar to that of a plane machined by an end mill and was smooth to the touch. The circulate marks on the surface are a result of the change in direction and speed at the end of each linear interpolation and the regular feed marks. For comparison the same sized dish was produced using a ball nose end mill and is shown in Figure 3.7. Each surface was measured on a Coordinate Measurements Machine (CMM). Both techniques produce reasonably round spherical cavities, but the 3-axis technique using a ball nose mill produces maximum deviations of approximately 50 μm whereas the new technique results in a mere 2.5 μm deviation.

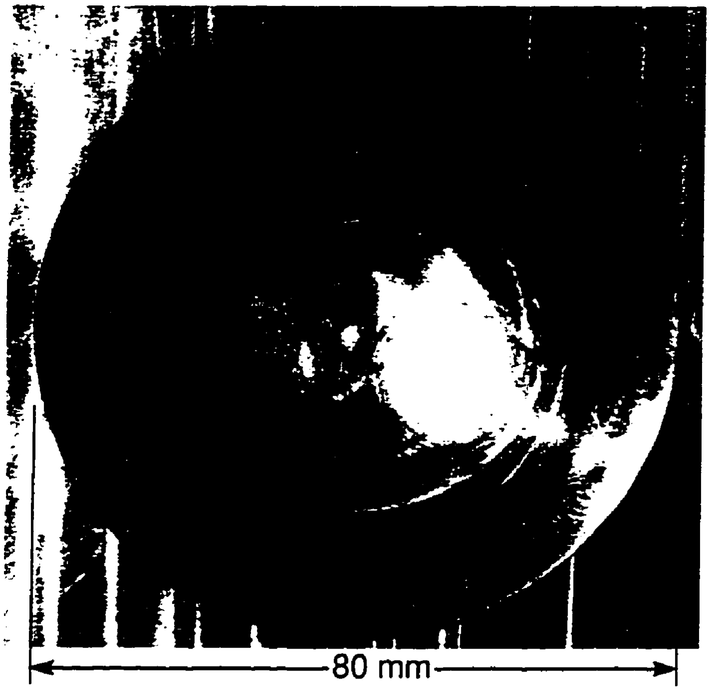


Figure 3.6 A spherical cavity machined with a toroidal end mill.

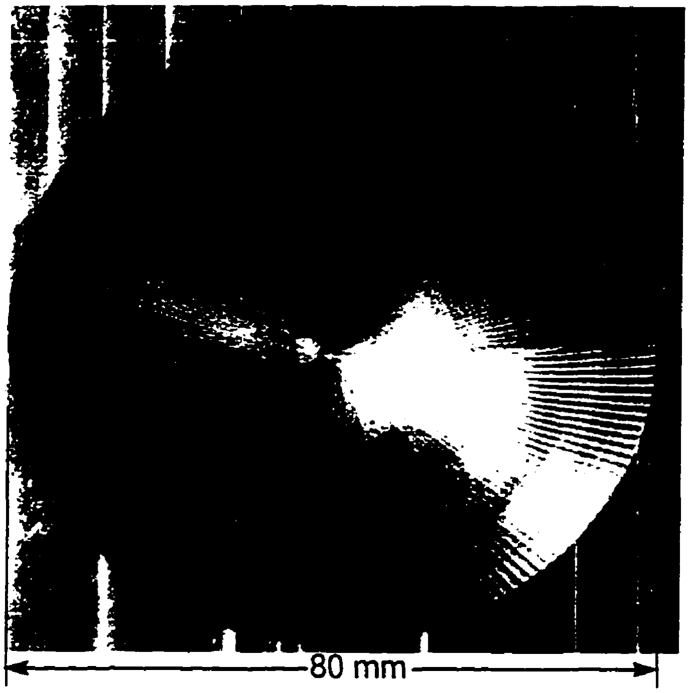


Figure 3.7 A spherical cavity machined with a ball nosed end mill.

The sphere was milled on a FADAL VMC 4020 5-axis tilt-rotary table type milling machine at the University of Western Ontario, London, Canada. The dimensions of the sphere were the same as the dish. The cutting conditions used were: spindle speed = 3300 RPM, feed = 0.0325 mm/tooth, 0.5 mm depth of cut, using a 25.4 mm diameter toroidal cutter with two 9.525 mm diameter carbide inserts. The tool path consisted of 490 points. Linear interpolation was used to move between the points. A photograph of the surface is shown in Figure 3.8. CMM measurements were also conducted on the sphere. The sphere had good roundness with a maximum radial error of 9 μm. This error was larger than the spherical cavity because further analysis showed that there was a significant misalignment(estimated at 0.2°) of the rotary table. This indicates that the proposed method is sensitive to this type of misalignment. This adverse effect, however, could be utilized in evaluating the misalignment, by mounting an accurate sphere and replacing the tool with a suitable probe.

Figure 3.8 Multi-point machining of a spherical dome.

3.4 Chapter Summary

Multi-point contact has been shown to exist between a spherical surface and a torus. Based on this result, a spherical cavity and dome were machined. The derivation of this tool positioning strategy was elementary because of the special nature of spherical surfaces. However, it was included in this thesis because it illustrates the power of the multi-point concept and provides independent conformation of the multi-point tool positioning techniques presented in the next chapters.

Chapter 4

Multi-Point Contact With a Toroidal End Mill

Chapter 3 demonstrated that tool positioning strategies based on multi-point contact could be extremely effective. The concept could be applied to spherical surfaces in a straight forward manner because the form of multi-point contact could be deduced from the "drop the coin concept" for spherical surfaces. However, the nature of multi-point contact between a tool and a more complex surface is not as obvious. In order to develop a multi-point tool positioning strategy the number and approximate location of the cutter contact points must be developed. In other words, how many cutter contact points should be expected and where are they located? Without the answers to these questions the development of the multi-point tool positioning strategy would have been virtually impossible. To answer these questions a system of equations was developed to model tool-surface contact. These equations were applied to a number of representative surfaces. The insight gained from this investigation will be used to develop a fast and efficient multi-point tool positioning strategy.

4.1 Classification of Surfaces based on Curvature for Multi-Point Machining

Tool positioning strategies are most influenced by the form of the surface underneath the tool. Most tool positioning strategies will not lead to gouging provided the curvature of the points underneath the tool meet certain requirements. For instance, a ball nosed cutter will not gouge a surface if the curvature under the tool is less than the curvature of the tool. Unfortunately, it is not feasible to check the curvature of all points beneath the tool. Instead, a single point under the tool is used to predict the form of the surface. The cutter contact point is generally used for this purpose. The success of the resulting tool position depends on how much the surface deviates from the predicted form.

Using differential geometry, the form of the surface under the tool can be classified by the maximum and minimum curvatures, κ_1 and κ_2 at one point. If κ_1 and κ_2 are both positive then the point is said to be elliptic. For example all points on an ellipsoid are elliptic. When both curvatures are equal as in the case of a sphere, the point is referred to as umbilic. If κ_1 and κ_2 have different signs, the point under consideration is called hyperbolic. All points on a hyperboloid are hyperbolic points. If either $\kappa_1 = 0$ or $\kappa_2 = 0$, the point is called parabolic. For example all points on a cylinder are parabolic. When $\kappa_1 = 0$ and $\kappa_2 = 0$, the point is called flat. All points on a plane are flat.

In the latter sections of this chapter, the contact between a tool and flat, spherical, elliptic, parabolic and hyperbolic surfaces will be examined. In classical differential geometry, these forms are determined from the perspective of a point. The region in the immediate vicinity of the point may indeed have this form but a tool may not be able to fit in this region. Therefore, these surface forms have to be determined from the perspective of the tool. Provided the curvature of the surface is less than the curvature of a sphere that just contains the cutting surface of the tool, the tool will be able to fit in that surface. The radius of this bounding sphere is equal to R+r, where R is the torus radius and r is the insert radius of the tool. Therefore, it will be assumed that the surfaces under consideration will always have curvature less then 1/(R+r).

4.2 Modeling Tool Surface Contact

When the cutting surface of a tool is in tangential contact with a surface, two criteria must be satisfied. First, a point on the tool and a point on the surface must share the same location in space. Second, the surface and the tool must share a tangent plane at this location. In order for the tool and surface to be tangent, the surface normals at these points must be collinear. If we consider the parametric definitions of the tool **T** = T(θ, ϕ) and a surface S = S(u, v) these criteria can be expressed mathematically as:

location criterion $$\mathbf{T}(\theta,\phi)-\mathbf{S}(u,v)=0 \qquad 4.1$$

tangency criterion $$\left(\frac{\partial \mathbf{T}(\theta,\phi)}{\partial\theta}\times\frac{\partial \mathbf{T}(\theta,\phi)}{\partial\phi}\right)\times\left(\frac{\partial \mathbf{S}(u,v)}{\partial u}\times\frac{\partial \mathbf{S}(u,v)}{\partial v}\right)=0 \qquad 4.2$$

Note that the bracketed terms in the tangency equation are vectors in the normal direction and that the cross product of these collinear vectors must be zero. The solution to these sets of equations produce a set of (θ, ϕ, u, v) parameters corresponding to the cutter contact points on the tool and on the surface. All tool positioning strategies must satisfy these equations simultaneously if the tool is to contact the surface without gouging. Note that these equations make no assumptions about the number and locations of the cutter contact points. Theoretically, a tool positioning strategy could be developed based on these equations. However it is too time consuming to solve these equations for every point on a tool path. Instead, the solution to these equations will give insight into the development of a multi-point tool positioning strategy for the five surface forms described earlier.

In order to satisfy equations 4.1 and 4.2, a set of six equations in θ, ϕ, u, and v must be solved simultaneously. These equations are difficult if not impossible to solve for all but the simplest cases. However considered by itself, equation 4.1 is the equation of intersection of two parametric surfaces. Almost every Computer-Aided Design package has a requirement for intersection calculations. As a result this problem has been extensively studied. A review of the state of the art intersection algorithms can be found in Patrikalakis [43]. Thus existing intersection methods can be used to find a set of solutions for these equations, some of which may be contact points. The above concept can be illustrated by examining the intersection of two curves shown in Figure 4.1. In this figure two curves, $\mathbf{c}_1$ and $\mathbf{c}_2$, are gradually moved apart along a common normal. Just before the two curves separate, they are tangent to each other. In other words ***tangency is the boundary case between intersection and no intersection***. The singular nature of tangent points makes them extremely difficult if not impossible to locate.

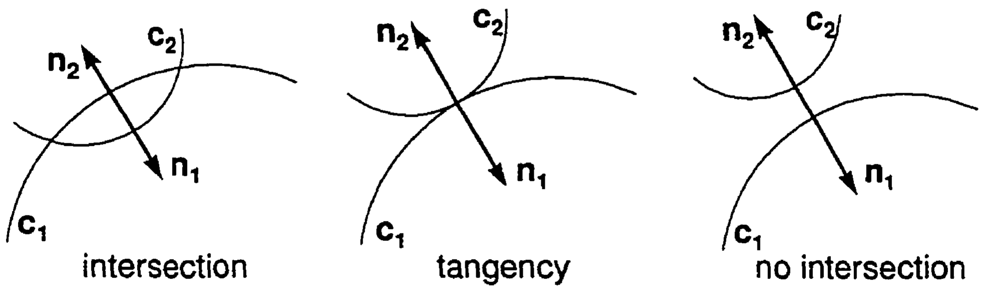


Figure 4.1 Relationship between intersection and no intersection.

This concept can be expressed more formally by considering what is known as the oriented distance function between the parametric surfaces $\mathbf{r}(u, v)$ and $\mathbf{q}(s, t)$ as defined by Kriezis [29].

$$\varphi(u,v) = \mathbf{n}_q(\mathbf{r}(u,v) - \mathbf{Q}(\mathbf{r}(u,v))) \qquad 4.3$$

$\mathbf{Q}(\mathbf{r}(u, v))$ is the orthogonal projection of the point $\mathbf{r}(u, v)$ onto the surface $\mathbf{q}(s, t)$, and $\mathbf{n}_q$ is the unit normal vector at $\mathbf{Q}(\mathbf{r}(u,v))$ on the surface $\mathbf{q}(s, t)$. The orthogonal projection is used to uniquely map a point on the surface **q**(s, t) into the surface **r**(u, v). Pegna and Wolter [45] defined this projection as:

$$(\mathbf{r}(u,v)-\mathbf{q}(s,t))\cdot\mathbf{q}_s=0 \qquad 4.4$$

$$(\mathbf{r}(u,v)-\mathbf{q}(s,t))\cdot\mathbf{q}_t=0 \qquad 4.5$$

where $\mathbf{q}_s$ and $\mathbf{q}_t$ are the partial derivatives with respect to s and t. The magnitude of the distance function, $|\varphi(u,v)|$, is the distance of point **r**(u, v) from the point **Q**(**r**(u, v)) on the surface **q**(s, t). Thus, if you are given two surfaces **r**(u, v) and **q**(s, t), the oriented distance function defines the distance between the surface for every (u, v) point on the surface **r**(u, v).

Since intersection points are zero distance apart, the intersection problem can then be solved by finding the zero level contour of the distance function.

$$\varphi(u,v)=0 \qquad 4.6$$

The resulting solution set may consist of arcs, loops and tangential points as shown in Figure 4.2. Markot and Magedson [38] and Krieziz [29] showed that when a curve of intersection forms a closed loop, there is at least one pair of points within the loop that share a common normal. These points are called critical points of the distance function. They are the local maxima or minima of the distance function and occur when the partial derivatives of the distance functions are zero.

$$\frac{\partial\varphi(u,v)}{\partial u}=\frac{\partial\varphi(u,v)}{\partial v}=0 \qquad 4.7$$

Since the characteristic points share a common normal, they are located on two parallel tangent planes. As the area of the loop approaches zero, the location of the characteristic point approaches the location of a tangent point. This means that the center of a small intersection loop can be used as a good approximation to a tangent point. The area of this loop provides an indication of the accuracy of that approximation.

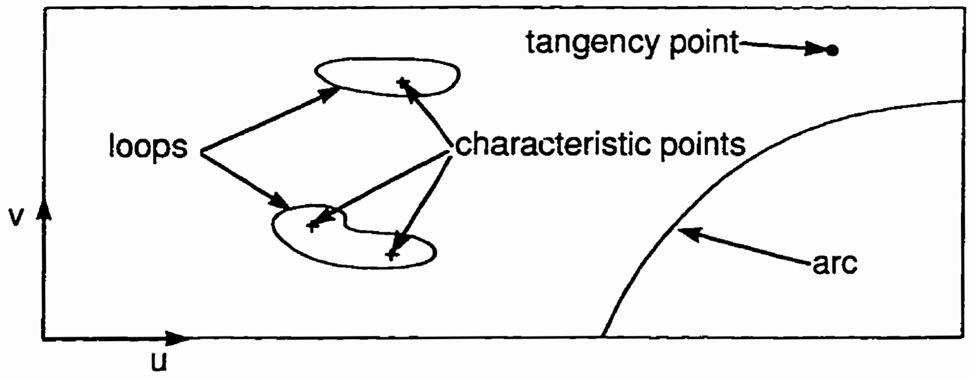


Figure 4.2 Topology of intersection in parametric space.

4.3 Finding Multi-Point Tool Positions Using Intersections

In this section the methodology used to approximate the location of tangential points of contact between a tool and a surface will be described. In the previous section, it was noted that the approximate location of tangency points could be obtained by solving the intersection equation by itself. Since intersections are used extensively in the Computer Aided Design industry, techniques for solving this type of problem are available. Thus the procedure will be to model the contact between a tool and a surface as an intersection problem. The solution to these equations will lead to the location of tangential contact points.

In the previous section equation 4.1 was used to define the intersection problem. It could be solved to find intersection solutions, but it is not in the most convenient form to do so. The tool parameters θ and ϕ and the surface parameters u and v describe the same set of intersection points. The Cartesian coordinates of these points can be determined by substitution into the respective tool or surface equation. However, since we have the luxury of being able to define the tool implicitly, this unnecessary duplication can be avoided. The parametric equation of the surface can be substituted into the implicit definition of the tool. This will give a single set of parameters defining the intersection point.

In order to find an intersection, the surface, S, must be transformed into the tool coordinate system as shown in Figure 4.3. The surface is translated so that the point P(u$_0$,v$_0$) is located at the center of the torus. The surface is rotated so that surface normal is aligned with the z-axis of the tool, and the directions of minimum and maximum curvatures are aligned with the x and y-axes respectively by rotating the surface by γ_z about the z-axis and by γ_x about the x-axis. In this orientation the surface equation becomes:

$$S(u,v)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos(\gamma_x) & \sin(\gamma_x)\\ 0 & -\sin(\gamma_x) & \cos(\gamma_x)\end{bmatrix}\begin{bmatrix}\cos(\gamma_z) & -\sin(\gamma_z) & 0\\ \sin(\gamma_z) & \cos(\gamma_z) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}S_x(u,v)-P_x(u_0,v_0)\\ S_y(u,v)-P_y(u_0,v_0)\\ S_z(u,v)-P_z(u_0,v_0)\end{bmatrix} \qquad 4.8$$

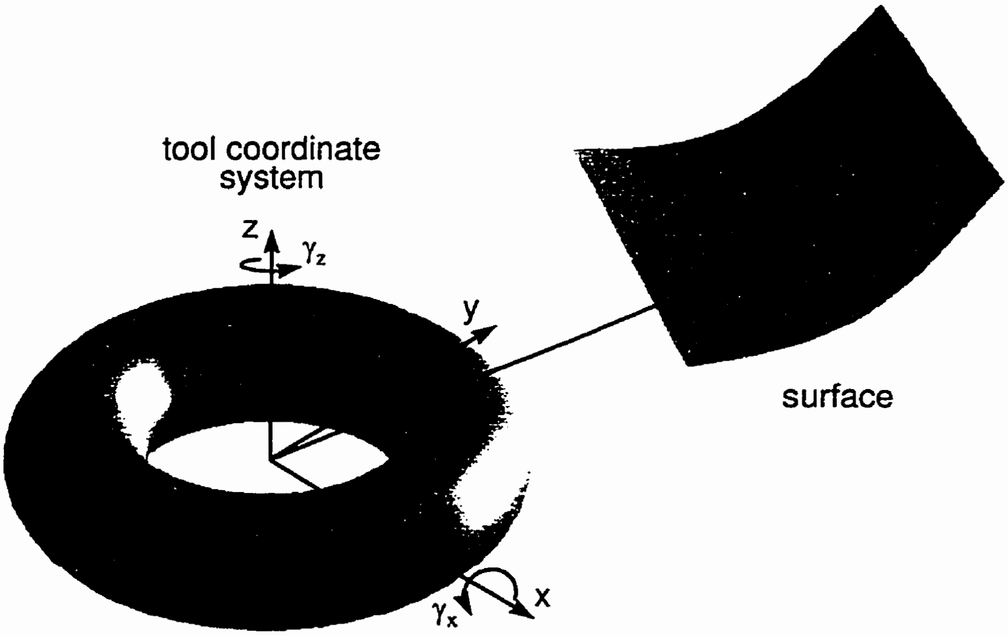


Figure 4.3 Tool coordinate system.

Having expressed the surface in the tool coordinate system, the intersection curves between the tool and the workpiece can be calculated. Different intersections can be obtained by rotating the surface about the y-axis by the angle, β, and about the x-axis by the angle, α, followed by a translation along the tool axis of a distance d as illustrated in Figure 4.4.

$$S(u,v)=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos(\alpha) & \sin(\alpha)\\ 0 & \sin(\alpha) & \cos(\alpha)\end{bmatrix}\begin{bmatrix}\cos(\beta) & 0 & \sin(\beta)\\ 0 & 1 & 0\\ -\sin(\beta) & 0 & \cos(\beta)\end{bmatrix}\begin{bmatrix}S_x(u,v)\\ S_y(u,v)\\ S_z(u,v)\end{bmatrix}-\begin{bmatrix}0\\ 0\\ d\end{bmatrix} \quad 4.9$$

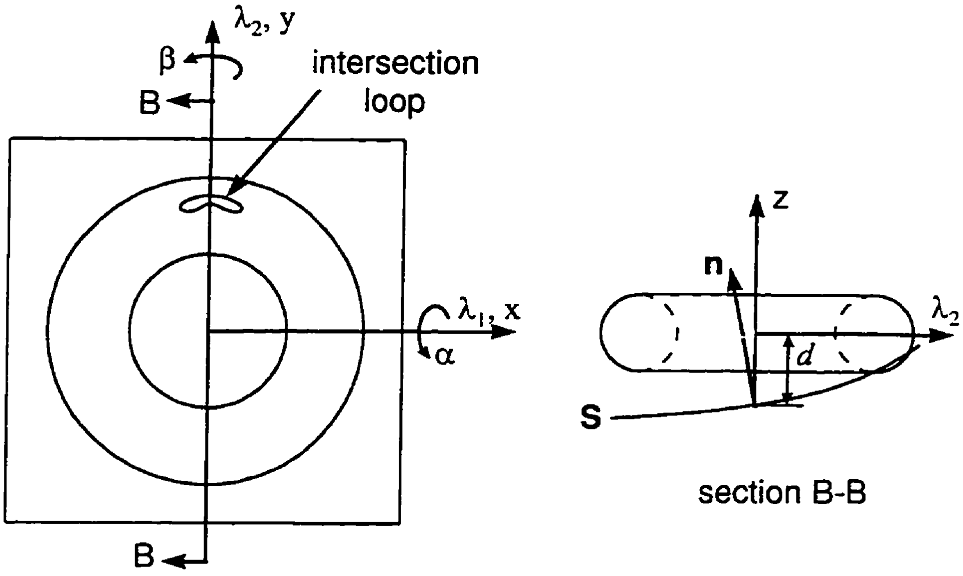


Figure 4.4 Intersection of tool with transformed surface.

The surface equation, S(u, v), is then substituted into the implicit equation of the toroidal cutter which is defined by:

$$(x^2+y^2+z^2+R^2-r^2)^2-4R^2(x^2+y^2)\ =0 \quad 4.10$$

The equation of intersection in the (u, v) plane is:

$$(S_x^2+S_y^2+S_z^2+R^2-r^2)^2-4R^2(S_x^2+S_y^2)=0 \quad 4.11$$

The solution to this equation for a given set of (α, β, *d*) will be an intersection loop, an intersection curve or no solution. Each intersection loop will contain the location of at least one tangential contact point. The center of a small loop of intersection will be the approximate location of a tangential contact point. A set of (α, β, *d*) parameters that produce more then one small loop of intersection will define a multi-point tool position. This procedure was implemented using the symbolic computation package MAPLE. The IMPLICITPLOT function was used to graphically display the resulting intersection. The process of determining (α, β, *d*) parameters that produce multi-point contact was accomplished by trial and error. Loops of intersection were deemed small enough when a change in the offset parameter d of 0.01 mm caused an intersection loop to disappear as shown in Figure 4.5.

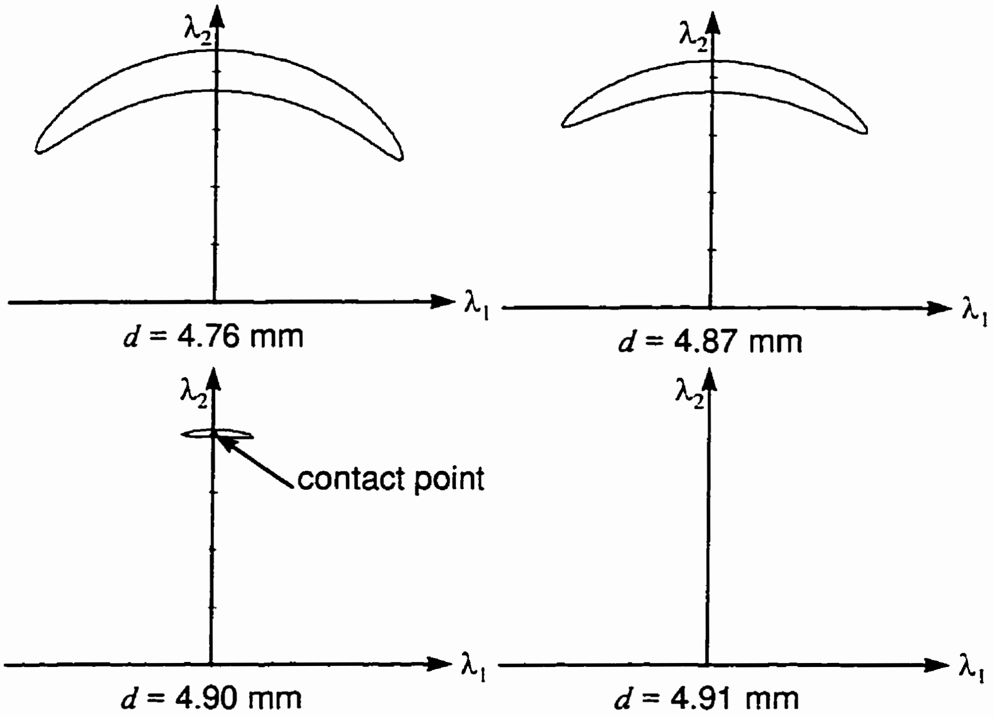


Figure 4.5 Locating a cutter contact point.

4.4 Multi-Point Contact Between a Toroidal Tool and The Five Characteristic Surfaces

Previously, in this chapter it was shown that all surfaces can be characterized into five quadratic forms based on the curvature of a point on the surface. These characteristic surfaces are: planar, parabolic, spherical, elliptic and hyperbolic. The intersection technique will be used to determine the number and arrangement of contact points between these characteristic surfaces and a toroidal end mill. The result of this investigation will make it possible to position a tool on a surface such that the maximum number of contact points can be achieved.

In the following sections the results of the investigation will be presented. Each section will present the results for one of the five quadratic surfaces. Note that the surfaces have been defined so that the parameters (u, v) = (x, y). Intersection loops are labeled with lower case letters. Subscripts correspond to intersection loops generated by a single tool position. For this investigation the tool parameters were R=7.9375 mm and r=4.7625 mm. Therefore, the curvature of the tool's bounding sphere is 1/12.7 mm. The point of interest for each set of intersections was always $\mathbf{P}(u_0, v_0) = \mathbf{P}(0, 0)$.

4.4.1 Multi-Point Contact Between a Toroidal Tool and a Planer Surface

The plane is the simplest of the characteristic forms. It arises when the principle curvatures at a point are both zero. Although tool positioning on a flat surface is well understood, it will be included for the sake of completeness. A plane transformed into the tool coordinate system would pass through the tool center at (0, 0, 0) and the normal to the plane would line up with the z-axis. Such a plane is illustrated in Figure 4.6 and defined by

$$S = S(u,v) = \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} = \begin{bmatrix} u \\ v \\ 0 \end{bmatrix} \tag{4.12}$$

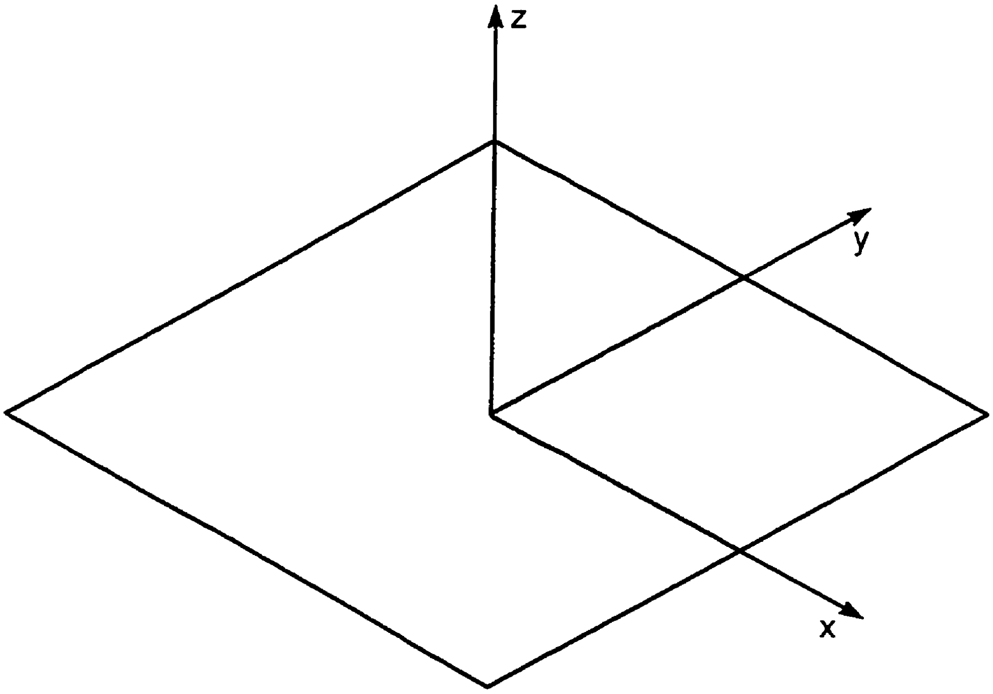
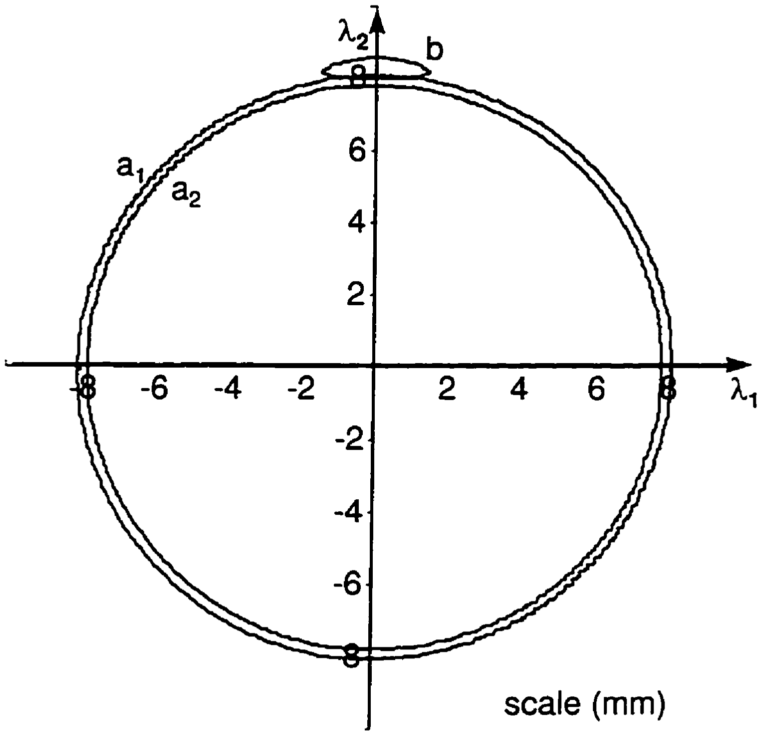


Figure 4.6 Planer surface.

The intersections confirmed that two types of contact are possible between a toroidal cutter and a plane. A ring of contact occurs when the tool axis is lined up with the surface normal and the tool center is offset from the surface by the insert radius, r. This situation corresponds to the intersection circles labeled a_1 and a_2 in Figure 4.7. If the torus was pulled slightly further out of the plane, these circles would converge to a circle of contact between the two intersection circles. If the tool is tilted, only one intersection loop will occur. Intersection b is an example of this situation. The center of this loop is the approximate location of a tangential contact point.



Intersection	a	b
α°	0.0	4.0
β°	0.0	0.0
d (mm)	5.46	5.49

Figure 4.7 Intersections between a toroidal end mill and a plane.

4.4.2 Multi-Point Contact Between a Toroidal Tool and a Parabolic Surface

When one of the principle curvatures is zero at a point, the point is said to be parabolic. All points on a cylinder are parabolic. The axis of a cylinder in the tool coordinate system would line up with the y-axis and pass through the tool center at (0, 0, 0) as shown in Figure 4.8. The circular cylinder defined by equation 4.13 is an example of this type of surface.

$$S = S(u,v) = \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} = \begin{bmatrix} u \\ v \\ -\sqrt{\left(\frac{1}{\kappa_1}\right)^2 - u^2} \end{bmatrix} \qquad 4.13$$

The maximum curvature, κ_1, of this surface was selected to be approximately one fourth that of the tool's bounding sphere; therefore κ_1 = 50 mm.

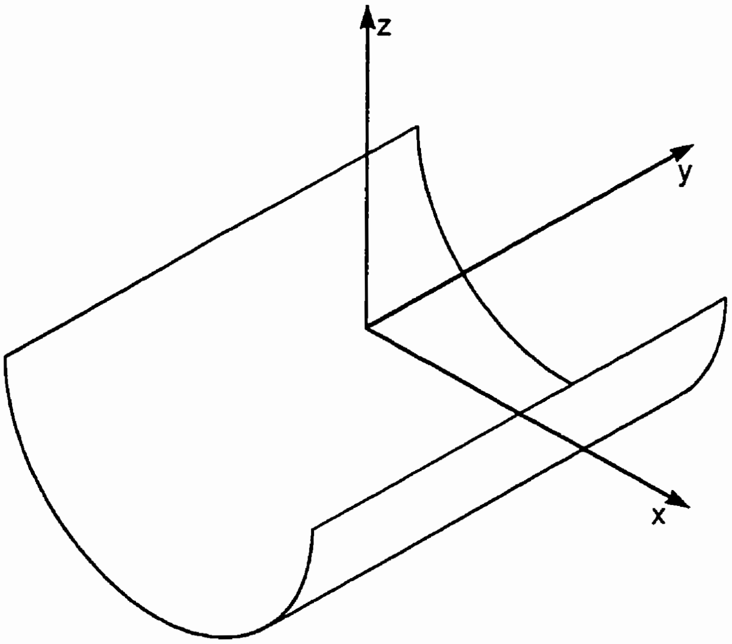
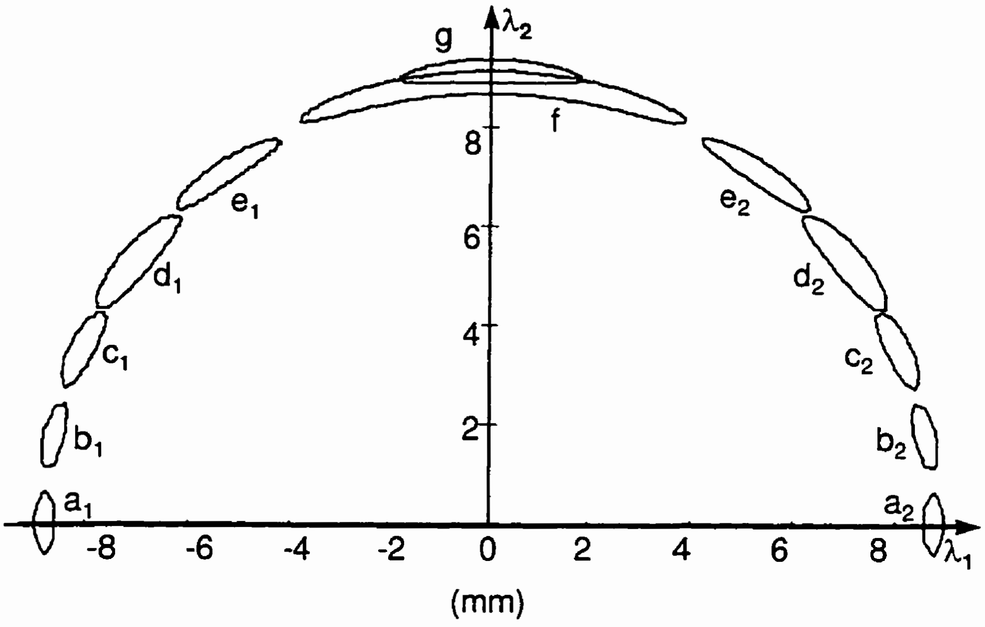


Figure 4.8 A parabolic surface.

The intersections summarized in Figure 4.9 reveal that two types of contact are possible between a toroidal cutter and a parabolic surface. Intersections a through e consist of two loops, the centers of which approximate the location of two tangential contact points. These two point contact solutions occur when the tool is inclined in the direction of minimum curvature, λ_1, and the inclination angle is less than approximately 10 degrees. Inclining the tool beyond 10 degrees in the direction of minimum curvature or inclining the tool in the direction of maximum curvature always produces a single contact point.



Intersection	a	b	c	d	e	f	g
α°	0.0	2.0	4.0	6.0	8.0	10.0	12.0
β°	0.0	0.0	0.0	0.0	0.0	0.0	0.0
d (mm)	5.46	5.49	5.58	5.73	5.95	6.23	6.55

Figure 4.9 Intersections between a toroidal end mill and a parabolic surface.

4.4.3 Multi-Point Contact Between a Toroidal Tool and a Spherical Surface

When both curvatures at a point a equal and non-zero, the point is said to be umbilic. The surface in the vicinity of this point can then be represented by a sphere. In Chapter 3, a geometric method of finding multi-point contact between a toroidal cutter and a spherical surface was developed. The results from Chapter 3 will provide independent confirmation to the intersection method of finding multi-point contact.

In the tool coordinate system, the center of the sphere is at the center of the tool. Such a sphere is illustrated in Figure 4.10 and defined by:

$$S = S(u,v) = \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} = \begin{bmatrix} u \\ v \\ -\sqrt{\left(\frac{1}{\kappa}\right)^2 - u^2 - v^2} \end{bmatrix} \qquad 4.14$$

where the curvature of the sphere is κ = 1/50.0 mm^{-1} is approximately one fourth the curvature of the tool's bounding sphere.

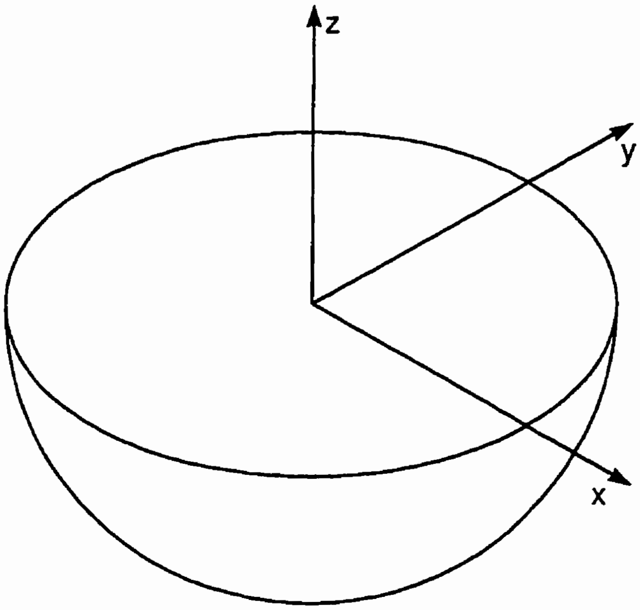
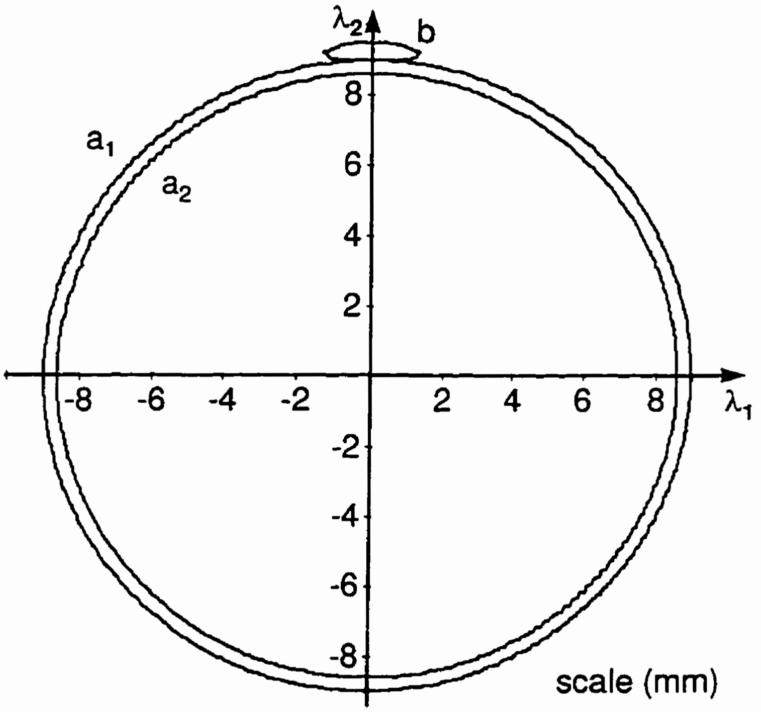


Figure 4.10 A sphere

The results of the intersections between the tool and the sphere are summarized in Figure 4.11. Two types of contact are possible. Intersection circles a_1 and a_2 confirm that circle of tangency between a toroidal cutter and a sphere is possible. This circle lies somewhere between a_1 and a_2. This type of contact occurs when the tool axis is aligned with the surface normal and the tool center is offset from the surface along the surface normal. The intersections indicate that this offset distance, *d*, is approximately 5.46 mm. In Chapter 3, equation 3.3 was developed to calculate the offset distance base on the tool and surface geometry. Using this expression, the offset distance for this tool and surface is 5.46 mm. If the tool is tilted away from the surface normal, only one point of contact will occur. Intersection b is an example of this situation.



Intersection	a	b
α°	0.0	4.0
β°	0.0	0.0
d (mm)	5.46	6.10

Figure 4.11 Intersections between a toroidal end mill and a sphere.

4.4.4 Multi-Point Contact Between a Toroidal Tool and an Elliptic Surface

When the principle curvatures at a point have the same sign but different non-zero values, the point is said to be elliptic. The surface in the vicinity of an elliptic point can be approximated by an ellipsoid. In the tool coordinate system the major and minor axis of the surface line up with the y-axis and x-axis. Thus, an elliptic surface in the tool coordinate system is defined by:

$$\mathbf{S} = \mathbf{S}(u, v) = \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} = \begin{bmatrix} u \\ v \\ -\sqrt{\left(\dfrac{1}{\kappa_1}\right)^2 - u^2 - 4\left(\dfrac{\kappa_2}{\kappa_1}\right)^2 v^2} \end{bmatrix} \qquad 4.15$$

where the maximum and minimum curvatures of the surface at (u, v) = (0, 0) are κ_1 and κ_2. For this investigation, $\kappa_1 = 1/50\ \text{mm}^{-1}$ and $\kappa_2 = 1/200\ \text{mm}^{-1}$ which corresponds to one forth and one sixteenth the curvature of the tool's bounding sphere. This surface is shown in Figure 4.12.

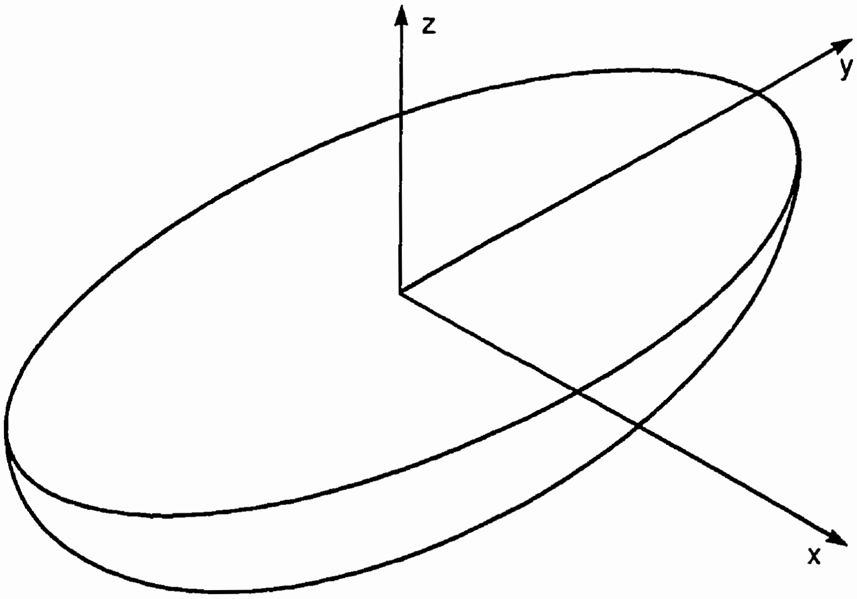
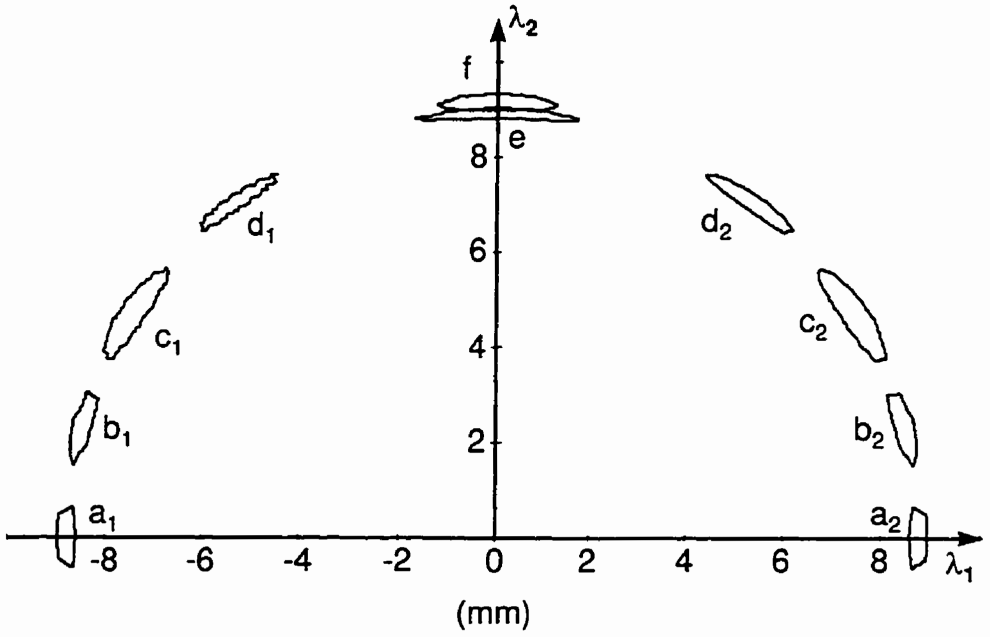


Figure 4.12 An elliptic surface

The intersections summarized in Figure 4.13 can be divided into two groups. Intersections a through d consist of two loops which approximate the locations of two tangential contact points. These two point contact solutions occur when the tool is inclined in the direction of minimum curvature and the inclination angle is less than approximately 8 degrees. Inclining the tool beyond 8 degrees in the direction of minimum curvature or inclining the tool in the direction of maximum curvature always produces a single contact point.



Intersection	a	b	c	d	e	f
α°	0.0	2.0	4.0	6.0	8.0	10.0
β°	0.0	0.0	0.0	0.0	0.0	0.0
d (mm)	5.46	5.50	5.62	5.83	6.12	6.44

Figure 4.13 Intersections between a toroidal end mill and an elliptic surface.

4.4.5 Multi-Point Contact Between a Toroidal Tool and a Hyperbolic Surface

A point is considered hyperbolic if its principle curvatures have opposite signs. All the points on a hyperbolic surface are hyperbolic. The surface in the region of a hyperbolic point can be approximated by a hyperboloid of one sheet. In the tool coordinate system, the origin of the surface is at the tool center and the maximum and minimum directions of curvature, line up with the x and y axes. A hyperbolic surface can be defined by:

$$\mathbf{S} = \mathbf{S}(u,v) = \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} = \begin{bmatrix} u \\ v \\ -\sqrt{\left(\frac{1}{\kappa_2}\right)^2 + v^2 - \left(\frac{\kappa_1}{\kappa_2}\right)u^2} \end{bmatrix} \qquad 4.16$$

where κ_1 and κ_2 are the maximum and minimum curvatures of this surface at (u, v) = (0, 0). For this study these values were $\kappa_1 = 1/25\ \text{mm}^{-1}$ and $\kappa_2 = 1/100\ \text{mm}^{-1}$ The resulting surface is shown in Figure 4.14.

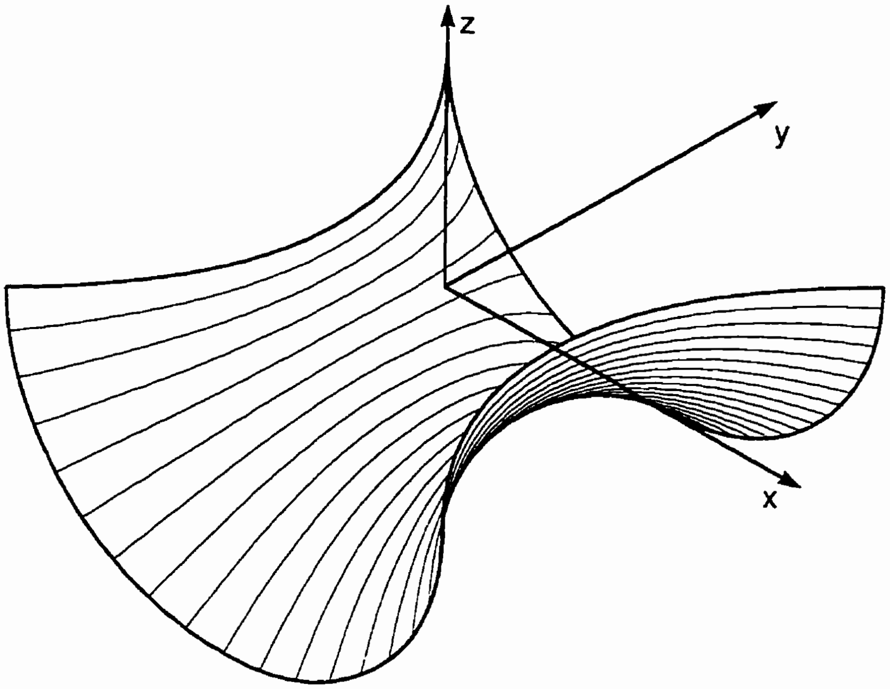
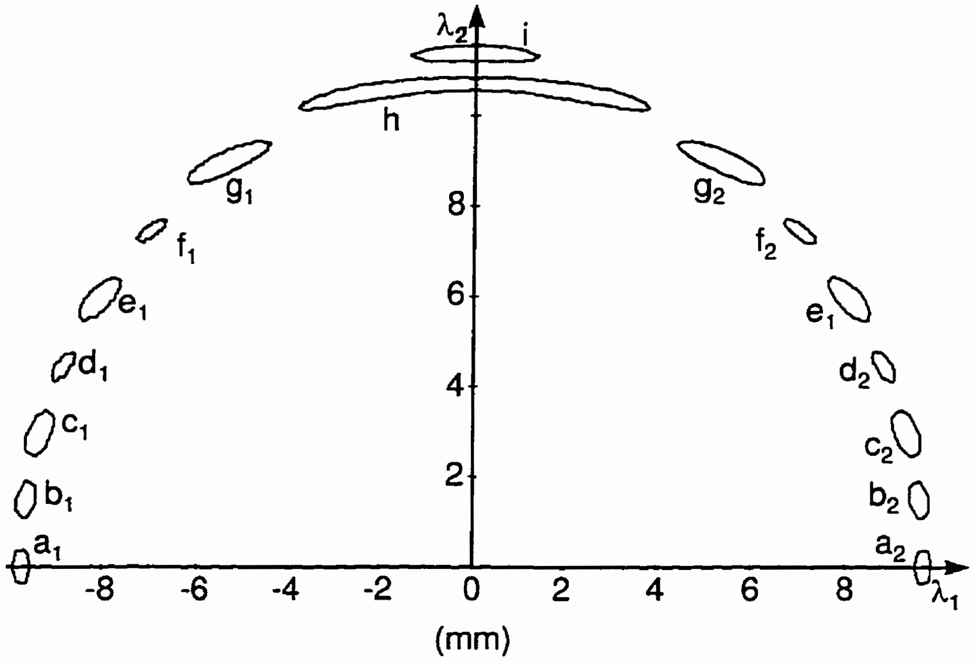


Figure 4.14 A Hyperbolic surface

The intersections summarized in Figure 4.15 can be divided into two groups. Intersections a through h consist of two loops approximating the locations of two tangential contact points. These two point contact solutions occur when the tool is inclined in the direction of minimum curvature and the inclination angle is less than approximately 28 degrees. Inclining the tool beyond 28 degrees in the direction of minimum curvature or inclining the tool in the direction of maximum curvature always produces a single contact point as for example point i.



Intersection	a	b	c	d	e	f	g	h	i
α°	0.0	4.0	6.0	8.0	12.0	16.0	20.0	28.0	32.0
β°	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
d (mm)	6.32	6.37	6.52	6.78	7.14	7.62	8.21	8.94	9.80

Figure 4.15 Intersections between a toroidal end mill and an hyperbolic surface.

4.5 Summary of Multi-Point Contact

Conventional tool positioning strategies always assume a single contact point between the tool and the surface. In this chapter, the contact between a tool and a surface was modeled without making any assumptions about the number or arrangement of contact points. These equations were used to find multi-point tool positions for the five characteristic surfaces: planer, parabolic, spherical, elliptic and hyperbolic. The results of this analysis showed that two distinct patterns of contact exist for these quadratic surfaces depending on their curvatures. The principle curvatures of planer and spherical surfaces are equal. For these surfaces, multi-point contact occurs when the tool axis is aligned with the surface normal and the tool center is offset from the surface. If the tool is tilted in any direction, only a single point of contact occurs. The maximum and minimum curvatures, κ_1 and κ_2, of parabolic, elliptic and hyperbolic surfaces are different from each other. For these surfaces, pairs of contact points are arranged symmetrically about the direction of minimum curvature when the tool is tilted in the direction of minimum curvature. If the tool is tilted beyond a certain angle, or if the tool is tilted in any direction other than the direction of minimum curvature, only a single point of contact will occur.

Chapter 5

Implementation of Multi-Point Machining

In the previous chapter, multi-point tool positions were found by intersecting the tool with the design surface. Although this process is ideal for examining the nature of contact between the tool and the design surface, it can be a slow and labour intensive process. In this chapter, the insight gained from performing the intersection studies will be used to develop a multi-point tool positioning strategy. The multi-point tool positioning strategy was combined with tool path planning and simulation software to form a basic multi-point CAM package.

In Chapter 4, the configurations of multi-point contact between a tool and the various quadratic surfaces were discovered. These surfaces can be used to approximate a surface in the vicinity of a cutter contact point. Therefore, the configuration of contact points for most surfaces should be similar to the results obtained for the quadratic surfaces.

Two types of multi-point contact are possible between a toroidal cutter and a quadratic surface. A maximum of two tangential cutter contact points could be achieved for parabolic, elliptic and hyperbolic surfaces. These points always lie in the direction of maximum curvature. Circles of contact could be achieved on planer and spherical surfaces. Since techniques for multi-point contact with a plane or a sphere have already been discussed, the remainder of this thesis will be devoted to two point contact.

The steps used to produce a multi-point tool path are as follows. The first cutter contact point, cc_1, is specified during the path planning stage as shown in Figure 6.1. This path is simply a set of cc_1 points on the surface called the cutter contact path. Multi-point tool positioning is then performed in two stages for every point on the cutter contact path. First, the second contact point, cc_2, is located a distance w away from cc_1 in the direction of maximum curvature, λ_1. The parameter w is called the separation distance and is specified for every point on the cutter contact path. This procedure specifies the location of the contact points on the surface. The geometry of the torus is then used to place the tool in contact with both of these cutter contact points.

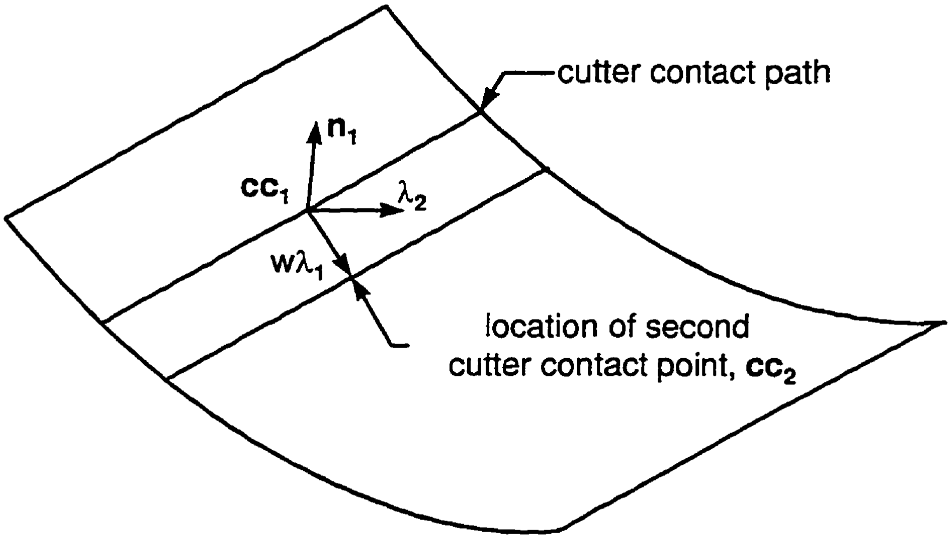


Figure 5.1 Paths of cutter contact points.

Once the tool path has been generated, simulations can be performed. These simulations are a necessity because the results of 5-axis joint motions are impossible to predict intuitively. This requirement is especially important when developing new techniques of tool positioning. For this reason software capable of simulating both metal removal and machine kinematics has been developed for the present work.

5.1 Multi-Point Tool Positioning

Multi-point tool positions will be found based on two assumptions. First, the tool should be able to "fit" inside the curvature of the surface. To satisfy this criteria, the maximum curvature of all points under the tool must be less than the curvature of a sphere that just bounds the tool; with the maximum curvature given by

$$\kappa_1 \leq \frac{1}{R+r}, \qquad 5.1$$

where r is the insert radius and R is the torus radius. The second assumption is that the region of the surface underneath the tool can be reasonably approximated by a quadratic surface. With this assumption the contact between the tool and the surface should be similar to those obtained using intersections in Chapter 4. In the intersection study, two scenarios were found to exist for multi-point contact. If the principle curvatures underneath a tool are equal, then a circle of contact between the tool and the surface is possible. However, if the principle curvatures are different, multi-point contact can be achieved by inclining the tool in the direction of minimum curvature and offsetting the tool from the surface. The resulting pairs of contact points are arranged symmetrically about the direction of minimum curvature.

Multi-point tool positioning consists of two stages. In the first stage, two potential cutter contact points are located on the surface. The first contact point is specified during tool path planning, while the second potential cutter contact point is located assuming the surface is quadratic. In the second stage, the tool is placed in contact with these points.

5.1.1 Determining Cutter Contact Points

In order to find a multi-point tool position, the location of the cutter contact points on the surface must be found. These points will be determined by assuming the surface is quadratic. For example, Figure 5.2 shows a torus in 2-point contact with a parabolic surface. The direction of maximum curvature, λ_1, is across the valley and the direction of minimum curvature, λ_2, is along the valley. The points of contact, **cc**$_1$ and **cc**$_2$, are separated by the separation distance, w, in the direction of maximum curvature.

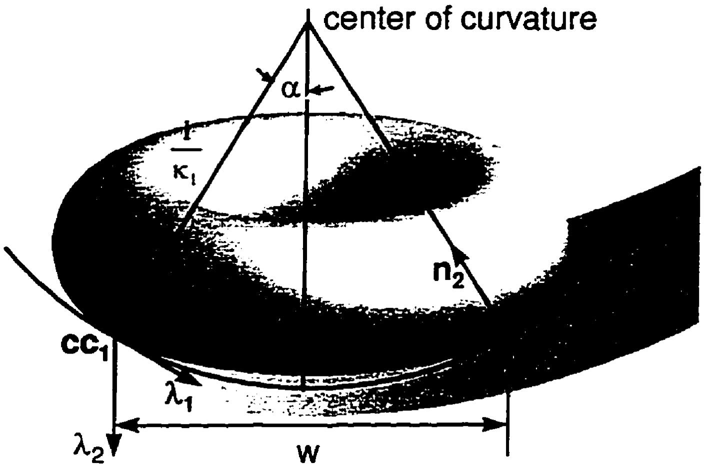


Figure 5.2 Torus placed on a symmetric surface.

The first contact point $\mathbf{cc}_1$ is specified prior to finding the tool position. It may be any point on the surface. The second contact point can be found by noting that $\mathbf{cc}_1$ and $\mathbf{cc}_2$ lie on a circle whose radius is equal to $\frac{1}{\kappa_1}$. This circle lies in a plane containing the surface normal $\mathbf{n}_1$ and the direction of maximum curvature, λ_1. These vectors form a planar coordinate system at $\mathbf{cc}_1$ that is used to calculate the location of $\mathbf{cc}_2$. Note that boldface indicates a vector or a point. The vector $\mathbf{cc}_2$ - $\mathbf{cc}_1$ specifies the location of $\mathbf{cc}_2$ in terms of the planer coordinate system. This vector can be expressed in terms of $\mathbf{n}_1$ and λ_1.

$$\mathbf{cc}_2 - \mathbf{cc}_1 = ((\mathbf{cc}_2 - \mathbf{cc}_1)\cdot\mathbf{n}_1)\mathbf{n}_1 + ((\mathbf{cc}_2 - \mathbf{cc}_1)\cdot\lambda_1)\lambda_1 \qquad 5.2$$

where $(\mathbf{cc}_2 - \mathbf{cc}_1)\cdot\mathbf{n}_1$ and $(\mathbf{cc}_2 - \mathbf{cc}_1)\cdot\lambda_1$ are the components of $\mathbf{cc}_2$ - $\mathbf{cc}_1$ on $\mathbf{n}_1$ and λ_1 respectively. These components may be expressed in terms of w and the angle α.

$$\mathbf{cc}_2 - \mathbf{cc}_1 = w(\sin(\alpha)\mathbf{n}_1 + \cos(\alpha)\lambda_1) \qquad 5.3$$

The position of $\mathbf{cc}_2$ can then be found by rearranging expression 5.3.

$$\mathbf{cc}_2 = w(\sin(\alpha)\mathbf{n}_1 + \cos(\alpha)\lambda_1) + \mathbf{cc}_1 \qquad 5.4$$

The angle α depends on the maximum radius of curvature and the separation between cutter contact points, according to

$$\alpha = \sin^{-1}\left(\frac{\kappa_1 w}{2}\right) \qquad 5.5$$

Note that there will be an error in the location of cc_2 if the surface is not one of the symmetric quadratic forms. In most instances the calculated cc_2 will not lie on the surface, as shown in Figure 5.3. In this case cc_2 is projected onto the surface in the z direction.

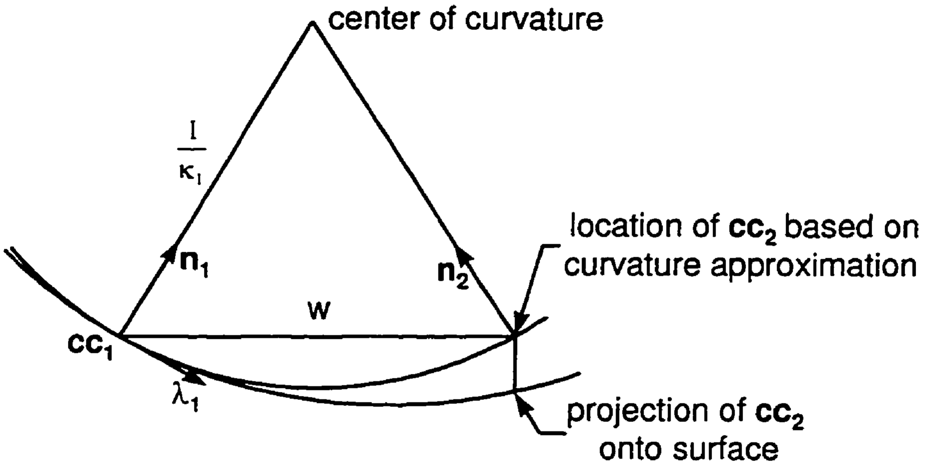
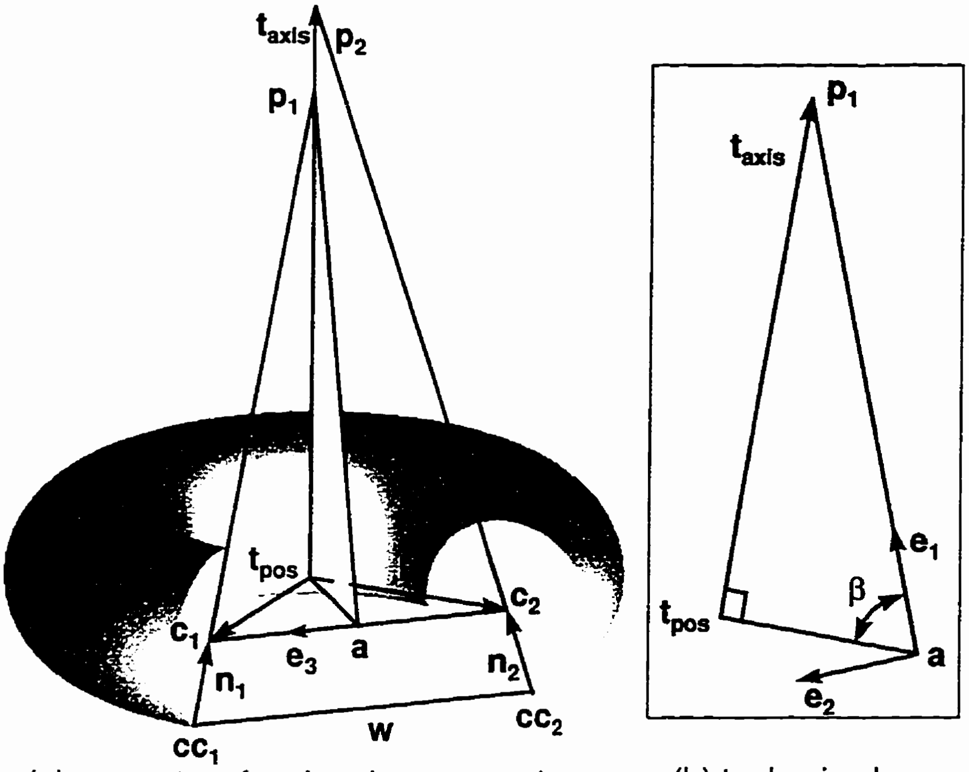

center of curvature
$\frac{1}{\kappa_1}$
n_1
n_2
w
cc_1
λ_1
location of cc_2 based on curvature approximation
projection of cc_2 onto surface

Figure 5.3 Projection of cc_2 onto design surface.

5.1.2 Determining The Multi-Point Tool Position

Once both potential cutter contact points are located, the tool position can be found based entirely on the geometry of the tool and these two cutter contact points. The tool will be positioned such that tangential contact exists between the tool and at least one cutter contact point. Inappropriate selection of potential cutter contact points will result in an error at second cutter contact point because there are only certain combinations of cc_1 and cc_2 that will produce multi-point tool positions.

Figure 5.4(a) shows the tool in tangential contact with cc_1 and cc_2. The lines formed by the normal vectors, $\mathbf{n}_1$ and $\mathbf{n}_2$, at the cutter contact points, cc_1 and cc_2, pass through the insert centers at c_1 and c_2 and intersects the tool axis at $\mathbf{p}_1$ and $\mathbf{p}_2$. Note that $\mathbf{p}_1$ and $\mathbf{p}_2$ would be at the same location if the surface was symmetric. However, for most surfaces this will not be the case. Therefore, the tool position will be calculated without assuming that $\mathbf{p}_1$ and $\mathbf{p}_2$ are at the same location.



(a) geometry of tool and contact points (b) tool axis plane

Figure 5.4 Geometry of multi-point contact.

The position and orientation of the tool can be specified by determining the location of two points on the tool axis. Thus, the points $\mathbf{p}_1$ and $\mathbf{t}_{pos}$ will be found in order to calculate the tool position. The point $\mathbf{t}_{pos}$ will specify the location of the tool and the vector $\mathbf{t}_{axis} = \mathbf{p}_1 - \mathbf{t}_{pos}$ will specify the orientation of the tool.

The point $\mathbf{p}_1$ can be found by intersecting a plane containing the tool axis with the line defined by the points $\mathbf{cc}_1$ and $\mathbf{c}_1$. Therefore, the first step in multi-point tool positioning is to define a plane containing the tool axis. One such plane is the plane perpendicular to the line joining $\mathbf{c}_1$ and $\mathbf{c}_2$ that passes through the midpoint between $\mathbf{c}_1$ and $\mathbf{c}_2$. This plane will be referred to as the tool axis plane and is shown in Figure 5.4(b). The points $\mathbf{c}_1$ and $\mathbf{c}_2$ are located a distance r along the normal vectors $\mathbf{n}_1$ and $\mathbf{n}_2$ from the cutter contact points $\mathbf{cc}_1$ and $\mathbf{cc}_2$:

$$\mathbf{c}_1 = \mathbf{cc}_1 + r\mathbf{n}_1 \qquad 5.6$$
$$\mathbf{c}_2 = \mathbf{cc}_2 + r\mathbf{n}_2$$

Point **a** is the midpoint between $\mathbf{c}_1$ and $\mathbf{c}_2$.

$$\mathbf{a} = \frac{\mathbf{c}_1 + \mathbf{c}_2}{2} \qquad 5.7$$

A vector normal to the tool axis plane, $\mathbf{e}_3$, can be found by noting that the tool axis plane is normal to the vector joining $\mathbf{c}_1$ and $\mathbf{c}_2$.

$$\mathbf{e}_3 = \frac{(\mathbf{c}_1 - \mathbf{c}_2)}{|\mathbf{c}_1 - \mathbf{c}_2|} \qquad 5.8$$

The equation of the tool axis plane is defined by

$$\mathbf{e}_3 \cdot \mathbf{p} - \mathbf{e}_3 \cdot \mathbf{a} = 0 \qquad 5.9$$

where the points, **a** and **p** lie on the plane.

The line joining $\mathbf{cc}_1$ and $\mathbf{c}_1$ is now defined. A point **p** on this line can be defined by $\mathbf{cc}_1$ and $\mathbf{n}_1$ as:

$$\mathbf{p} = \mathbf{cc}_1 + \eta\mathbf{n}_1 \qquad 5.10$$

where η is the distance alone the line from $\mathbf{cc}_1$. The point $\mathbf{p}_1$ can now be found by intersecting the tool axis plane with this line by substituting equation 5.10 into 5.9. The resulting value of η gives the distance between $\mathbf{cc}_1$ and $\mathbf{p}_1$.

$$\eta = \frac{\mathbf{e}_3 \cdot \mathbf{a} - \mathbf{e}_3 \cdot \mathbf{cc}_1}{\mathbf{e}_3 \cdot \mathbf{n}_1} \qquad 5.11$$

Substituting η into equation 5.10 will determine the Cartesian coordinate of the intersection point, $\mathbf{p}_1$.

With $\mathbf{p}_1$ now calculated, the second point, $\mathbf{t}_{pos}$, needs to be determined. This point will be found by considering the geometry of points $\mathbf{t}_{pos}$, $\mathbf{p}_1$ and $\mathbf{a}$ in the tool axis plane as shown in Figure 5.4(b). Note that these three points form a right angle triangle because the plane containing $\mathbf{t}_{pos}$, $\mathbf{c}_1$ and $\mathbf{c}_2$ is always perpendicular to the tool axis. Since this plane is in an arbitrary orientation, bases vectors at point $\mathbf{a}$ must be constructed in order to use planar geometry to locate $\mathbf{t}_{pos}$. A unit vector, $\mathbf{e}_1$, in the direction from $\mathbf{a}$ to $\mathbf{p}_1$ is given by:

$$\mathbf{e}_1 = \frac{(\mathbf{p}_1 - \mathbf{a})}{|\mathbf{p}_1 - \mathbf{a}|} \qquad 5.12$$

A second unit vector, $\mathbf{e}_2$, perpendicular to $\mathbf{e}_1$ and $\mathbf{e}_3$ may be expressed as:

$$\mathbf{e}_2 = \mathbf{e}_1 \times \mathbf{e}_3 \qquad 5.13$$

The distance, d, between the center of the tool, $\mathbf{t}_{pos}$, and point $\mathbf{a}$ is given by:

$$d = |\mathbf{a} - \mathbf{t}_{pos}| = \sqrt{R^2 - \frac{|\mathbf{c}_2 - \mathbf{c}_1|^2}{2}} \qquad 5.14$$

The tool position can now be calculated by:

$$\mathbf{t}_{pos} = \mathbf{a} + d\sin(\beta)\cdot\mathbf{e}_1 + d\cos(\beta)\cdot\mathbf{e}_2 \qquad 5.15$$

where:

$$\beta = \cos^{-1}\left(\frac{d}{|\mathbf{p}_1 - \mathbf{a}|}\right) \qquad 5.16$$

Given two points on the tool axis, the tool axis vector, $\mathbf{t}_{axis}$, is calculated by normalizing the vector from $\mathbf{t}_{pos}$ to $\mathbf{p}_1$.

$$\mathbf{t}_{axis} = \frac{(\mathbf{p}_1 - \mathbf{t}_{pos})}{|\mathbf{p}_1 - \mathbf{t}_{pos}|} \qquad 5.17$$

Together, the tool axis vector, $\mathbf{t}_{axis}$, and the tool position vector, $\mathbf{t}_{pos}$, define the orientation and position of a multi-point tool position. However, the geometry used to calculate the multi-point tool position is a necessary but not sufficient condition for 2-point contact. 2- point contact is not possible for any pair of points on the surface. If the potential cutter contact points are selected correctly, the resulting tool position will produce tangential contact at $\mathbf{cc}_1$ or $\mathbf{cc}_2$. However, if the two potential cutter contact points are selected incorrectly, the resulting tool position will not produce tangential contact at either cutter contact point. For instance, if $\mathbf{cc}_1$ and $\mathbf{cc}_2$ do not lie in the direction of maximum curvature, tangential contact will not be achieved at both points. The final stage of the multi-point positioning strategy involves placing the tool in tangential contact with at least one of the cutter contact points. The resulting tool position will then machine the design surface correctly at one point if not both.

This process is accomplished by moving the position of the tool such that tangential contact is achieved at $\mathbf{cc}_1$. In this process the tool position will be altered but the tool axis will remain the same. Basically a point on the torus, $\mathbf{p}_t$, that could produce tangential contact at $\mathbf{cc}_1$ is located. Then the tool is moved so that $\mathbf{p}_t$ is in tangential contact with $\mathbf{cc}_1$. In order for $\mathbf{cc}_1$ and $\mathbf{p}_t$ to be tangential, their normal vectors must be collinear. In Figure 5.5, a point on the torus, $\mathbf{p}_t$, with a normal vector $\mathbf{n}_t$, collinear with the surface normal $\mathbf{n}_1$, is located relative to the tool center. This point must lie on a plane containing the tool axis, $\mathbf{t}_{\mathbf{axis}}$, and the surface normal, $\mathbf{n}_1$. The normal to this plane is:

$$\mathbf{n} = \mathbf{n}_1 \times \mathbf{t}_{\mathbf{axis}} \qquad 5.18$$

The position of point $\mathbf{p}_t$ in the tool coordinate system is:

$$\mathbf{p}_t = \mathrm{R}(\mathbf{t}_{\mathbf{axis}} \times \mathbf{n}) - \mathrm{r}\mathbf{n}_1 \qquad 5.19$$

In order to achieve tangential contact at $\mathbf{cc}_1$, the tool must be translated by the distance between $\mathbf{cc}_1$ and $\mathbf{p}_t$, which is:

$$\mathbf{cc}_1 - (\mathbf{t}_{\mathbf{pos}} + \mathbf{p}_t) \qquad 5.20$$

Note that $\mathbf{t_{pos}}$ was added to $\mathbf{p_t}$ to convert from the tool coordinate system to the workpiece coordinate system. The tool position is now translated by the distance between $\mathbf{cc_l}$ and $\mathbf{p_t}$.

$$\mathbf{t}_{pos} = \mathbf{t}_{pos} + \mathbf{cc}_l - (\mathbf{t}_{pos} + \mathbf{p}_t) \tag{5.21}$$

which reduces to:

$$\mathbf{t}_{pos} = \mathbf{cc}_l - \mathbf{p}_t \tag{5.22}$$

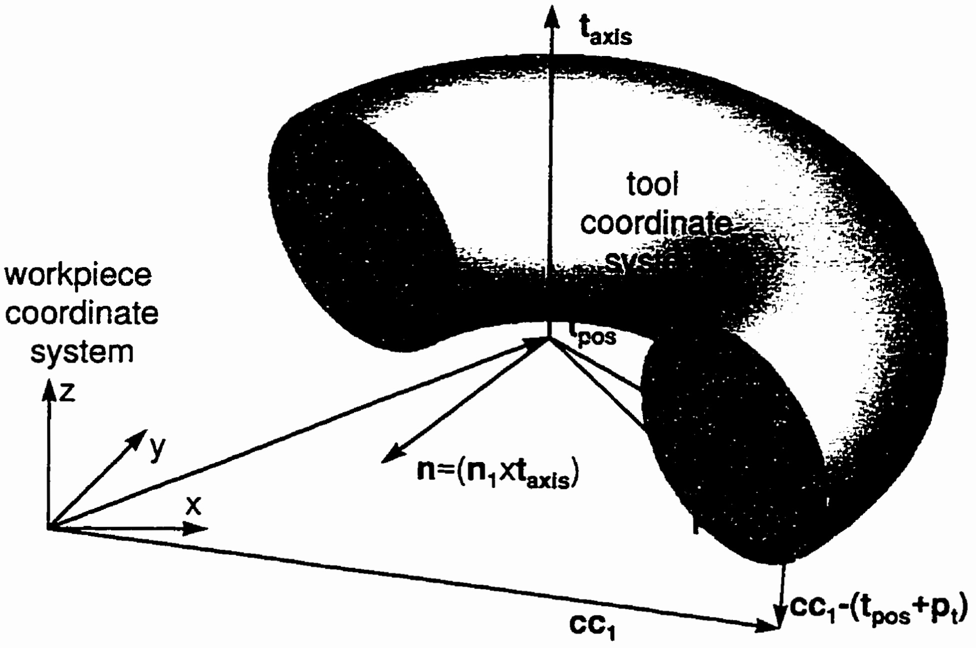


Figure 5.5 Placing tool tangent to $\mathbf{cc_1}$.

5.2 Multi-Point Tool Path Planning

Tool path planning and optimization is not within the scope of this thesis. However, to test the effectiveness of the multi-point tool positioning strategy, a basic tool path generation package was developed.

In Chapter two, the two most important issues in tool path planning were identified as the spacing of points along a tool pass and the distance between adjacent tool passes. This spacing is known as the tool pass interval. In multi-point machining the tool path planner must also consider the spacing and location of the cutter contact points. The approach used for tool path planning is to define the tool path in terms of the first cutter contact point, $\mathbf{cc}_1$. The tool path can then be generated using proven tool path planning techniques. The second cutter contact point is assumed to be offset from the tool path in the direction of maximum curvature as shown in Figure 5.1.

A non-parametic path for $\mathbf{cc}_1$ has been implemented to eliminate the possibility of diverging tool paths. Tool path planning takes place in the xy plane as shown in Figure 5.6. The user must specify: the upper and lower corners of the bounding rectangle; the feed direction; and the location of the starting point of the tool path in the xy plane. Tool passes are then generated in cutter planes that are perpendicular to the xy plane shown in Figure 5.7. The tool pass interval is calculated by considering the tool pass interval plane shown in Figure 5.8.

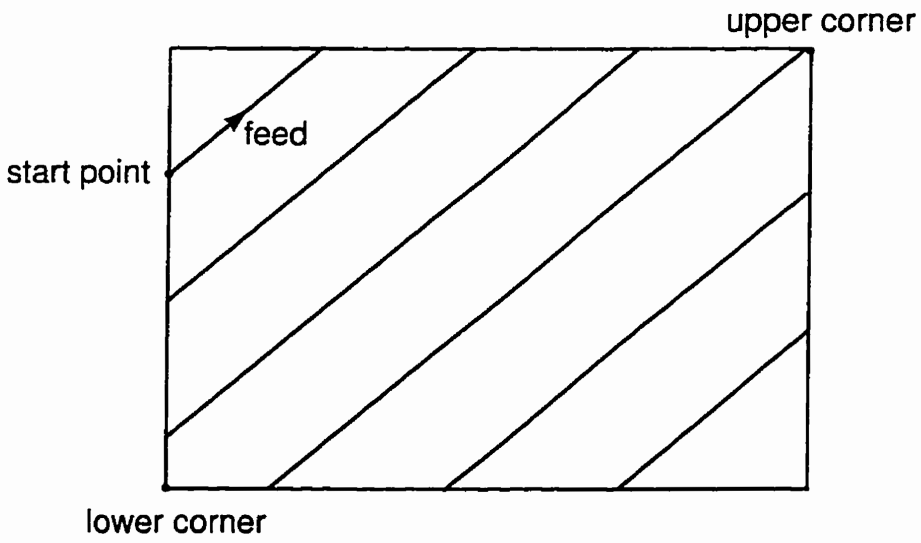


Figure 5.6 Foot print of tool path.

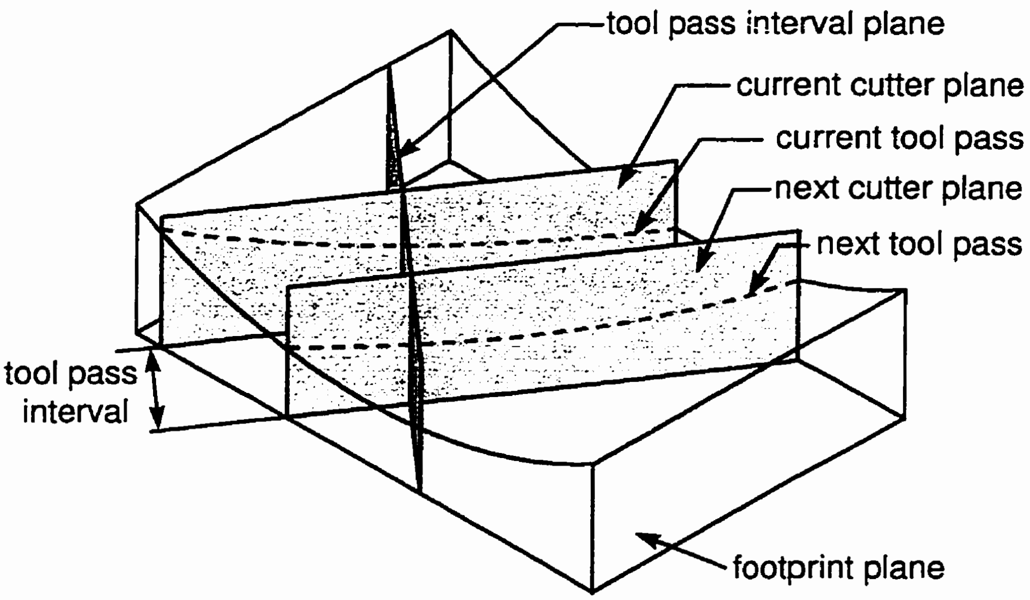


Figure 5.7 Planes used for tool path planning.

5.2.1 The Tool Position Spacing

Points along a tool pass should be spaced such that the tool does not deviate significantly from the surface as the NC controller performs interpolation. At the same time, the overall number of points should be minimized to maximize controller throughput. In 5-axis machining, this problem is difficult to solve because the interpolation takes place in machine joint space instead of Cartesian space. This means that the path taken by the tool is non-linear and depends on the configuration of the machine and the workpiece setup. A simple example can be used to illustrate this problem. Consider two tool positions with tool axis vectors defined by (0.097, 0.026, 0.995) and (0.026, 0.097, 0.995). The angle between these vectors is only 5.7°. However, after being postprocessed for a tilt-rotary table type 5-axis machine, the A and C rotations required to achieve these orientations are (5.7, 255.0) and (5.7, 195.0) respectively-the C axis will have to move 60° between these two orientations! This motion would probably move the tool off the design surface. This problem is entirely due to the machine kinematics and could be fixed by calculating intermediate positions.

Since machining accuracy is more important than controller throughput, a conservative approach to tool position spacing has been adopted. Cutter locations are evenly spaced along a tool pass such that the cordal distance δ is constant, as shown in Figure 5.8. As long as δ is small enough, the tool will not gouge the surface. Experience has shown that a δ value of 0.1 mm will ensure that gouging will not occur. While this value may seem excessively small, the example has shown that even small changes in a tool orientation can result in large joint motion.

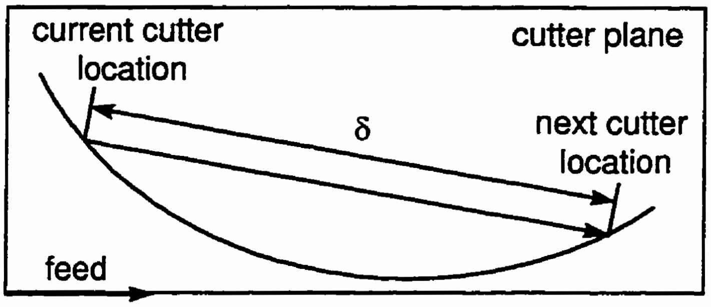


Figure 5.8 Stepping along tool path.

5.2.2 The Tool Pass Interval

For this investigation, the tool pass interval will be calculated in the tool pass interval plane as shown in Figure 5.9. The tool pass interval is calculated such that the cordal distance between the cutter planes, χ, is constant. Note that using this methodology, the tool pass interval would vary along the length of the tool pass. In order to maintain a constant distance between cutter planes, the tool pass interval is calculated for the endpoints and midpoint of each tool pass. The average value is then used for the interval between cutter planes.

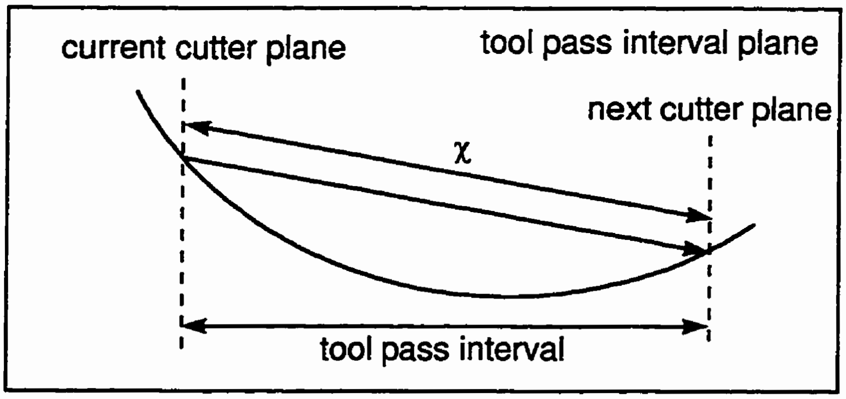


Figure 5.9 The tool pass interval.

5.3 Simulation

Two types of machining simulations can be performed. Metal removal simulations are used to estimate scallop geometry prior to post-processing. After post-processing, the kinematics of the NC machine are simulated to confirm that the tool path generated is correct and it will not produce undesired interference between the tool and the machine.

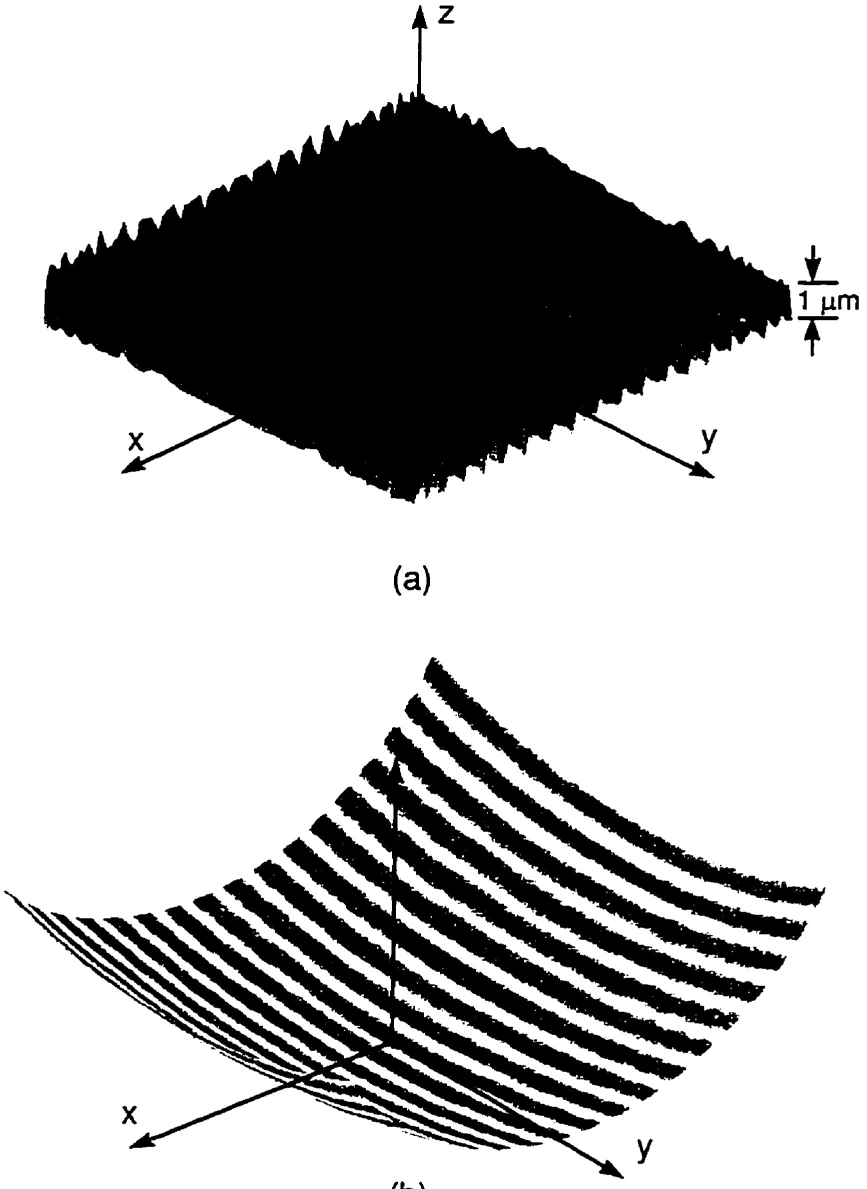
5.3.1 Simulation of Metal Removal

The "mow the grass" technique described in section 2.6.1 and illustrated in Figure 2.23 is used to simulate metal removal. This technique has been found to be fast and reliable for the simulation of small workpieces.

In the "mow the grass" technique, vectors are grown from the design surface in the direction normal to the surface. The length of these vectors are stored in an array. Each vector is intersected with the tool at each tool position. The length of the intersected vector is compared to its original length. If the length of the vector has been reduced, the new length is stored in the array.

When the simulation is complete, every vector has been intersected by every tool position. The resulting set of vectors represents the deviations between the design surface and the machined surface. Positive deviations indicate scallops and negative deviations indicate gouges. The maximum deviation is used as a measure of the maximum scallop height and the minimum deviation is used as a measure of the maximum gouge. The positive and negative deviations can also be integrated to determine the scallop and gouge volumes.

The results of a simulation for the elliptic surface described in section 4.3.4 are shown in Figure 5.10. This surface was machined using the multi-point technique with a tool pass interval and cutter contact separation of 5.0 mm. In part (a) of the figure, the surface deviations have been tessellated and displayed as a surface in Cartesian coordinates. The xy plane corresponds to xy positions on the surface and the deviations are measured along the z-axis. The surface deviations can also be mapped onto the surface as shown in part (b). In this case, darker shades indicate larger scallop heights. A summary of the results is given in part (c). For comparison the simulation was also performed using the same tool path with the same sized ball nosed tool. Note that neither techniques gouged the surface and that the scallops produced by multi-point machining were far smaller than those produced by the ball nosed end mill.



(a)

(b)

	MPM R = 7.9375, r = 4.7625 (mm)	**ball** r = 12.7 mm
scallop height (mm)	1.3	351.2
gouge (mm)	0	0
scallop volume (mm3)	3.7	726.5
gouge volume (mm3)	0	0

(c)

Figure 5.10 Example of simulation results for elliptic surface. a) surface deviations, b) color map, c)summary of numerical results.

5.3.2 Machine Simulation

Once the cutter location data has been created, it can be converted to G-code using a post-processor and sent to the NC controller to machine a workpiece. However, prior to executing the G-code program on a real machine it is a good idea to simulate it to ensure that the tool path will not produce any unanticipated interference between the tool and the machine or the fixtures. After all, machining through a simulated clamp is much cheaper than machining through a real one! For this reason, the Virtual Rambaudi was developed to simulate Waterloo's new 5-axis NC, machine as shown in Figure 5.11. More details of the Virtual Rambaudi can be found in Appendix B.

The Virtual Ramboudi is a realistic 3-D model of the target 5-axis milling machine. The machine model was constructed in CorelCAD, which is a solid modeling based Computer Aided Design package. This solid model was then converted into polygons. The polygon model could then be animated using the kinematics discussed in Chapter 2. Rendering of the 3-D model was performed using the OpenGL API. This methodology makes it possible to quickly develop or modify the simulator for different machine types and to create models of customized tooling and fixtures.

The user can interact with the simulator through the controller dialog box shown in Figure 5.12. This Graphical user interface was constructed using the Microsoft Foundation Class library in conjunction with Visual C++. A high level of interaction was maintained by multi-threading the controller dialog box separately from the simulation. This means that the user can modify the simulation through the controller while a simulation is in progress instead of waiting for the completion of the simulation.

The controller dialog box provides most of the functionality of a real NC controller. For example, the user can: load in a G-code program and visually confirm that the desired motions will be performed correctly by the machine tool; select different tools and workpiece configurations; and manually position any of the machine's axes. The user can also interact with the simulator by modifying the view of the machine. The user can pan, zoom or orbit the machine in order to observe a particular feature of interest.

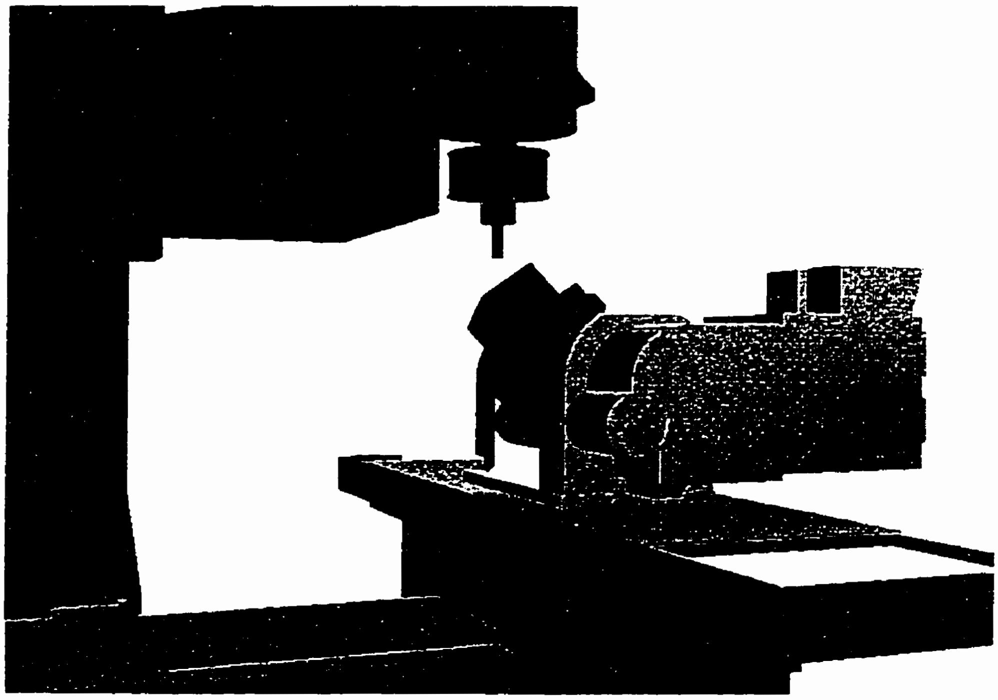
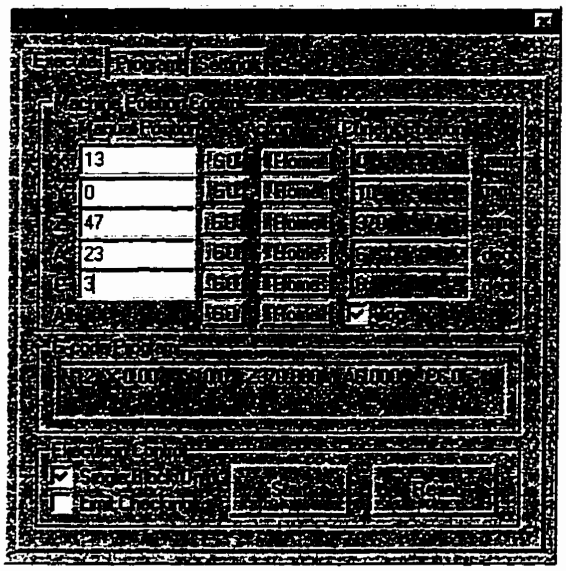


Figure 5.11 The Virtual Ramboudi



13

0

47

23

3

Figure 5.12 The Virtual Ramboudi controller interface.

Chapter 6

Algorithms for Tool Position Adjustment

Like all tool positioning strategies, multi-point machining may fail when the assumptions about the surface under the tool are violated. The result may be excessive gouging or scallops. The methods described in this chapter will be used to correct tool positions that do not achieve multi-point contact. These methods are not just gouge detection and correction algorithms; they can also detect and modify sub-optimal tool positions that are not gouging the surface. This is a major departure from the traditional gouge detection and correction approach.

The method of calculating multi-point tool positions discussed in Chapter 5 is based on the assumptions that the principle directions of curvature are known and that they are constant under the tool. This information is needed to calculate the location of the cutter contact points. However, curvature information may not be known precisely. This problem is quite common when the surface is defined outside of the Computer Aided Manufacturing software. For example, a surface may be defined by a set of points generated by measuring a prototype object with a Coordinate Measurement Machine (CMM). In such cases normal and curvature information must be approximated numerically. Errors in these approximations will result in errors in the tool position. Even when curvature information is available, the principle directions of curvature may change radically in the region under the tool and the resulting quadratic approximation of the surface may be unreasonable. This may occur when using high order surfaces or at the juncture of two surface patches. The result will be an error in the location of the second cutter contact point and a corresponding error in the tool position.

Given that curvature information may not exist, be poorly approximated or change drastically underneath the tool, two algorithms have been developed in this work to adjust a tool position such that multi-point contact is achieved. Both algorithms use the tool positioning technique described in Chapter 5 as an initial solution. In this solution the direction of maximum curvature is used to determine the position of the second cutter contact point. If the direction of maximum curvature is unavailable, it is assumed to be perpendicular to both the feed direction and surface normal.

The adjustment algorithms assume that only 2-point contact is possible between the surface and the tool. The intersection studies documented in Chapter 4 showed that 2-point contact was the most likely scenario for multi-point contact. On surfaces that allow circles of contact, tool positioning is relatively straightforward and adjustment techniques are not required. One can still imagine rare instances when three or more point contact can occur. These rare cases will be left for future studies.

Both adjustment algorithms search for 2-point contact incrementally. The first technique approaches the problem from the perspective of the tool, whereas the second approaches the problem from the perspective of the surface. These approaches will be designated the *tool approach* and the *surface approach*.

6.1 The Tool Approach

Figure 6.1 illustrates the tool approach. In this approach the tool is rotated incrementally. After each rotation the tool is placed in tangential contact with the surface at **cc**$_1$. A second potential cutter contact point, **cc**$_2$, is then identified. The normal distance between this point and the tool is used as a measure of the error at **cc**$_2$. This error is minimized in order to identify the true location of **cc**$_2$.

This approach depends on two sub-algorithms. The heart of the approach is a sub-algorithm used to locate potential cutter contact points on the surface. A second sub-algorithm adjusts the tool position based on the error at **cc**$_2$.

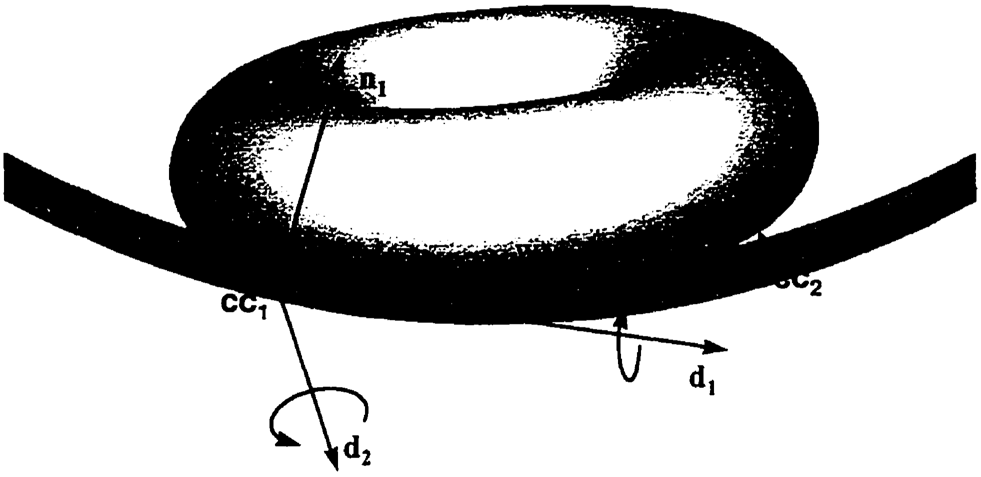


Figure 6.1 Minimizing error at second cutter contact point by rotating the tool.

6.1.1 Locating a Potential Cutter Contact Point

Recall from Chapter 4 that a cutter contact point must satisfy equations 4.1 and 4.2. These equations state that for tangential contact, a point on the design surface and a point on the tool must share the same location in space and the normal vectors at these points must be collinear. When the tangential contact problem was reformulated as a special case of the intersection problem, tangential contact could be explained using oriented distance functions. An oriented distance function defines the distance between two surfaces. The local minima of the oriented distance function occur at the characteristic points of the two surfaces. These points have collinear normal vectors, but do not necessarily share the same location in space. When the value of the oriented distance function is zero at a characteristic point, it is also a tangential contact point. In a sense, characteristic points are potential cutter contact points. Therefore, cutter contact points can be located by finding a pair of characteristic points whose separation is zero.

The search for characteristic points is illustrated in figures 6.2 and 6.3. First, a point on the tool and a point on the surface are located such that their normal vectors are parallel. A point $\mathbf{p}_s$ on the surface **S** is selected and its normal $\mathbf{n}_s$ is calculated. A vector in the tool coordinate system, $\mathbf{p}_t$, which describes the location of the a point on the tool with a normal, $\mathbf{n}_t$, in the same direction as $\mathbf{n}_s$ is determined. The vector $\mathbf{p}_t$ must lie in a plane containing $\mathbf{t}_{axis}$ and $\mathbf{n}_s$ as shown in Figure 6.3. The normal to this plane is given by:

$$\mathbf{n} = \mathbf{n}_s \times \mathbf{t}_{axis} \tag{6.1}$$

The position of point, $\mathbf{p}_t$, is given by

$$\mathbf{p}_t = R(\mathbf{t}_{axis} \times \mathbf{n}) - r\mathbf{n}_s + \mathbf{t}_{pos} \tag{6.2}$$

Note that the addition of the tool position, $\mathbf{t}_{pos}$, transforms $\mathbf{p}_t$ into the surface coordinate system. Once the location of both normal vectors are identified, the perpendicular distance, d, between these normal vectors can be calculated:

$$d = \left| \frac{(\mathbf{p}_t - \mathbf{p}_s) \cdot (\mathbf{n}_t \times \mathbf{n}_s)}{\mathbf{n}_t \times \mathbf{n}_s} \right| \tag{6.3}$$

For a pair of characteristic points, the value of d should be zero since characteristic points have collinear normal vectors. Powell's method [46] was used to search the uv plane for a point that minimizes d. When d is less than a given tolerance, the normal vectors are considered to be collinear and a pair of characteristic points has been located. In this work this tolerance was set at 0.0001 mm. The distance between the pair of characteristic points along the surface normal represents for the error in locating the second cutter contact point.

$$\text{error} = |\mathbf{p}_t - \mathbf{p}_s| \qquad 6.4$$

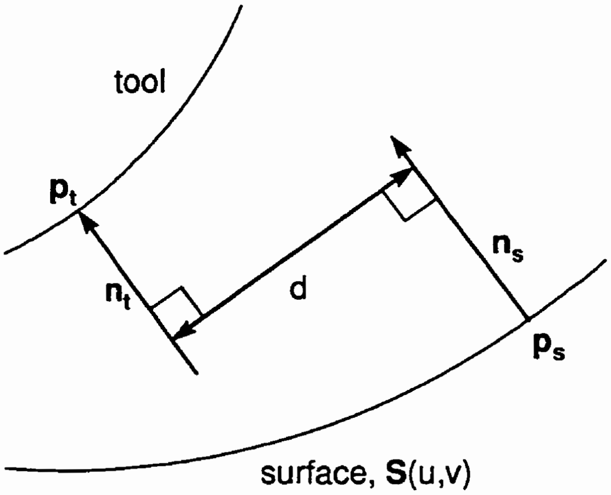


Figure 6.2 Search for a pair of characteristic points.

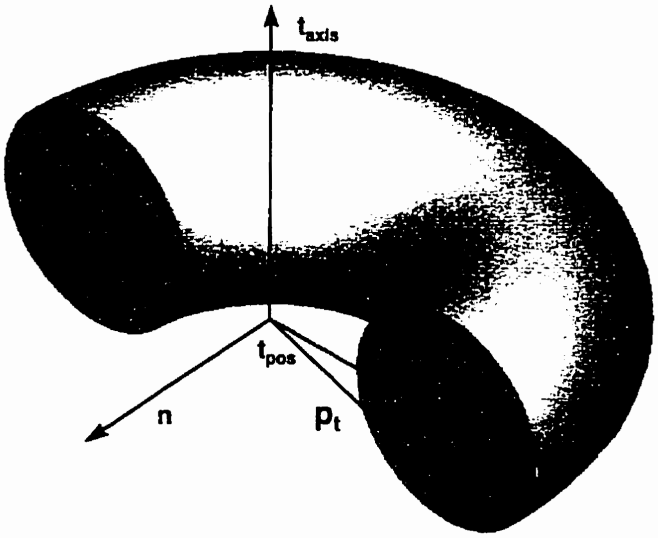


Figure 6.3 Locating a normal on the tool in the tool coordinate system.

The search for a second potential cutter contact point is complicated by two factors. First of all, there is no guarantee that a second characteristic point exists. For example, if the tool is positioned on a plane such that it is in tangential contact with the surface at one point, a second characteristic point will not exist. The second complication is produced by the first cutter contact point. The algorithm has a tendency to converge on this point because it is also a characteristic point. These complications are overcome by starting the algorithm at eight different locations on the surface as shown in Figure 6.4. These starting points are arranged in a circle about the expected location of the second contact point. This location is based on the previous tool position. The best results were obtained when the radius of this circle was a tenth of the desired separation distance, w. If the algorithm converges to **cc₁** after each restart, a second characteristic point would be assumed not to exist.

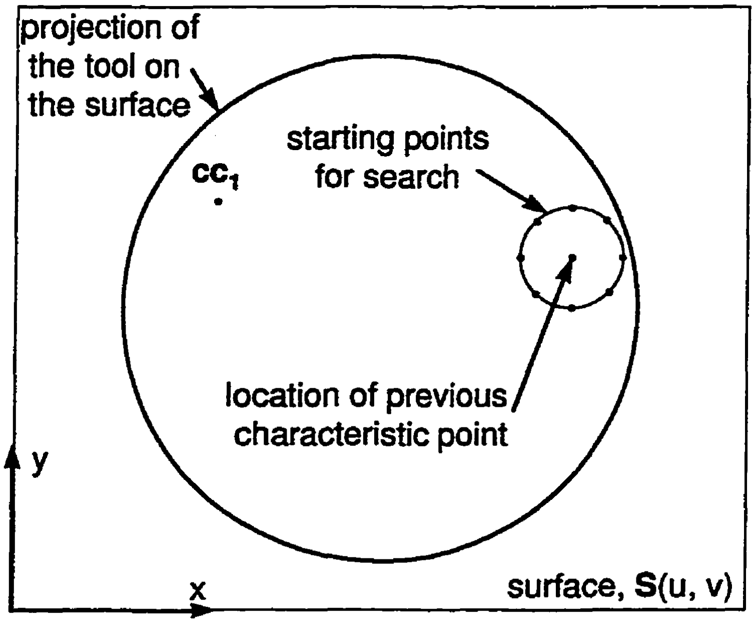

projection of the tool on the surface
starting points for search
cc_1
location of previous characteristic point
y
x
surface, **S**(u, v)

Figure 6.4 Starting points of search for characteristic points.

6.1.2 Tool Position Adjustment

Tool position adjustment is accomplished by rotating the tool while maintaining tangential contact at the first cutter contact point until a second cutter contact point is located. This process is complicated by the fact that the separation distance, w, between cutter contact points must be kept within a specified tolerance. Thus tool position adjustment can be posed as a constrained non-linear optimization problem. The error at the second cutter contact point is the objective function to be minimized and the rotations are the variables of interest. In classical optimization, a number of sophisticated techniques such as the conjugate gradient methods [46] or quasi-Newton methods [46] can be used to solve this type of problem. Typically the objective function is considered to be a surface in the space defined by the variable of interest. Most optimization techniques locate a minimum by traveling along the surface in a series of directions. These techniques are distinguished by the methods used to decide in which direction to travel and by how far to travel in a given direction. Unfortunately, the most powerful of these techniques require the evaluation of the partial derivatives with respect to the variable of interest to determine the best direction to travel. Since the error at the second cutter contact point must be calculated numerically, these techniques can not be implemented efficiently. Therefore, in the present work, the direction used for minimization will be determined based simply on the geometry of the problem.

The directions used for minimization are the axes of rotation. In general, a minimum of two independent axes of rotation are required to orient the tool. From the intersection study it was determined that the locations of the cutter contact points are dependent on the directions of principle curvature underneath the tool. Therefore, these are the logical choice for the axes of rotation. However, the proposed method should be able to deal with surfaces in which the directions of curvature are poorly defined or those that change significantly under the tool. Furthermore, these directions should be easy to determine. Therefore the directions of rotation should contain curvature information implicitly without actually performing curvature calculations.

The axes of rotation used for tool position optimization are based on the positions of **cc₁** and **cc₂** shown in Figure 6.5. The first direction, **d₁**, approximates the direction of maximum curvature and the second direction, **d₂**, approximates the direction of minimum curvature. The direction **d₂** lies in the tangent plane at **cc₁** and is perpendicular to the vector (**cc₂**-**cc₁**). Thus, **d₂** can be calculated from

$$\mathbf{d}_2 = \left|(\mathbf{cc}_2 - \mathbf{cc}_1) \times \mathbf{n}_1\right| , \qquad 6.5$$

where **n₁** is the normal vector at **cc₁**. The direction **d₁** also lies in the tangent plane, but is perpendicular to **d₂**. Therefore:

$$\mathbf{d}_1 = \mathbf{n}_1 \times \mathbf{d}_2 \qquad 6.6$$

The directions of rotation **d₁** and **d₂** depend on the current tool position and the location of **cc₂**. They are used to incrementally rotate the tool until the error at **cc₂** has been minimized. Mathematically the rotations are accomplished by multiplying the tool axis, $\mathbf{t}_{\mathbf{axis}}$, by the rotation matrix

$$\begin{bmatrix} u_x^2 + (1-u_x^2)\cos\alpha & u_x u_y(1-\cos\alpha) - u_z \sin\alpha & u_x u_z(1-\cos\alpha) + u_y \sin\alpha \\ u_x u_y(1-\cos\alpha) + u_z \sin\alpha & u_y^2 + (1-u_y^2)\cos\alpha & u_y u_z(1-\cos\alpha) - u_x \sin\alpha \\ u_x u_z(1-\cos\alpha) - u_y \sin\alpha & u_y u_z(1-\cos\alpha) + u_x \sin\alpha & u_z^2 + (1-u_z^2)\cos\alpha \end{bmatrix} \qquad 6.7$$

where the components of the either rotation direction **d₁** or **d₂** are substituted for $\begin{bmatrix} u_x & u_y & u_z \end{bmatrix}^T$ and the corresponding rotation angle, α_{d1} or α_{d2}, is substituted for α. The derivation of this matrix can be found in Faux and Pratt [14].

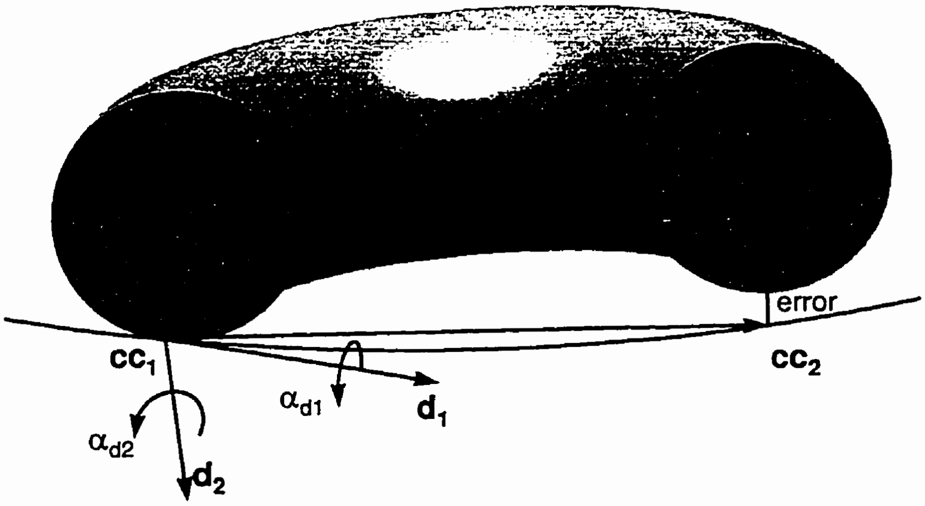


Figure 6.5 Axes of rotation for tool position optimization.

Each rotation affects cc_1 as well as cc_2. In most cases a rotation will destroy the tangential contact at cc_1. After every rotation, tangential contact is restored at cc_1 by shifting the tool position slightly in the manner illustrated by Figure 5.5.

A search technique was required to efficiently determine the angle of rotation that would minimize the error at cc_2. The golden section search [46] was ultimately selected for this purpose for the following reasons. Search techniques that relied on derivative information about the error function could not be used because expressions for the derivative does not exists. Higher order methods that employed quadratic approximation were found to be less reliable when constraints were employed because of the discontinuity of the error function at the constraints. Even though the golden section search is not the fastest method available, its reliability made it the most suitable.

Constraints on the location of the second cutter contact point were required to ensure that the specified separation distance between the cutter contact points was maintained. These constraints were implemented by modifying the error function. The desired separation,

w_d, between the cutter contact points was specified at each tool position. The actual separation, w_a, was calculated as follows

$$w_a = |\mathbf{cc}_2 - \mathbf{cc}_1| \tag{6.8}$$

If the difference between the actual and desired separation was within a predefined tolerance, the error function was expressed by equation 6.4. If the difference between the actual and desired separation was out of tolerance, the error was modified by adding a barrier function, f, that artificially increase the value of the error when the solution is outside the constraints.

$$\text{error} = \text{error} + f(w_a - w_d) \tag{6.9}$$

Uniform, linear, quadratic, and exponential barrier functions were used to modify the error function. The following quadratic form was found to work the best.

$$f(w_a - w_d) = 1000 * (w_a - w_d)^2 \tag{6.10}$$

The implementations of the constraints on the error functions resulted in some difficulties for the minimization technique. If a characteristic point is located outside the constraints, the algorithm would tend to gravitate to the point on the constraint closest to the characteristic point as illustrated in Figure 6.7. Sometimes, both rotation directions, $\mathbf{d}_1$ and $\mathbf{d}_2$, would get "stuck" at a constraint. This problem was overcome by selecting a third search direction. In many cases rotating the tool about the normal vector $\mathbf{n}_1$ at $\mathbf{cc}_1$ would be sufficient to continue the search because it was perpendicular to both $\mathbf{d}_1$ and $\mathbf{d}_2$.

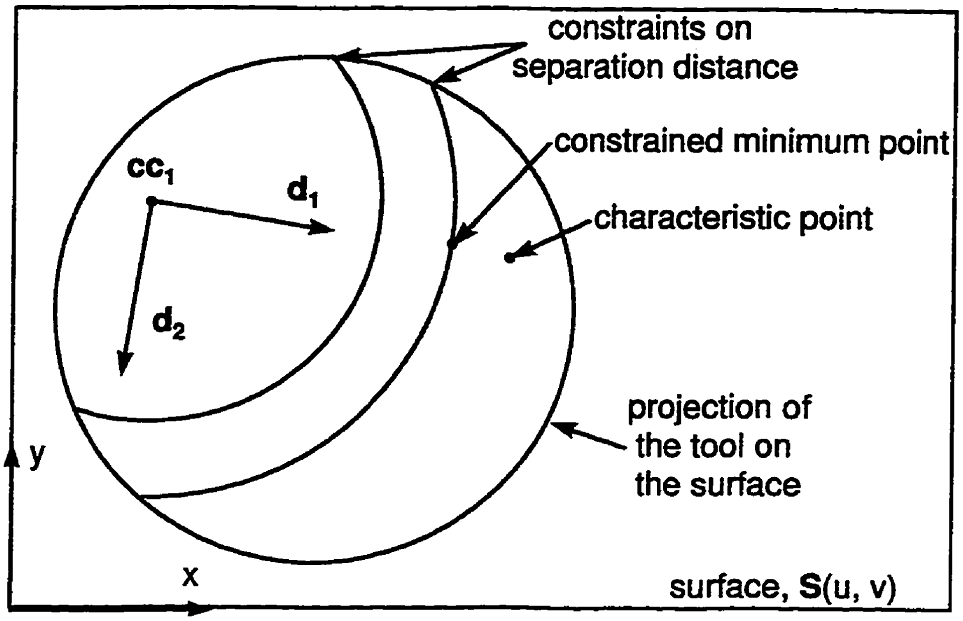

constraints on separation distance
constrained minimum point
characteristic point
projection of the tool on the surface
surface, S(u, v)
cc_1
d_1
d_2
y
x

Figure 6.7 Constraints on the location of second cutter contact point.

6.1.3 Summary of The Tool Algorithm

The flowchart in Figure 6.8 describes the major steps involved in the tool approach algorithm. The algorithm requires an initial multi-point tool position and the location of the first cutter contact point, $\mathbf{cc}_1$, and the location of a potential second cutter contact point, $\mathbf{cc}_2$. In this work the initial solution was calculated based on the multi-point tool positioning strategy discussed in Chapter 5. Next the error in the tool position is calculated by determining the distance between the characteristic points on the tool and the surface. If the error is within tolerance the tool position adjustment is completed. If the error is unacceptable, the tool position is modified in the following manner. First the directions of rotation, $\mathbf{d}_1$ and $\mathbf{d}_2$, are calculated based on the positions of $\mathbf{cc}_1$ and $\mathbf{cc}_2$ and the surface normal $\mathbf{n}_1$ at $\mathbf{cc}_1$. The tool is then rotated about $\mathbf{d}_1$ and $\mathbf{d}_2$ until the error at $\mathbf{cc}_2$ is minimized. After a set of rotations, the algorithm checks the solution to see if it has become stuck at a constraint. This check is performed by determining if the error has decreased. A decrease in the error implies that the rotations have been able to improve the solution and tool adjustment may continue until the error is within tolerance. If the error did not decrease, the solution must be stuck on a constraint. Therefore a new direction of rotation is needed. Since the surface normal at cc_1 is perpendicular to the previous directions it is a logical choice. If the new direction improves the solution, the algorithm will continue until the error is within tolerance. Otherwise the adjustment procedure is terminated. Even when the algorithm fails the resulting tool position will still have been improved.

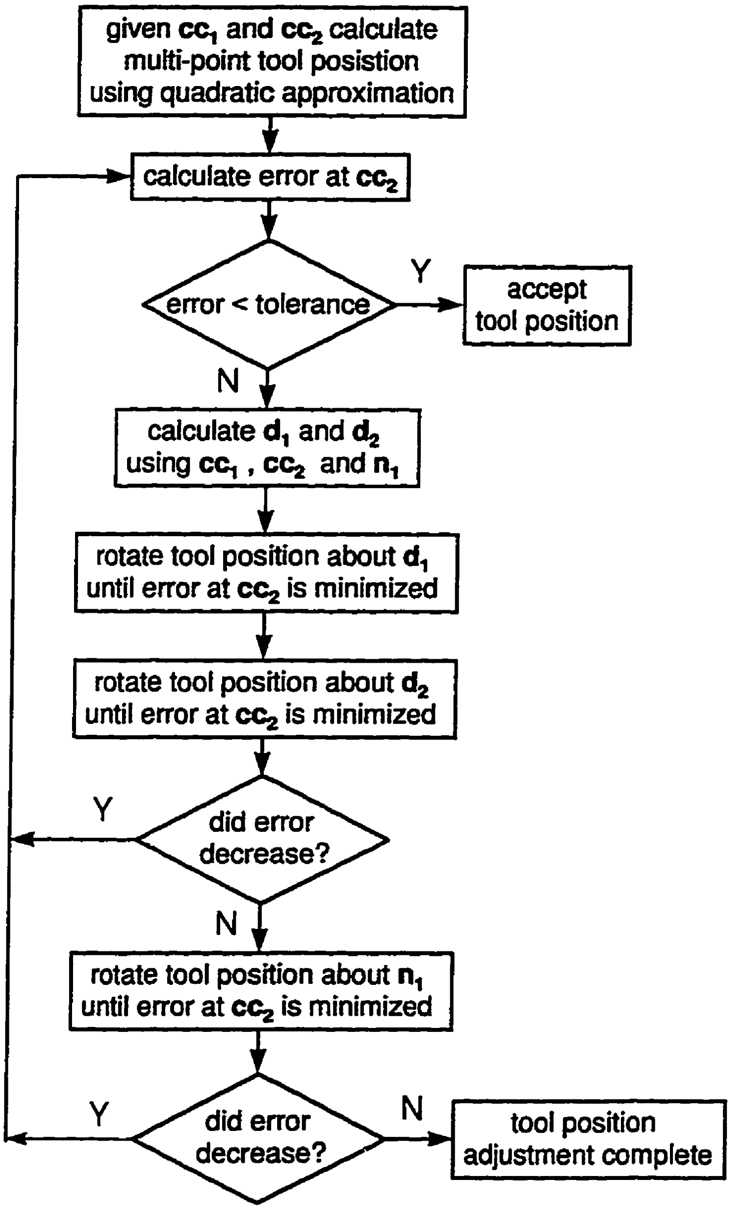

given $\mathbf{cc_1}$ and $\mathbf{cc_2}$ calculate multi-point tool posistion using quadratic approximation
calculate error at $\mathbf{cc_2}$
error < tolerance
Y
accept tool position
N
calculate $\mathbf{d_1}$ and $\mathbf{d_2}$ using $\mathbf{cc_1}$, $\mathbf{cc_2}$ and $\mathbf{n_1}$
rotate tool position about $\mathbf{d_1}$ until error at $\mathbf{cc_2}$ is minimized
rotate tool position about $\mathbf{d_2}$ until error at $\mathbf{cc_2}$ is minimized
Y
did error decrease?
N
rotate tool position about $\mathbf{n_1}$ until error at $\mathbf{cc_2}$ is minimized
Y
did error decrease?
N
tool position adjustment complete

Figure 6.8 The tool approach algorithm.

6.1.4 Example of The Tool Algorithm

The test surface, described by equation 6.11, is an open concave surface similar to those commonly found in the mold industry.

$$S(u,v)=\begin{bmatrix} S_x \\ S_x \\ S_x \end{bmatrix}=\begin{bmatrix} 80u-20u^2 \\ 120v-20v^2 \\ 15+25v^2-30u-30uv^2+50u^2+30u^2v^2-30u^2v \end{bmatrix} \quad u,v\in[0,1] \qquad 6.11$$

The tool path used to machine the workpiece was generated for a cutter contact separation and tool pass interval of 10.0 mm. The cutter contact separation was allowed to vary by ±10%, and the maximum error on the second contact point was ±0.0001 mm. The tolerance on the separation distance was found to have a profound effect on the speed and the success of this technique. A small tolerance would require considerable computational time and reduce the success of locating a second contact point. A large tolerance resulted in a large variance in the separation between the contact points **cc_1** and **cc_2** and in unwanted fluctuations in scallop height. A tolerance of 10% on the location of **cc_2** was found to be a good compromise. With this tolerance the tool correction algorithm required approximately 30 sec on a SPARC 2 workstation for one tool position and successfully found 2-point solutions approximately 99% of the time.

The resulting tool path was used to machine the test surface on a FADAL VMC 4020 5-axis tilt-rotary table milling machine at McMaster University, Hamilton, Ontario, Canada. The cutting conditions used were: spindle speed = 4000 rpm, feed = 0.05 mm/tooth, 1.0 mm depth of cut, using a 25.4 mm diameter toroidal cutter with two 4.7625 mm inserts. Figure 6.9 shows the machined surface. Faint jagged lines are formed along the cutter contact points. The maximum surface deviation occurs at the midpoint between the two cutter contact points. The jaggedness of the line is due to the ±10% allowable wander in the second contact point. The small variations in the location of **cc_2** produced small variations in the line where successive tool passes overlap. Measured results were obtained on a coordinate measuring machine (CMM). The average scallop height was 27 μm, which is almost 100 times smaller than the 2420 μm scallops that would have been produced had the same tool path been used with a ball nosed end mill of the same size.

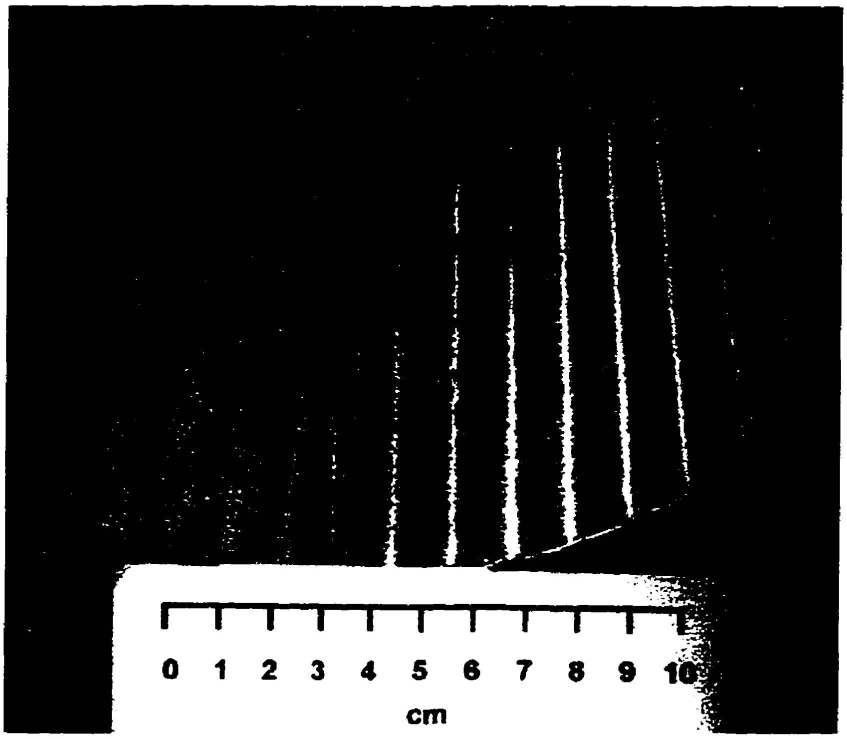


Figure 6.9 Multi-Point Machining with tool position modification.

6.2 The Surface Approach

In this approach, a search is conducted for the correct location of the second cutter contact point, cc_2. This search can also be posed as a non-linear optimization problem whose main components are: the error function to be minimized, the search strategy, and the implementation of constraints.

6.2.1 The Error at a Potential Cutter Contact Point

In the multi-point machining strategy, the tool is placed on two potential cutter contact points, $\mathbf{cc}_1$ and $\mathbf{cc}_2$, such that the tool is in tangential contact with $\mathbf{cc}_1$. If the tool position is incorrect there will be an error at $\mathbf{cc}_2$, which can be measured using the inside-outside test. This test is a proven method of determining if a point lies on a surface or not and is frequently used in solid modeling and computer graphics [15, 37]. This test is illustrated with the sphere of radius, r_s, shown in Figure 6.10. This sphere can be defined by:

$$f(x,y,z) = x^2 + y^2 + z^2 - r_s^2 \qquad 6.12$$

The figure also shows three points, $\mathbf{p}_{in}$, $\mathbf{p}_{on}$ and $\mathbf{p}_{out}$, which represent points inside, on and outside the sphere. If the components of these points are substituted into f(x, y, z), the function value will be less than zero, equal to zero or greater than zero. Thus the sign of f(x, y, z) can be used to determine if a point lies inside, on or outside the surface. In addition, the magnitude of f(x, y, z) can be used as a measure of how far the point is from the surface.

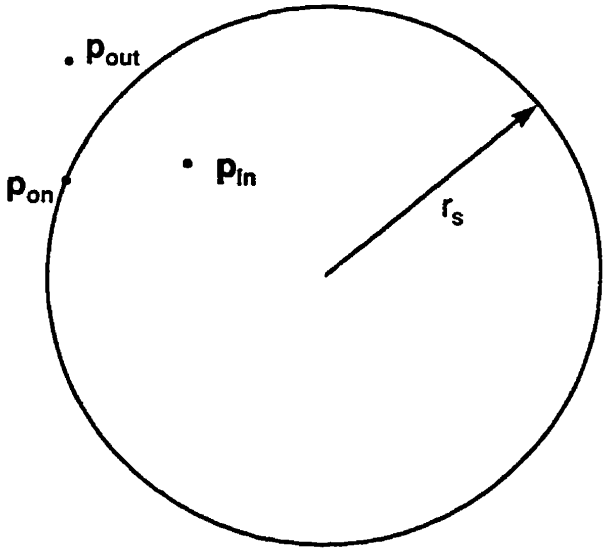


Figure 6.10 The inside-outside test.

The inside-outside test can be extended to any surface that can be defined implicitly. The implicit definition of a torus is:

$$f(x,y,z) = (x^2 + y^2 + z^2 + R^2 - r^2)^2 - 4R^2(x^2 + y^2) \qquad 6.13$$

If:

f(x, y, z) < 0 the point is inside the torus.

f(x, y, z) = 0 the point is on the torus.

f(x, y, z) > 0 the point is outside the torus.

The magnitude of f(x, y, z) is a measure of how close the point is to the surface. Note that even if a point is on the torus, the surfaces may not be in tangential contact. It may be an intersection point. Therefore, the point must also satisfy the tangency requirement given by:

$$\mathbf{n}_s \times \mathbf{n}_t = 0 \qquad 6.14$$

where $\mathbf{n}_s$ and $\mathbf{n}_t$ are the normal vectors of the design surface and the tool at $\mathbf{cc}_2$.

One disadvantage of the inside-outside test is that the implicit definition of the torus is centered about the origin. Therefore, the tested point must be transformed into the tool coordinate system as shown in Figure 6.11 before the inside-outside test can be performed. This means that the point must be translated such that the tool position, $\mathbf{t}_{pos}$, is at the origin and the tool axis, $\mathbf{t}_{axis}$, is on the z-axis. First, $\mathbf{cc}_2$ is translated back along the tool position vector, $\mathbf{t}_{pos}$.

$$\mathbf{cc}_2 = \mathbf{cc}_2 - \mathbf{t}_{pos} \qquad 6.15$$

Then $\mathbf{cc}_2$ is rotated about the origin such that the tool axis vector, $\mathbf{t}_{axis}$, aligned with the z-axis. This rotation can be accomplished using the transformation matrix given by 6.7 and reproduced below.

$$\begin{bmatrix} u_x^2+(1-u_x^2)\cos\alpha & u_x u_y(1-\cos\alpha)-u_z\sin\alpha & u_x u_z(1-\cos\alpha)+u_y\sin\alpha \\ u_x u_y(1-\cos\alpha)+u_z\sin\alpha & u_y^2+(1-u_y^2)\cos\alpha & u_y u_z(1-\cos\alpha)-u_x\sin\alpha \\ u_x u_z(1-\cos\alpha)-u_y\sin\alpha & u_y u_z(1-\cos\alpha)+u_x\sin\alpha & u_z^2+(1-u_z^2)\cos\alpha \end{bmatrix} \qquad 6.16$$

This transformation matrix can be used to rotate **cc₂** by α about an axis defined by $\mathbf{u}=\begin{bmatrix}u_x & u_y & u_z\end{bmatrix}^T$. The axis of rotation, **u**, must be perpendicular to both $\mathbf{t_{axis}}$ and the z-axis as shown in Figure 6.11.Therefore, **u** is defined by

$$\mathbf{u}=\bar{\mathbf{k}}\times\mathbf{t_{axis}}=\begin{vmatrix} i & j & k \\ 0 & 0 & 1 \\ t_{axis_x} & t_{axis_y} & t_{axis_z}\end{vmatrix}=\begin{bmatrix}-t_{axis_y}\\ t_{axis_x}\\ 0\end{bmatrix} \qquad 6.17$$

The angle, α, can be obtained from the z-component of $\mathbf{t_{axis}}$.

$$\alpha=-\cos^{-1}(t_{axis_z}) \qquad 6.18$$

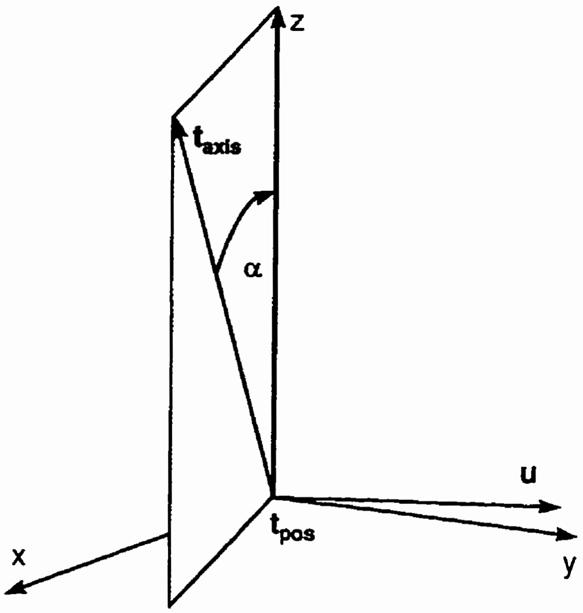


Figure 6.11 Rotation used to transform **cc₂** into tool coordinates.

The composite transformation is:

$$\begin{bmatrix}cc_{2_x}\\ cc_{2_y}\\ cc_{2_z}\end{bmatrix}=\begin{bmatrix} t_{axis_y}^2+(1-t_{axis_y}^2)\cos\alpha & t_{axis\,x}t_{axis_y}(\cos\alpha-1)-t_{axis_z}\sin\alpha & t_{axis_x}\sin\alpha \\ t_{axis\,x}t_{axis_y}(\cos\alpha-1)+t_{axis_z}\sin\alpha & t_{axis_x}^2+(1-t_{axis_x}^2)\cos\alpha & t_{axis_y}\sin\alpha \\ -t_{axis_x}\sin\alpha & -t_{axis_y}\sin\alpha & \cos\alpha \end{bmatrix}\begin{bmatrix}cc_{2_x}-t_{pos_x}\\ cc_{2_y}-t_{pos_y}\\ cc_{2_z}-t_{pos_z}\end{bmatrix}$$

6.16

In summary, the error at $\mathbf{cc}_2$ is calculated by first transforming the point into the tool coordinate system using equation 6.16. The error at $\mathbf{cc}_2$ is then determined using the inside-outside test for a torus given by equation 6.13.

6.2.2 Locating the Second Cutter Contact Point

Now that the error at a potential cutter contact point has been quantified, the method used to search the surface for the correct $\mathbf{cc}_2$ point will be discussed.

One way to locate $\mathbf{cc}_2$ would be to perform a search in parametric space using an optimization technique such as Powell's method [46] to minimize the absolute value of the error found using the inside-outside test. This would require a set of line minimizations in uv space and would be complicated by constraints on the location of $\mathbf{cc}_2$. An alternate method is to combine the constraints with the search direction. In this approach, the constraint becomes the search direction. Since the constraint is in Cartesian space the search must also take place in Cartesian space. Instead of performing a number of 2D line searches in a constrained region of uv space, a single line search in Cartesian coordinates can be performed along a single constraint. Thus the order of the problem is reduced and the constraint on the location of $\mathbf{cc}_2$ is relaxed.

The results of Chapter 4 demonstrated that the second cutter contact point lies in approximately the direction of maximum curvature λ_1. If the tool is maintained in tangential contact with the first cutter contact point $\mathbf{cc}_1$, a set of valid second cutter contact points can be generated as illustrated in Figure 6.12. In other words, for every $\mathbf{cc}_1$ there will be a set of $\mathbf{cc}_2$ points that will satisfy the inside-outside test when the multi-point tool positioning strategy is used.

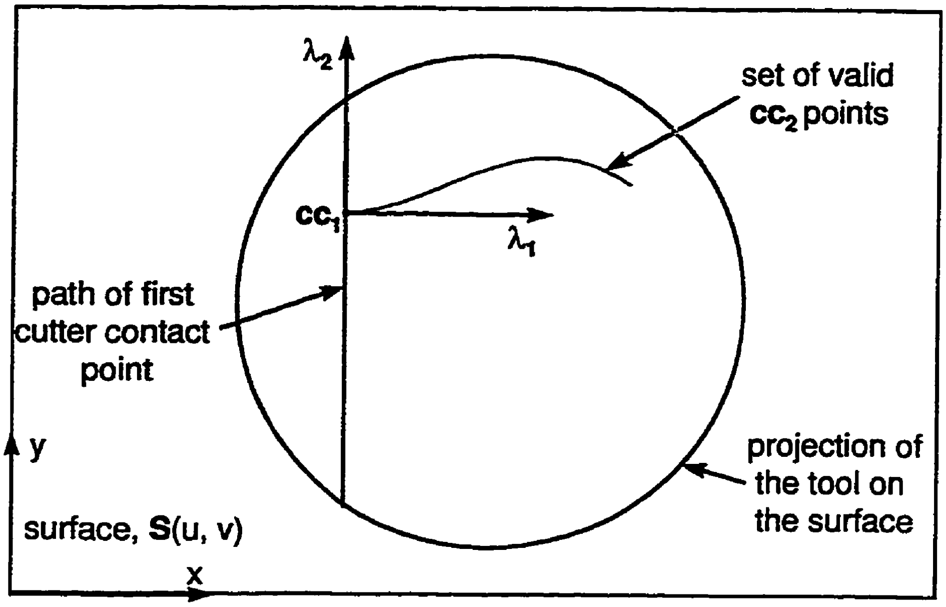


Figure 6.12 Set of valid cc_2 points.

One way to locate the desired second cutter contact point, cc_2, is to search the surface in a line that is approximately perpendicular to the set of valid cc_2 points. A line in the direction of minimum curvature, λ_2, would be ideal for this purpose. However to avoid the need to calculate directions of curvature, the feed direction is generally used instead as shown in Figure 6.13.

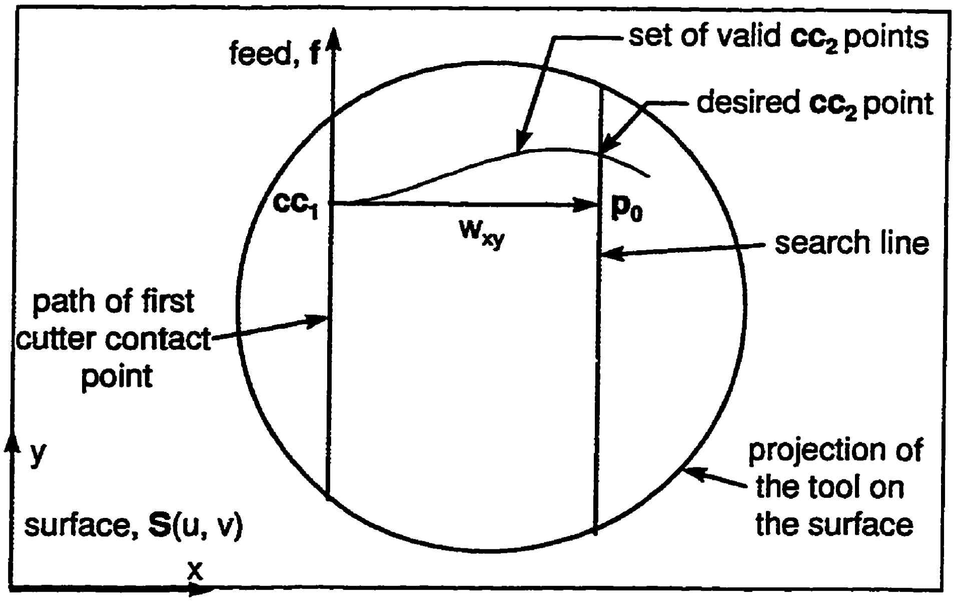


Figure 6.13 Locating the desired cc_2 point using a line search.

The search line is defined in the xy plane and then projected onto the surface. It is located a distance w_{xy} from the path of the first cutter contact point. The distance w_{xy} is calculated as a fraction of the tool pass interval. Typically, w_{xy} is 60-80% of the tool pass interval. This parameter will be investigated in Chapter 7. A point, $\mathbf{p_0}$, on the search line is located by

$$\mathbf{p_0} = w_{xy}\left(\hat{\mathbf{k}} \times \mathbf{f}\right), \tag{6.17}$$

where $\hat{\mathbf{k}}$ is a unit vector on the z-axis and **f** is the feed direction in the xy plane. Note that the $\hat{\mathbf{k}} \times \mathbf{f}$ is a unit vector perpendicular to **f** in the xy plane. A point **p** on the search line is given by:

$$\mathbf{p} = \mathbf{p_0} + d\mathbf{f} \tag{6.18}$$

where d is the distance from point $\mathbf{p_0}$ along the line.

A problem may arise with this search method when the direction of maximum curvature, λ_1, lies close to the feed direction as shown in Figure 6.14. The search line may miss the set of valid $\mathbf{cc_2}$ points entirely. In this circumstance an alternative search strategy is used. Instead of searching along a line, the search is performed along an arc. A point, **p**, on the arc line is given by.

$$\begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} cc_{1_x} + w_{xy}\cos(\theta) \\ cc_{1_y} + w_{xy}\sin(\theta) \end{bmatrix} \tag{6.20}$$

Generally, the line search is used instead of the arc search because the resulting paths of both cutter contact points would lie on straight lines. The resulting surface has a more uniform appearance. However, the arc search is a better choice if the direction of minimum curvature, λ_2 is expected to deviate significantly from the feed direction.

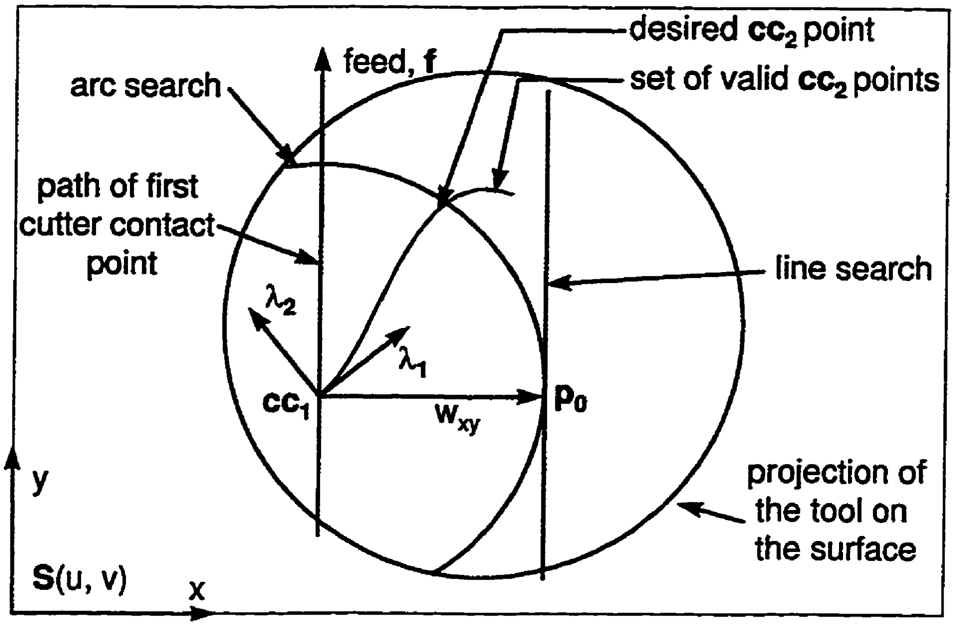


Figure 6.14 Locating the desired cc₂ point using an arc search.

The second cutter contact point is located by searching along one of the two paths until the inside-outside test is satisfied. This process could be carried out by using two different approaches. A root finding technique such as the bisection method or the secant method may be used to determine the location along the search path where the inside-outside test is zero. Alternatively, an optimization technique such as the golden section search may be used to find the minimum of the absolute value of the inside-outside test along the search path. In this work, the absolute value of the inside-outside test is implemented simply because the inside outside test may not have a root if the search path does not intersect the set of valid cc₂ points. The resulting tool position would be unpredictable. This scenario may occur when the separation distance is too large. On the other hand, an optimization technique will attempt to get as close as possible to the desired solution. If the line search misses the valid set of cc₂ points, the optimization technique will still produce the best possible solution. Thus the error function becomes:

$$\text{error} = \left|(x^2+y^2+z^2+R^2-r^2)^2-4R^2(x^2+y^2)\right| \qquad 6.21$$

The optimization is performed along the line or arc using the reliable golden section method. For a given line or arc parameter, d or θ, a potential second cutter contact point is determined. The tool position is the determined using the multi-point tool positioning strategy. Using the resulting tool position vector, $\mathbf{t}_{\mathbf{pos}}$, and the tool axis vector, $\mathbf{t}_{\mathbf{axis}}$, the second potential cutter contact point is transformed into the tool coordinate system using equation 6.16. The inside-outside test given by equation 6.21 is then used to calculate an error at $\mathbf{cc}_2$. Ultimately, a line or arc parameter is found that minimizes the error function. This parameter is used to calculated the correct tool position.

6.2.3 Summary of Surface Algorithm

The flowchart in Figure 6.15 describes the algorithm for the surface approach to tool position adjustment. In this case the line search has been implemented. The golden section search uses this algorithm to calculate the error at a potential cutter contact point. A multi-point tool position is achieved when this error is within tolerance. First, the search line is defined. This line is offset from $\mathbf{cc}_1$ by the separation distance, w, and is parallel to the feed direction, **f**. The point $\mathbf{p}_0$ is the point on the line perpendicular to $\mathbf{cc}_1$. The parameter, d, is the distance along the search line of a potential $\mathbf{cc}_2$ point from $\mathbf{p}_0$. The golden section search determines the values of d. The location of a potential $\mathbf{cc}_2$ point is calculated based on a d value. The multi-point tool positioning strategy explained in Chapter 5 calculates a tool position given the locations of $\mathbf{cc}_1$ and $\mathbf{cc}_2$. Next the potential $\mathbf{cc}_2$ point is transformed into the tool coordinate system. The inside-outside test determines the error at $\mathbf{cc}_2$. If the error is within tolerance the tool adjustment is complete. If not, the golden section search continues.

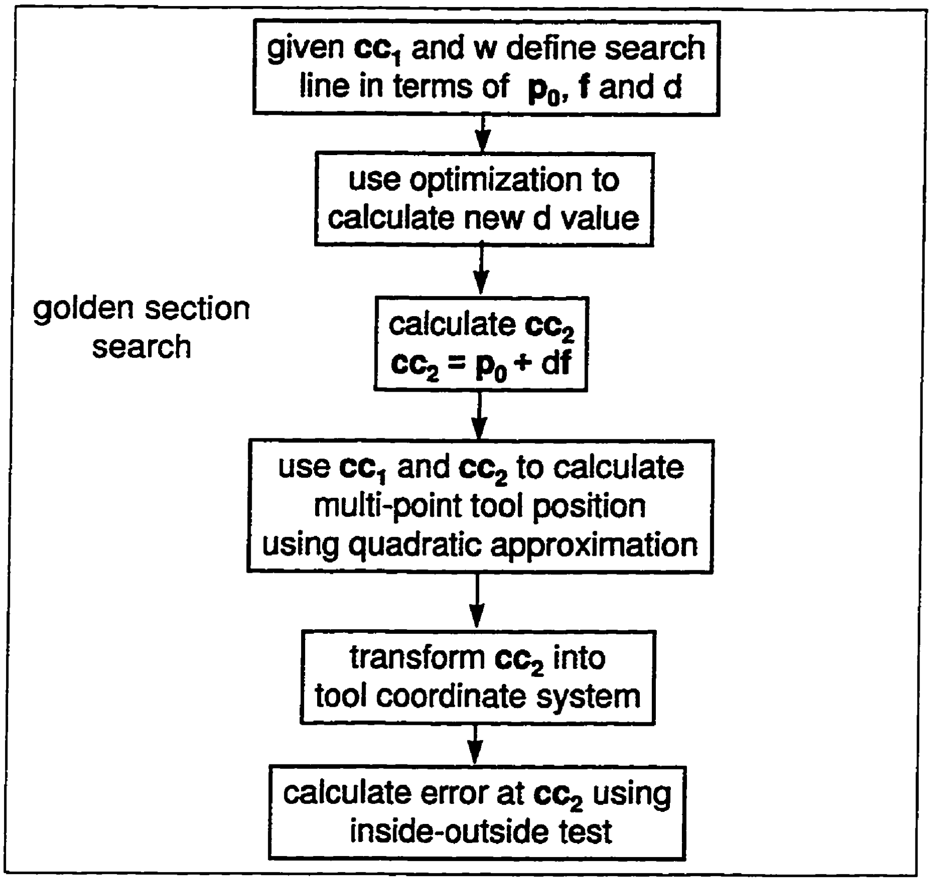

given cc_1 and w define search line in terms of p_0, f and d
use optimization to calculate new d value
golden section search
calculate cc_2
$cc_2 = p_0 + df$
use cc_1 and cc_2 to calculate multi-point tool position using quadratic approximation
transform cc_2 into tool coordinate system
calculate error at cc_2 using inside-outside test

Figure 6.15 The surface approach algorithm.

6.2.4 Example of Surface Approach to Tool Position Adjustment

The test surface, described by equation 6.11 will also be machined using the surface approach. As stated earlier this surface is typical of the type used in the mold and die industry.

The tool path used to machine the workpiece was generated for a cutter contact separation of 6.4 mm and tool pass interval of 8.0 mm. The maximum error at the second contact point was ±0.0001 mm. With this tolerance, the tool correction algorithm required approximately 166 sec to produce a tool path consisting of 18360 tool positions on a Pentium 166. All tool positions produced two point contact.

The resulting tool path was used to machine the test surface on the Rambaudi 5-axis tilt-rotary table milling machine at the University of Waterloo. The cutting conditions used were: spindle speed = 1200 rpm, feed = 200 mm/min, 1.0 mm depth of cut, using a 16 mm diameter toroidal cutter with two 3 mm radius inserts. Figure 6.16 shows the machined surface. The jagged lines formed along the cutter contact points where tool passes overlap are much less pronounced compared to those in Figure 6.9. Measured results were obtained on a coordinate measuring machine, (CMM). The average scallop height was 8 μm which is more then 400 times less then the 3234 μm scallop heights produced by the same sized ball nosed end mill with the same tool path.

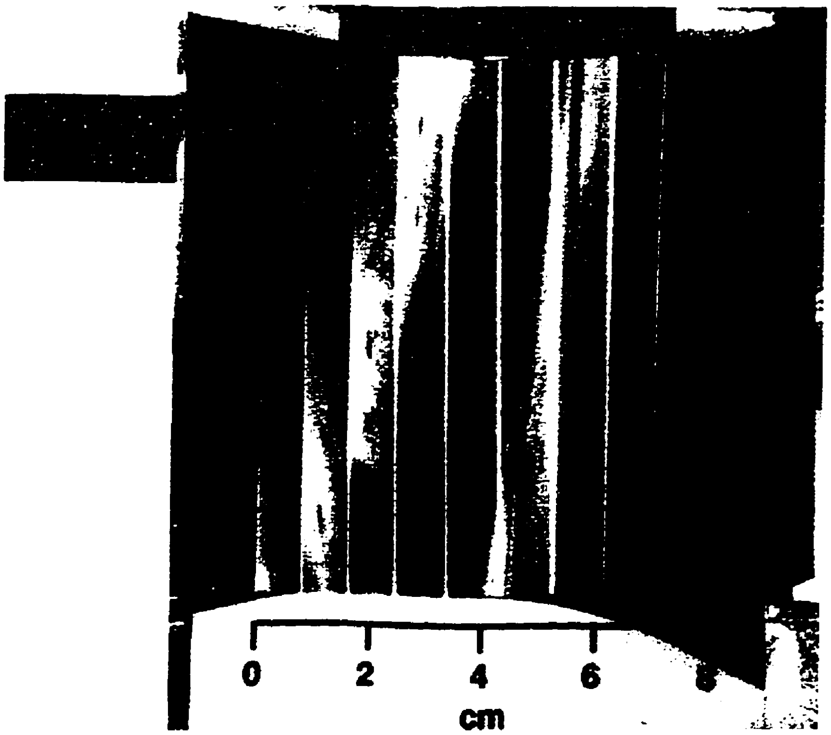


Figure 6.16 Multi-Point Machining using Surface Approach to Tool Position Adjustment

6.3 Comparison of The Two Algorithms

Two methods of adjusting the tool position in order to achieve multi-point contact have been presented in this chapter. In the tool approach, the tool is rotated incrementally; after each rotation the distance between the tool and the surface is used to determine the error at the second cutter contact point. The tool is positioned correctly when this error is within tolerance. In the surface approach, a pair of potential cutter contact point are located on the surface. The multi-point tool positioning algorithm is applied to these points. The resulting tool position will ensure tangential contact at the first cutter contact point. The inside-outside test is then used to determine the error at the second cutter contact point. A search of the surface for a second cutter contact point is conducted. The search is terminated when the error at the second cutter contact point is minimized.

The tool approach tends to be less accurate and less reliable than the surface approach. It requires a line search for every axis of rotation while the surface approach requires only a single line search. This problem is further aggravated by constraining the location of the second cutter contact point. The tool approach often gets stuck at the boundaries. When this occurs new directions for the line search must be calculated. In this situation the solution will often crawl along the constraints requiring a multitude of line searches. However, the tool approach is not without its virtues. This approach makes very few assumptions about the possible nature of contact between the tool and the surface. The surface approach assumes that the second cutter contact point exists on a curve that starts at the first cutter contact point. See for example Figure 6.12. This means that this approach may not work for surfaces that violate this assumption. For instance, at the juncture of two surface patches the set of valid $\mathbf{cc}_2$ points might not be continuous. The line search used by the surface approach may miss the set of valid $\mathbf{cc}_2$ points entirely. Since the tool approach uses several search directions it may be able to hunt down the correct location of $\mathbf{cc}_2$.

Chapter 7

Results

In this chapter the algorithms developed in chapters 5 and 6 will be used to generate multi-point tool paths. Simulations and actual machining of test surfaces will be used to explore some of the features of multi-point machining and compare it with other tool positioning strategies.

This investigation will be carried out in the following manner. Simulations will be used to examine the effect of the tool path parameters on the surface finish produced by multi-point machining. These parameters include: the tool geometry, tool pass interval, feed direction and separation distance. Once the effects of these parameters are understood, multi-point machining will be compared with the ball nose, inclined tool and principle axis methods. This investigation is more demanding than those found in the literature because it will compare the proposed technique to the best of the competing 5-axis techniques and not just the 3-axis ball nose technique. The simulated results will then be verified with cutting tests.

The test surface chosen for the current investigation is shown in Figure 7.1 and defined by:

$$S(u,v)=\begin{bmatrix} S_x \\ S_x \\ S_x \end{bmatrix}=\begin{bmatrix} 80u-20u^2 \\ 120v-20v^2 \\ 15+25v^2-30u-30uv^2+50u^2+30u^2v^2-30u^2v \end{bmatrix} \qquad 7.1$$

This surface was selected because it is typical of the open concave surfaces commonly found in the mold and die industry; it has been used by other researchers [47, 48, 64] and has been accepted by the research community as a reasonable test surface.

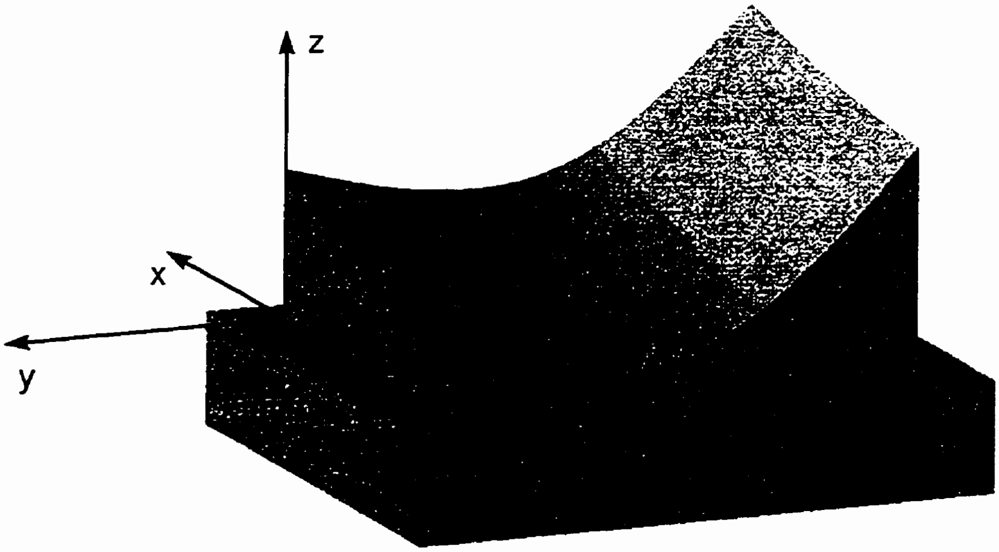


Figure 7.1 Test surface.

The various techniques were compared by performing metal removal simulations using the "mow the grass technique" described in Chapter 2. A 100 X 100 grid of vectors was used to represent the surface. The resulting spatial resolution of approximately 0.1 mm is sufficient to detect most features of the machined surface. However, some features such as sharply pointed scallops may be missed. For this reason two measures of surface finish are used. First, the surface deviations are sorted to determine the maximum scallop height and gouge. These are the traditional measures used to characterize the performance of a tool positioning strategy. In addition, the positive and negative deviations were integrated to approximate the scallop and gouge volumes. The integration tends to average out any errors in the simulation leading to a more robust measure. Instead of just representing the surface by an isolated peak, the amount of material left on the entire surface gives a global measure of surface finish. Furthermore, the volume provides additional information about the scallop geometry. A large scallop height and small scallop volume may indicate that the scallops are tall and skinny. The results of the simulations will be presented in graphical form in the main text of this chapter. They are also tabulated and included in Appendix C.

7.1 Properties of Multi-Point Machining

The effect of tool geometry, feed direction, and separation distance on multi-point machining will be investigated in the following three sections.

7.1.1 Effect of Tool Geometry

One of the most important decisions to be made during tool path planning is to determine the size of the tool. In other words, what values of the torus radius, R, and the insert radius, r, will maximize the tool pass interval for a given surface tolerance. To answer this question, seven sets of simulations were performed for different values of R and r. For each set of simulations the separation distance, w, was the same as the tool pass interval and the feed direction was along the x-axis. The results were calculated for tool pass intervals ranging from 1.0 mm to a maximum value in increments of 1.0 mm. The maximum value was determined by the occurrence of gouging. For example, the maximum gouge when R = 5.0 mm, r = 3.0 mm and tool pass interval = 11.0 mm was approximately 0.7 μm. On the other hand if the tool pass interval was 12.0 mm, the maximum gouge was 15 μm. Therefore, 11.0 mm would be considered the maximum tool pass interval in this case.

The effect of tool dimensions were examined in three different ways. First, the insert radius, r, was held constant at 3.0 mm while the torus radius, R, was assigned the values 3.0 mm, 5.0 mm and 7.0 mm. The resulting maximum scallop heights and volumes are graphed in figures 7.2 and 7.3 respectively. From these results it can be concluded that the largest possible torus radius, R, for a given insert radius, r, will produce the smallest scallops regardless of the tool pass interval. Then, the torus radius, R, was held constant at 5 mm while the insert radius, r, was assigned the values 3.0 mm, 5.0 mm and 7.0 mm. The resulting maximum scallop heights and volumes are plotted in figures 7.4 and 7.5. These figures show that the tool with the largest insert radius, r, for a given torus radius, R, will produce the smallest scallops.

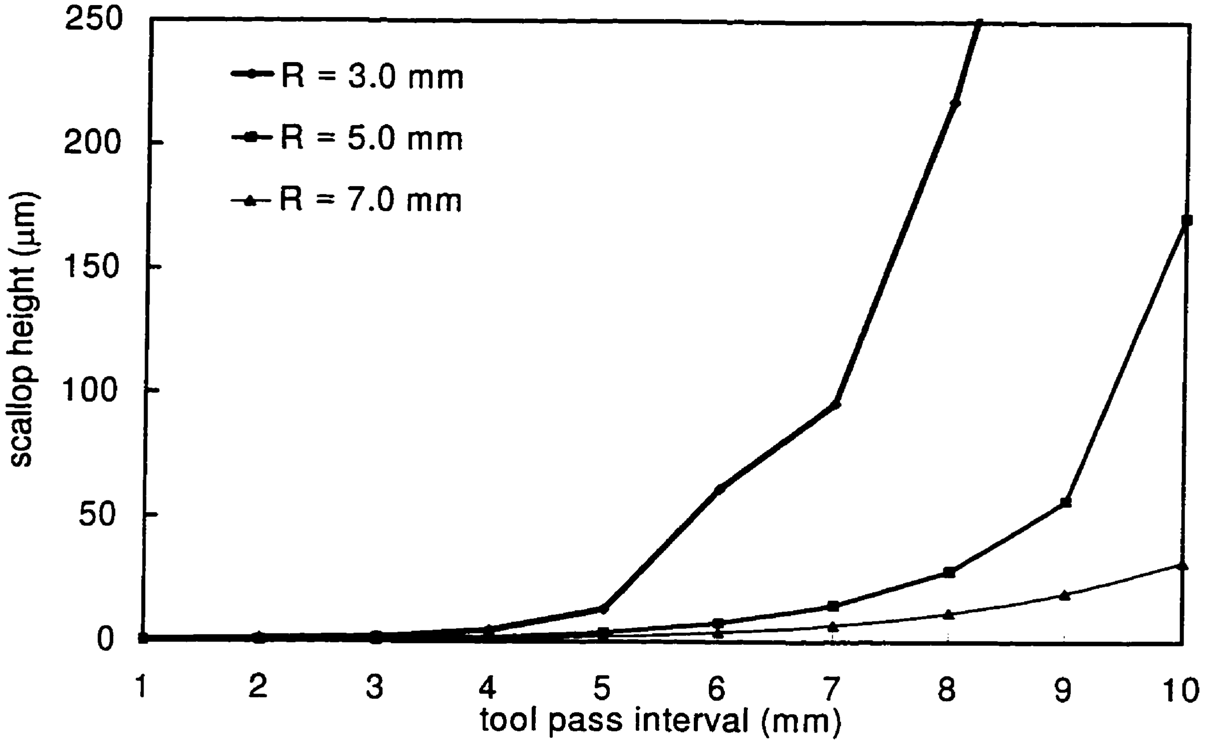


Figure 7.2 Effect of R on scallop height. r = 3.0 mm

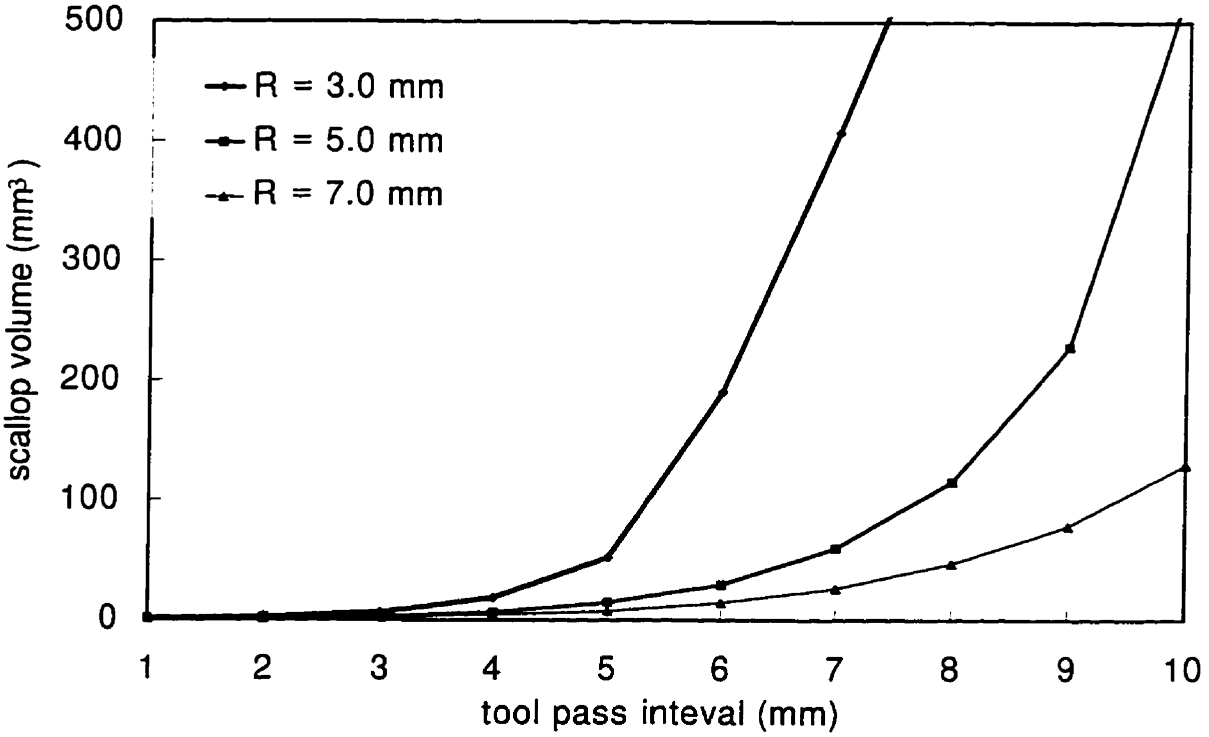


Figure 7.3 Effect of R on scallop volume. r = 3.0 mm

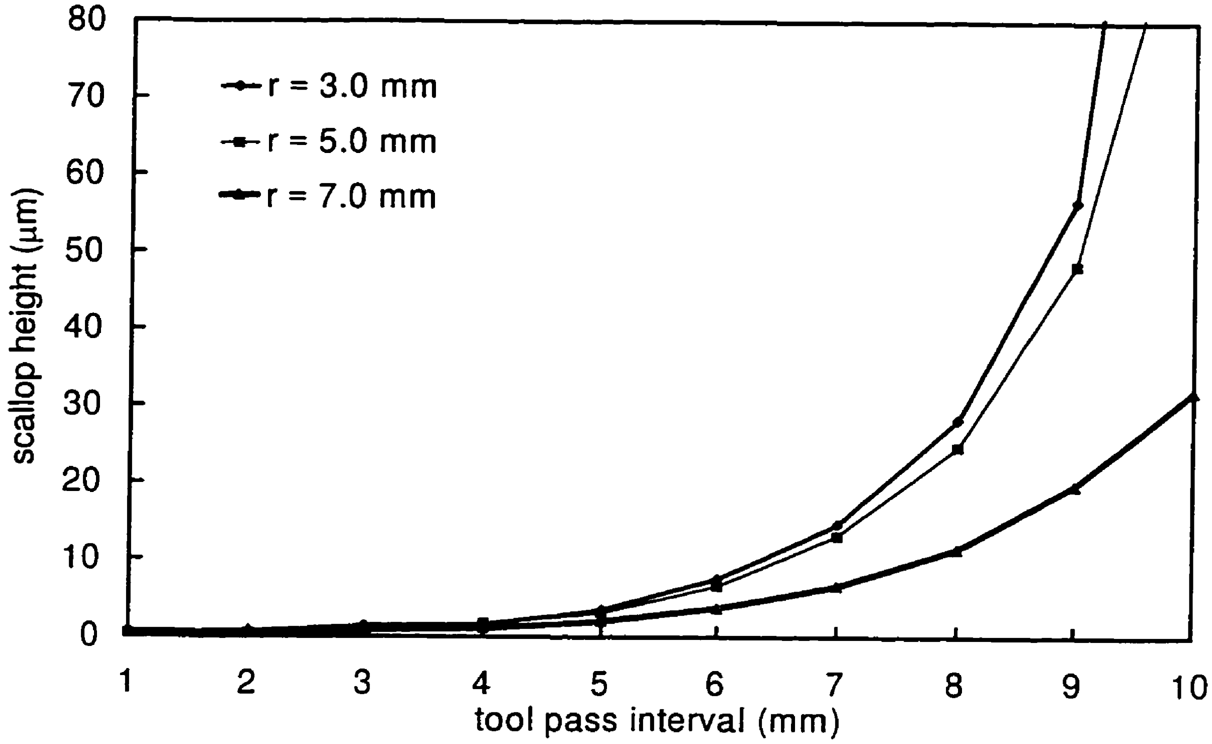


Figure 7.4 Effect of r on scallop height. R = 5 mm

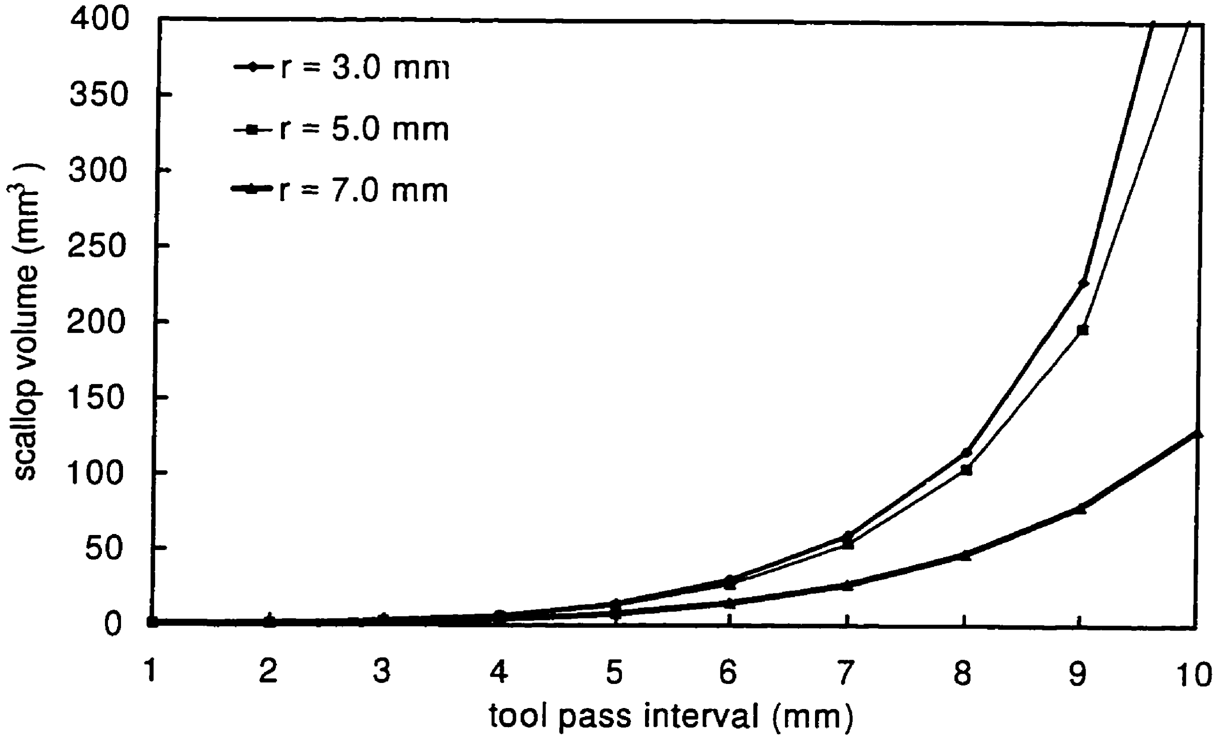


Figure 7.5 Effect of r on scallop volume. R = 5 mm

The proceeding sets of simulations suggest that the largest available tool will produce the best results. However, for a given tool diameter what is the best value of R and r? This question is addressed by the next set of results shown in figures 7.6 and 7.7. In these simulations, the tool radius, R + r, remained constant at 8.0 mm while the tool parameter ratio, r/(R+r) varied. Note that tool parameter ratios of 0.0 and 1.0 correspond to flat and ball nosed end mills respectively. The best results were obtained for a tool parameter ratio of 0.125 indicating that R should be maximized and r should be minimized. In other words, a flat end mill is the best tool choice! This result was also found for the spherical cavity machined in Chapter 3. It should also be noted from figures 7.6 and 7.7 that the maximum tool pass interval declines significantly as the tool parameter ratio decreases. For a tool parameter ratio of 0.875 the maximum tool pass interval was only 5.0 mm.

Do the preceding results mean that there is no benefit from using a toroidal cutter? The answer is "no" for a number of reasons. Mathematically, tool positioning becomes more uncertain with a flat end mill. The discontinuity of the cutting surface at the corner of a flat end mill means that the tangency criterion can not be applied to this type of tool. As a result tangential contact (i.e., non-gouging) at the cutter contact points can not be guaranteed. Secondly, these simulations are for static tool positions based on a mathematical description of the tool's cutting surface. In reality, the tool moves along the surface and individual teeth remove chips of material from the surface. The round inserts of a toroidal cutter tend to leave a much smoother finish than the square teeth of a flat end mill. Finally, the corner of a real flat end mill is never perfectly square. Wear and manufacturing imperfections tend to round the corner making it very difficult to determine the exact geometry of a flat end mill. In addition, a circular insert tends to wear more smoothly than a flat end mill because wear is spread over a longer cutting edge. For these reasons I would choose a toroidal cutter with the smallest available inserts. The Carboloy MM16-0.630-R7.6-MD07 tool had the smallest tool parameter ratio of any of the 16.0 mm diameter tools commercially available; its dimensions were R = 5.0 mm and r = 3.0 mm.

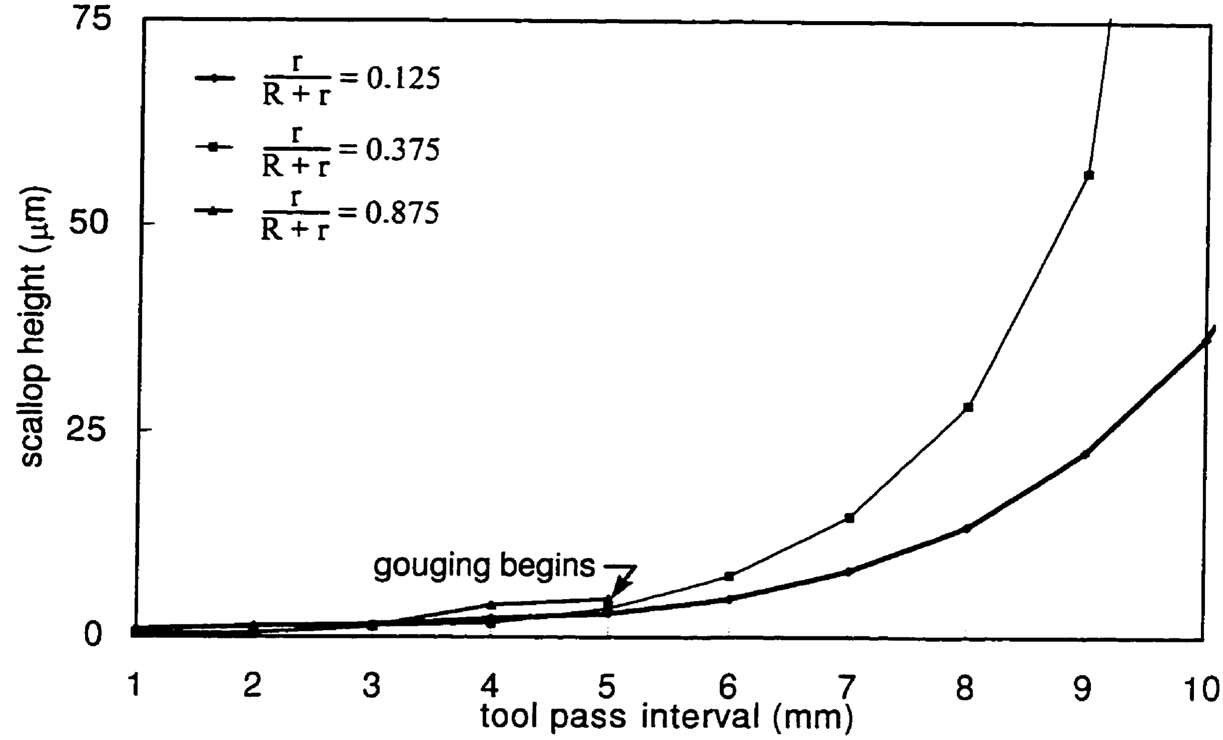


Figure 7.6 Effect of torus dimensions on scallop height. R + r = 8.0 mm

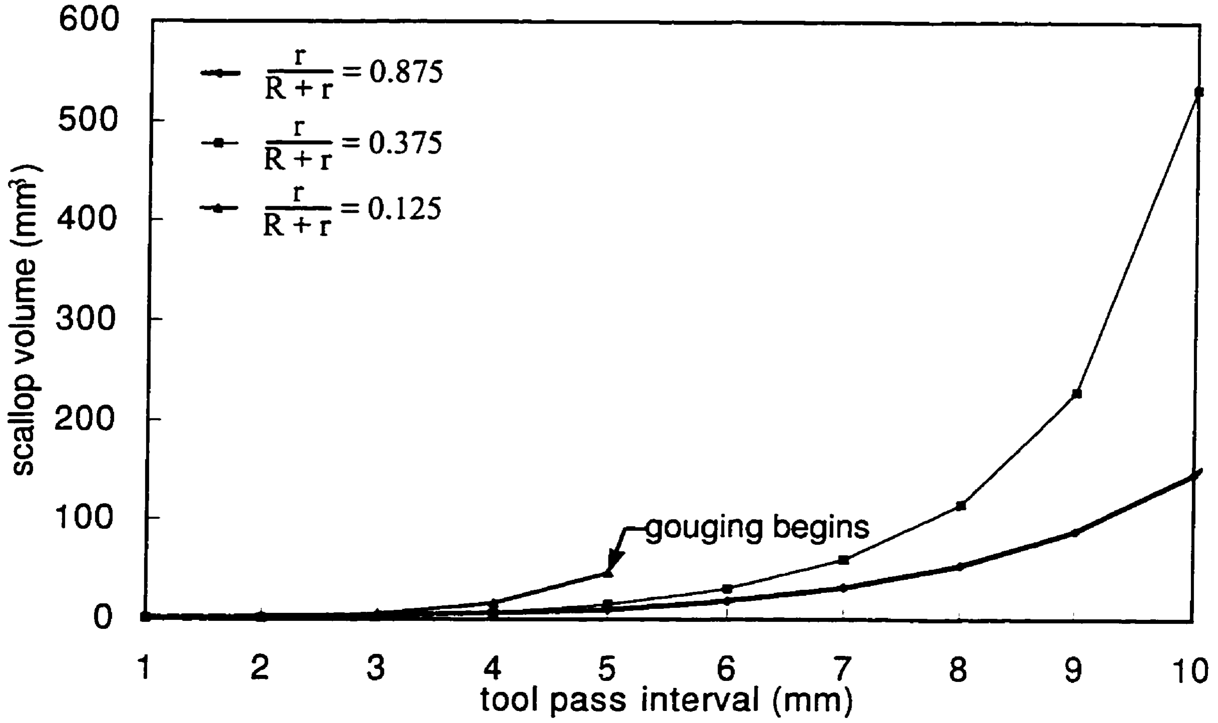


Figure 7.7 Effect of torus dimensions on scallop volume. R + r = 8.0 mm

7.1.2 Effect of Cutter Contact Point Separation Distance

Scallop formation occurs because the tool geometry does not match the surface exactly. This single fact is responsible for the considerable quantity of research into tool positioning. Figure 7.8 compares the scallop formation in single point machining and multi-point machining. In traditional single point tool positioning, scallops are produced between tool passes as shown on the left. For a given tool, these scallops are controlled by the tool pass interval. In multi-point machining, scallops are primarily produced between the cutter contact points as shown in the Figure on the right. This scallop is influenced by the separation distance, w, between cutter contact points. Multi-point machining can also produce scallops between tool position, a phenomenon that will be discussed latter.

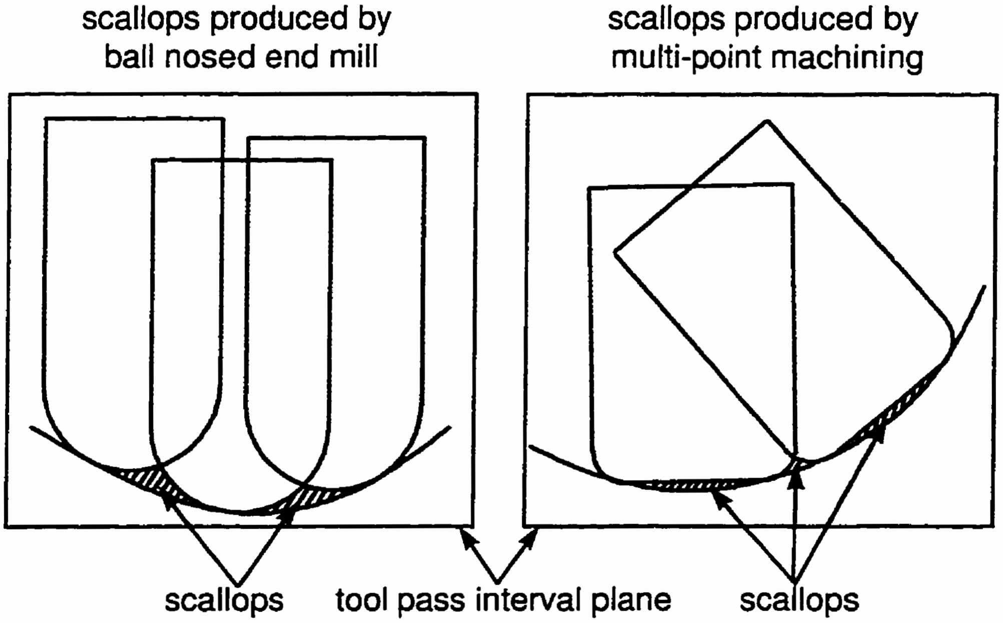

scallops produced by ball nosed end mill
scallops produced by multi-point machining
scallops
tool pass interval plane
scallops

Figure 7.8 Scallop formation for ball nosed and toroidal end mill.

Figure 7.9 shows six surface deviation profiles of a toroidal tool in multi-point contact with the test surface such that the first cutter contact point is the point specified by (u, v) = (0.5, 0.5). These profiles were generated by performing the "mow the grass" simulation for a single tool position and projecting the resulting deviations onto the tool pass interval plane. The resulting profiles are "W" shaped. For small values of the separation distance, w, the graphs look very similar to those expected for a single point of contact. The center part of the

"W" for the small separation distances can only be observed under high magnification. As the separation distance increases, the surface deviation between the cutter contact points become more pronounced.

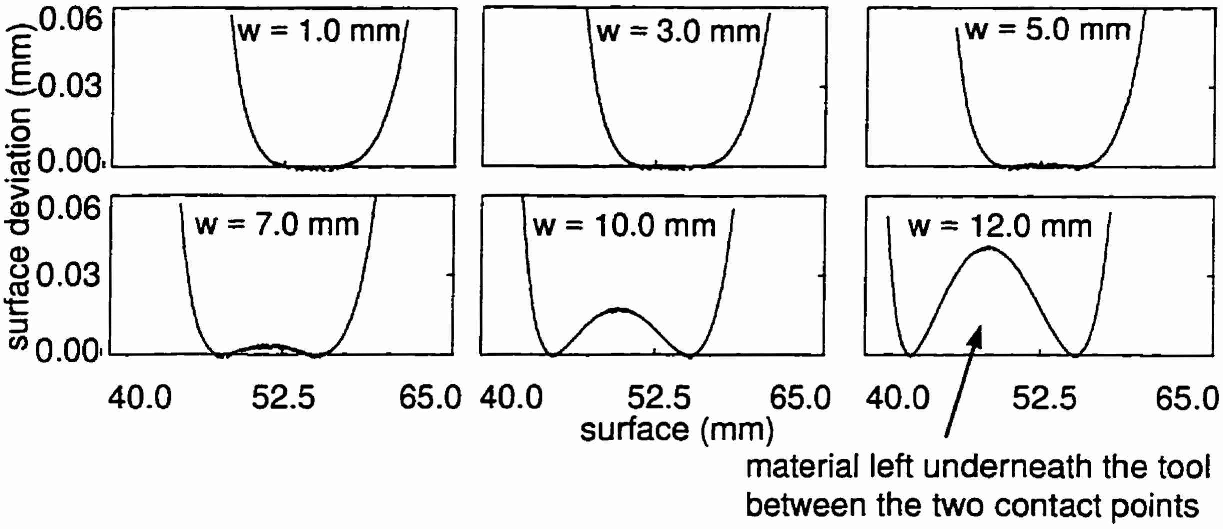


Figure 7.9 Effect of separation distance on multi-point scallop formation
R = 7.9375 mm, r = 4.5625 mm.

The appearance of a scallop underneath the tool adds complexity to tool path planning. Scallop geometry is now influenced by the tool pass interval, χ, and by the separation distance, w. The effects of both χ and w can be combined into one parameter, namely the separation ratio, $\frac{w}{\chi}$. Different types of scallops will be produced depending on the separation ratio as shown in Figure 7.10. If the separation ratio is equal to one, scallops will form only between the cutter contact points. If the separation ratio is less then one, scallops will form between the cutter contact points and between the tool positions. Finally, if the separation ratio is greater than one, the resulting scallops will be produced due to the combination of both scallop formation mechanisms.

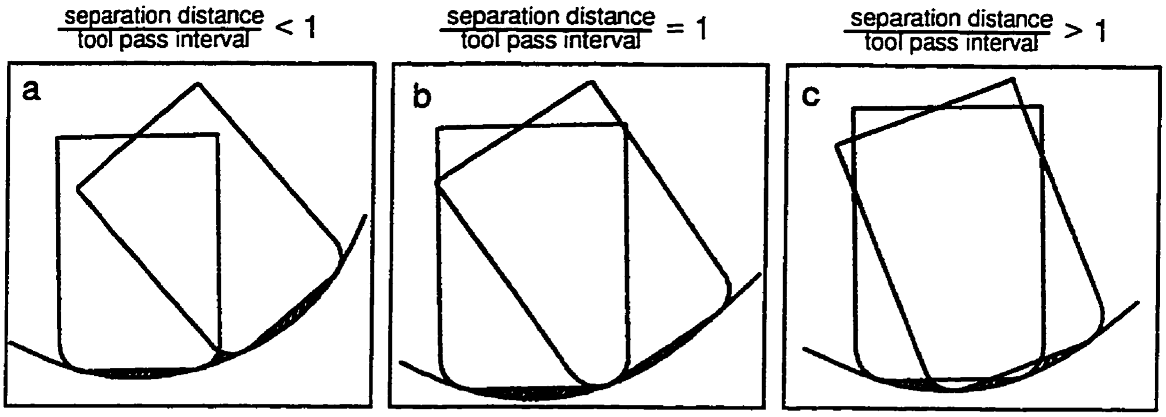

separation distance / tool pass interval < 1
a
separation distance / tool pass interval = 1
b
separation distance / tool pass interval > 1
c

Figure 7.10 Effect of separation ratio in multi-point machining.

A set of simulations were performed using a tool with r = 3.0 mm and R = 5.0 mm for tool pass intervals of 2.0 mm, 5.0 mm and 8.0 mm. In each set of simulations, the separation ratio was varied from 0.4 to 1.8. The results are plotted in figures 7.11 and 7.12 and listed in tables C.12, C.13 and C.14. As with single point machining, scallop height and volume can be reduced by decreasing the tool pass interval. More importantly, the scallop can also be controlled by selecting an optimal separation ratio for a particular tool path interval. For example, the graphs show that the smallest scallop height and volume for a tool pass interval of 5.0 mm can be achieved with a separation ratio of between 0.4 and 0.8. In other words, for this case it is best to have the type of scallops shown in Figure 7.10(a). These results show that there is a need to develop a model based on tool parameters, tool pass interval, separation ratio and surface curvature for predicting scallop geometry. This model will help formulate a multi-point tool path planning strategy. Developing such a model could be the subject for future research.

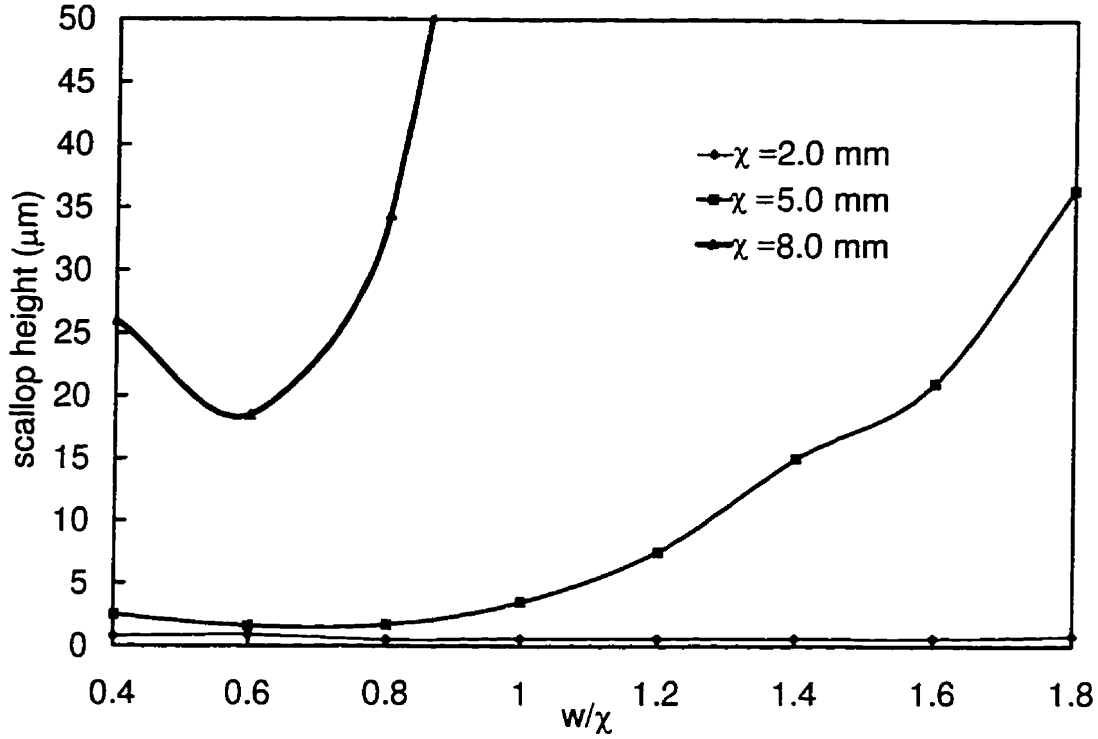


Figure 7.11 Effect of cutter contact separation distance, w, on scallop height.
R = 5.0 mm r = 3.0 mm

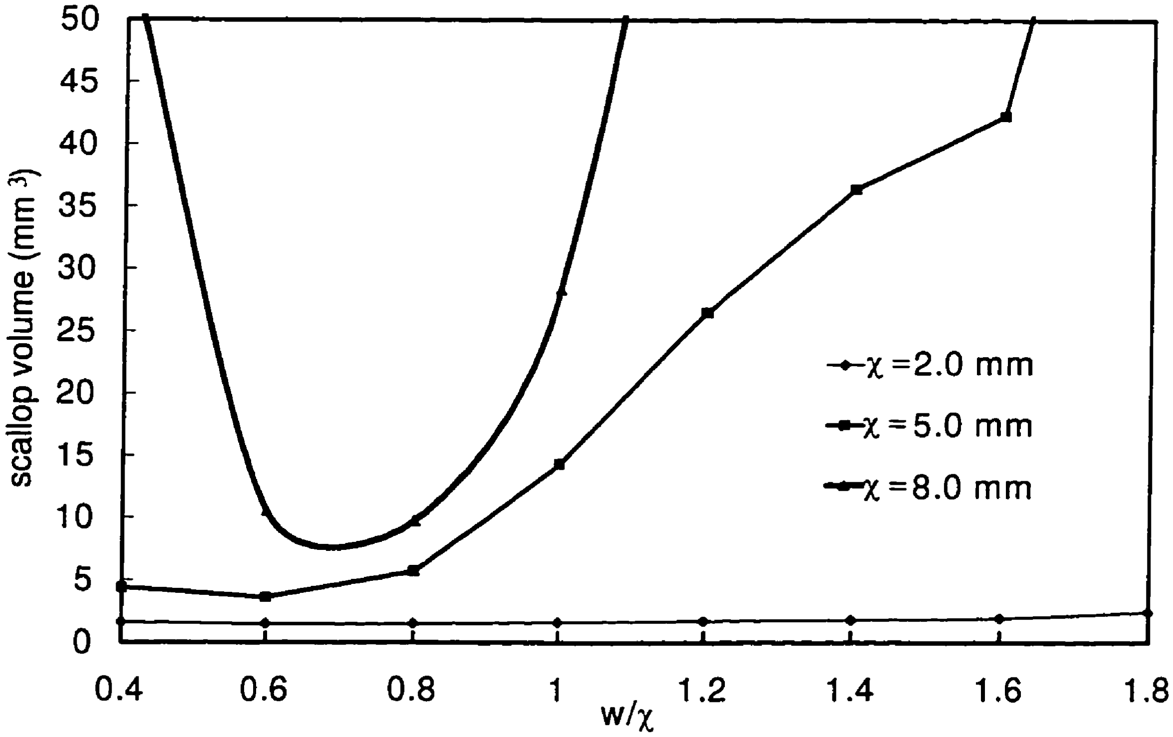


Figure 7.12 Effect of cutter contact separation distance, w, on scallop volume.
R = 5.0 mm r = 3.0 mm

In order to visualize the effect of the separation ratio on the resulting scallops surface, deviations for the test surface are shown in Figure 7.13. The tool path parameters were: R = 5.0 mm, r = 3.0 mm, tool pass interval = 8.0 mm and separation ratio = 0.7. The large round scallops are produced between the cutter contact points and the small sharp scallops are produced between tool positions.

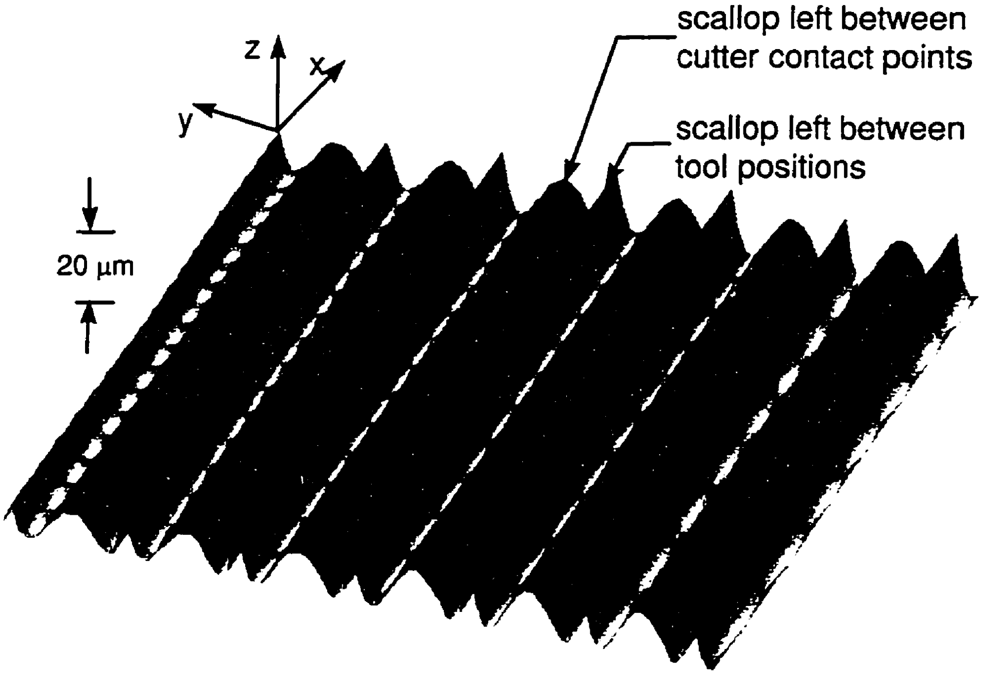

z
x
y
20 μm
scallop left between cutter contact points
scallop left between tool positions

Figure 7.13 Surface deviations for test surface. R = 5.0 mm, r = 3.0 mm, tool pass interval = 8 mm, separation ratio = 0.7

7.1.3 Effect of Feed Direction

This section will examine the effect of feed direction on multi-point tool positioning. Recall that the feed direction is specified in the xy plane during tool path planning. In this section the feed direction will be referenced to the direction of average minimum curvature, λ_2 as shown in Figure 7.14. This direction was obtained by taking the average of the minimum curvatures calculated at 10, 000 evenly spaced points on the surface and projecting the result onto the xy plane. The feed angle is the angle between λ_2 and the feed direction. A feed angle of zero means that the feed direction is the same as the direction of minimum curvature. Note that feed direction has no effect on the location of the cutter contact points, **cc**$_1$ and **cc**$_2$. They always lie in the direction of maximum curvature, λ_1. However, the projection of the cutter contact points onto the tool interval plane will change as the feed direction and thus tool pass interval plane rotates.

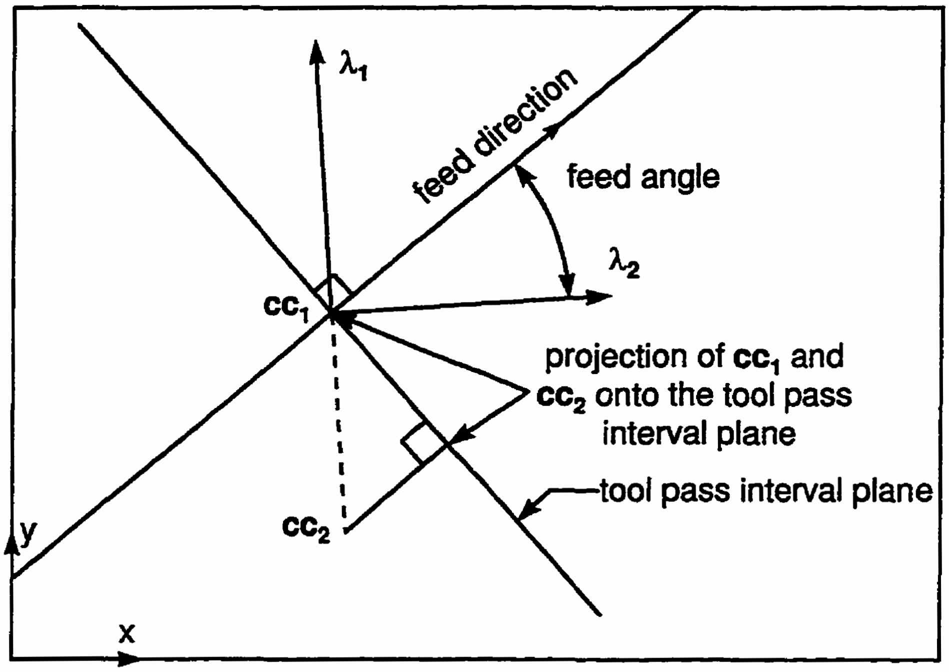


Figure 7.14 Feed angle.

Simulations were performed for feed angles between -50° and 50° at 10° intervals for the following parameters: R = 5.0 mm, r = 3.0 mm, separation distance = 5.0 mm, and tool pass interval = 5.0 mm. The results, plotted in Figure 7.15, show that the best feed direction is along the direction of minimum curvature. The reason for this result is illustrated in Figure 7.16. On the left, the tool pass interval plane is shown for machining in the direction of minimum curvature, λ_2. On the right, the tool pass interval plane is shown for machining in the direction of maximum curvature, λ_1. In each plane the tool and surface profiles have been drawn. When the feed is in the direction of minimum curvature, the projection of the cutter contact points, **cc**$_1$ and **cc**$_2$, onto the tool pass interval plane are at their maximum separation. The tool profile closely matches the surface and the result is small scallops. When the feed is in the direction of maximum curvature the projected cutter contact points are at their minimum separation. In fact, the projected points lie on top of each other. The tool profile does not match the surface in the tool pass interval plane very closely and the resulting scallop is large.

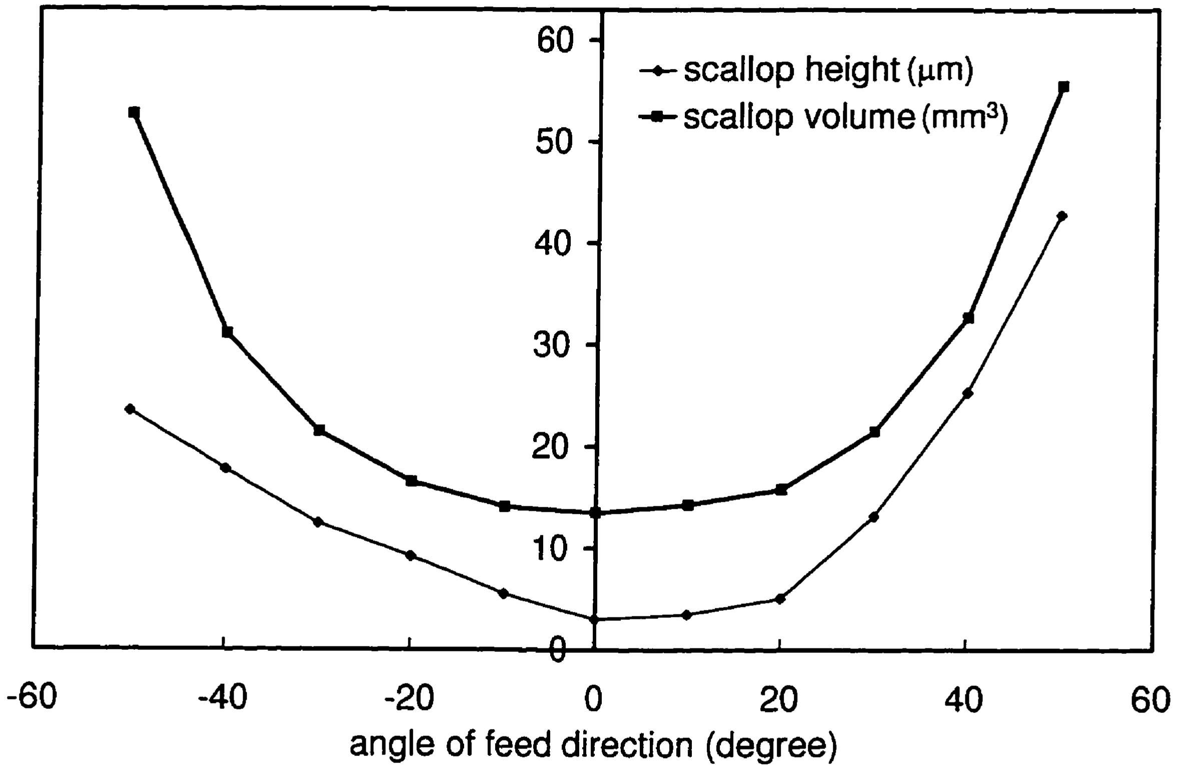
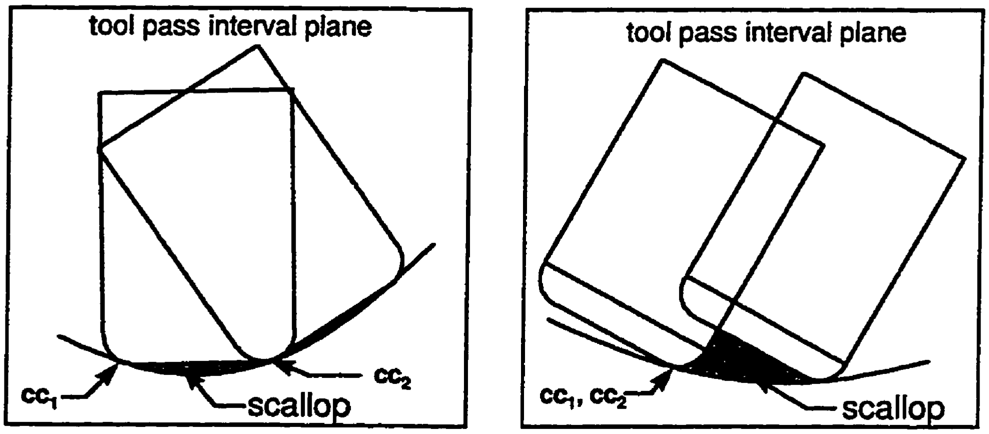


Figure 7.15 Effect of feed direction on scallop height and volume.
R = 5.0 mm, r = 3.0 mm, w = 5.0 mm, χ = 5.0 mm



feed direction same as direction of minimum curvature

feed direction same as direction of maximum curvature

Figure 7.16 Effect of feed direction on scallop.

7.2 Comparison of Tool Positioning Strategies

The multi-point tool positioning strategy will now be compared to the most popular competing tool positioning strategies: ball, inclined tool and the principle axis technique. This comparison will be accomplished by machining the test surface using each strategy. The same tool path and the same tool diameter, 16.0 mm, will be used in each simulation.

7.2.1 Ball Nosed Tool Positioning

Machining using a ball nosed cutter was the first method developed for sculptured surfaces and remains the most popular. Ball nosed tool positioning can be explained using Figure 7.17. It shows a ball nosed tool in tangential contact with a surface at the cutter contact point, **cc**. Since the cutting surface of the tool is spherical, the tool axis, $\mathbf{t_{axis}}$, has no effect on scallop geometry. Therefore, this comparison will use $\mathbf{t_{axis}} = [0, 0, 1]^T$. The tool position is offset along the surface normal, **n**, by a distance equal to the tool radius, r.

$$\mathbf{t_{pos}} = \mathbf{cc} + r\mathbf{n} \qquad 7.1$$

Thus, the tool path is generated by offsetting the tool position from each cutter contact point and aligning the tool axis with the z-axis. Figure 7.18 shows the surface deviations from the simulation using a ball nose cutter. This figure clearly illustrates the characteristic sharp scallops generated during single point machining. The maximum scallop height and scallop volume was 496 μm and 554 mm^3 respectively.

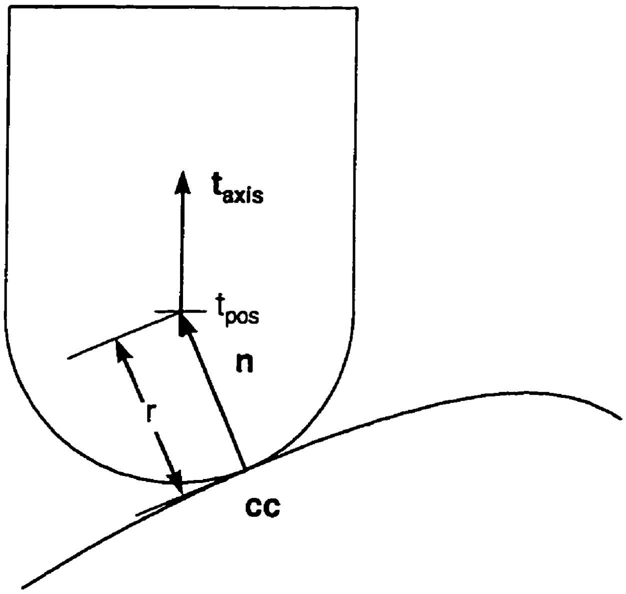


Figure 7.17 Positioning a ball nosed tool in 3-axis.

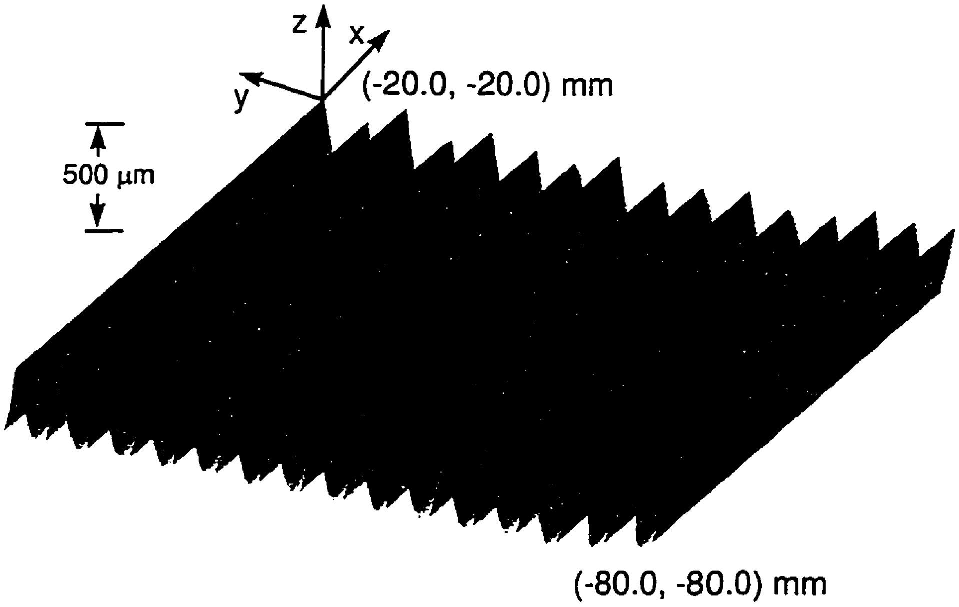


Figure 7.18 Surface deviations produced by a ball nosed end mill.
r = 8.0 mm, tool pass interval = 4.0 mm

7.2.2 Inclined Tool

The inclined tool positioning strategy also known as Strutz milling is the next most common of the 5-axis tool positioning strategies. It has been implemented in several high end CAM packages and has been shown to be far superior to the ball nosed technique on many occasions. For these reasons it is the most appropriate benchmark for other 5-axis tool positioning strategies.

Figure 7.19 shows how the inclined tool method works for a toroidal cutter. The tool axis, $\mathbf{t}_{\mathbf{axis}}$, is inclined in the feed direction by ϕ. This inclination angle is often called the Strutz angle. The tool position, $\mathbf{t}_{\mathbf{pos}}$, is calculated such that the tool is placed in tangential contact with the surface at **cc**.

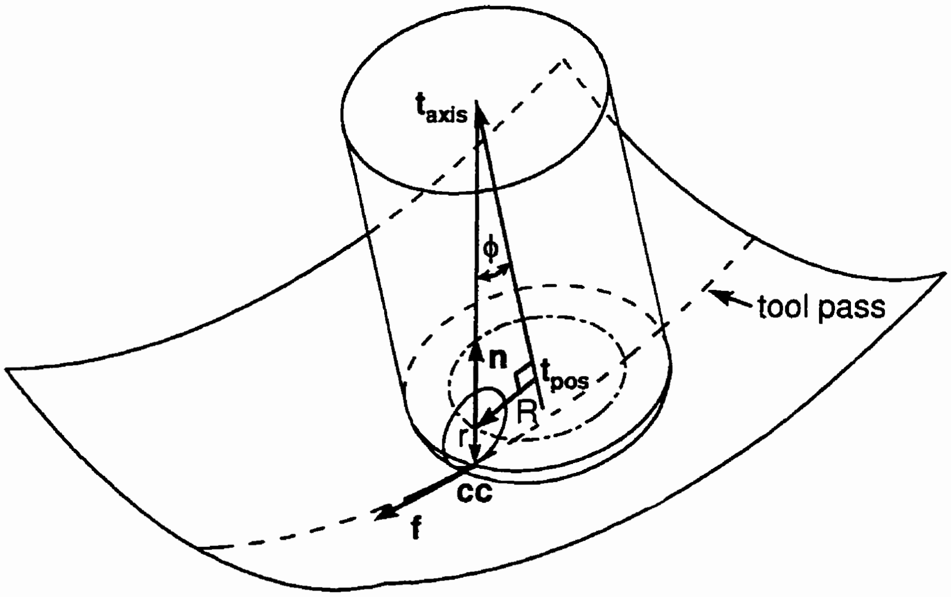


Figure 7.19 Positioning an inclined toroidal cutter.

The tool can be positioned by considering the plane containing the tool axis and the feed direction, **f**. This tool axis plane is shown in Figure 7.20. In order to use planer geometry, a coordinate system must be created in this plane at the insert center, **c**, which is located by:

$$\mathbf{c} = \mathbf{cc} + r\mathbf{n} \tag{7.2}$$

One of the coordinate axes will be the surface normal, **n**. The other coordinate axis must be perpendicular to **n** and yet lie in the plane. This vector, **e**, can be constructed using a triple vector product.

$$\mathbf{e} = \mathbf{n} \times (\mathbf{n} \times \mathbf{f}) \tag{7.3}$$

The tool axis is then calculated by:

$$\mathbf{t}_{\mathbf{axis}} = \cos(\phi)\mathbf{n} + \sin(\phi)\mathbf{e} \tag{7.4}$$

and the tool position is given by:

$$\mathbf{t}_{\mathbf{pos}} = \mathbf{c} + R\sin(\phi)\mathbf{n} - R\cos(\phi)\mathbf{e} \tag{7.5}$$

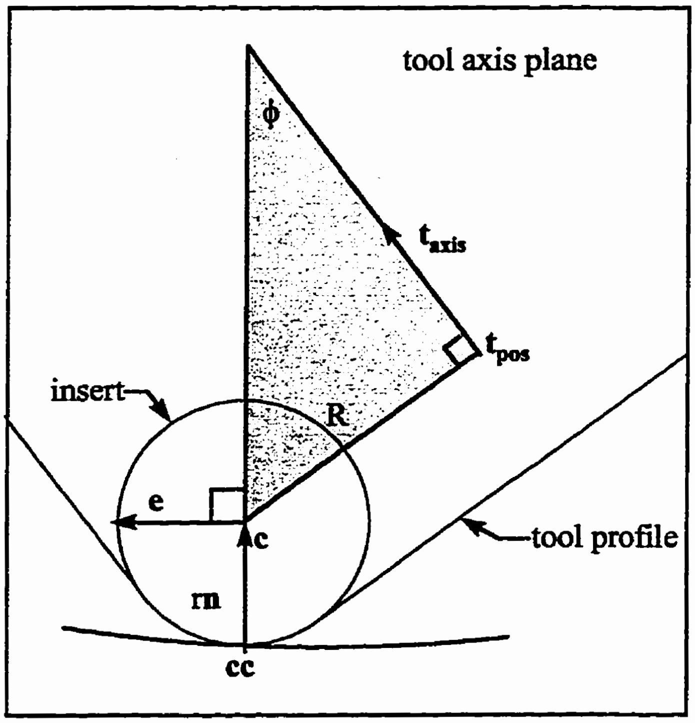


Figure 7.20 The tool axis plane for an inclined tool.

The tool paths for the incline tool method were generated by calculating the tool position and tool axis using equations 7.4 and 7.5 for each cutter contact point. These equations require the value of the inclination angle, ϕ, be specified prior to tool positioning.

A small value of ϕ will produce gouging and a large value will result in unnecessarily large scallops. A value of 6° was used for the inclination angle in the simulations. This value was selected by performing a series of simulations with progressively larger angles as shown in table 7.1. The smallest value that did not result in gouging was selected.

Incline angle, ϕ (degree)	Max gouge (μm)	Volume of gouged material (mm^3)
1	222.6	469.7
2	144.0	150.8
3	80.7	31.2
4	36.4	3.3
5	6.3	0.1
6	0.1	0.0
7	0.1	0.0

Table 7.1 Effect of inclination angle on 5-axis machining with an inclined toroidai cutter. r = 3.0 mm, R = 5.0 mm, 5.0 mm, tool pass interval = 8.0 mm, ϕ = 6°

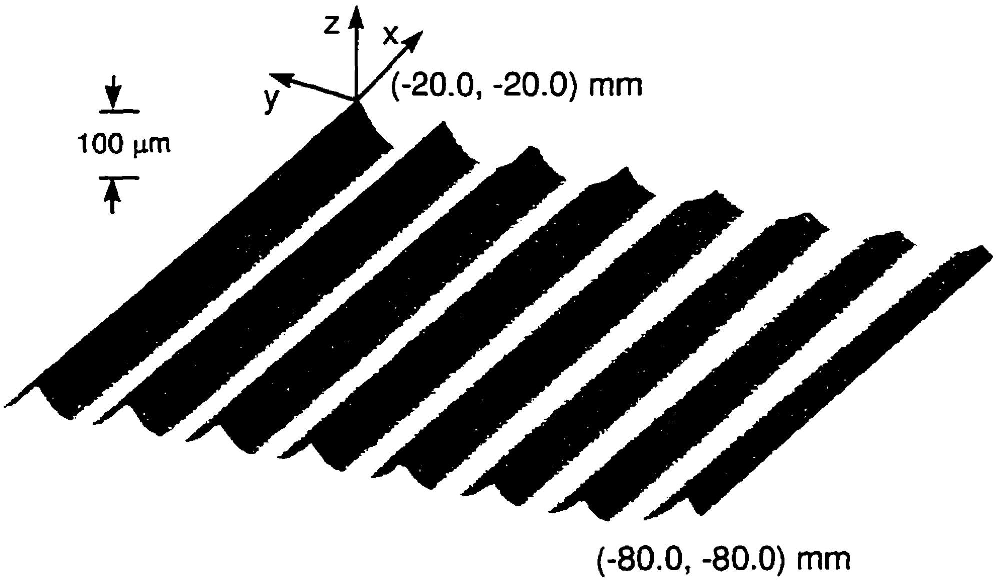


Figure 7.21 Surface deviations produced by an inclined toroidal cutter. R = 5.0 mm, r = 3.0 mm, tool pass interval = 8.0 mm, ϕ = 6°

Figure 7.21 shows a section of the surface deviations produced by an inclined tool simulation. The maximum scallop height and scallop volume was 111.5 μm and 96.7 mm^3 respectively. Even with twice the tool pass interval these values were less than a quarter of those produced by the ball nosed tool. The scallops are not very uniform because this method does not include curvature information. The scallop size varies as the curvature changes.

7.2.3 Principle Axis Method

The principle axis method is a modification of the inclined tool method. It was formulated to account for surface curvature. In the principle axis method, curvature information is incorporated into the tool position by modifying the tool axis plane and computing the inclination angle based on the curvature at the cutter contact point as shown in Figure 7.22. Rao et al. [48] showed that tool inclination could be optimized by inclining the tool in the direction of minimum curvature, λ_2, such that the minimum curvature of the tool equals the maximum curvature of the surface, κ_1.

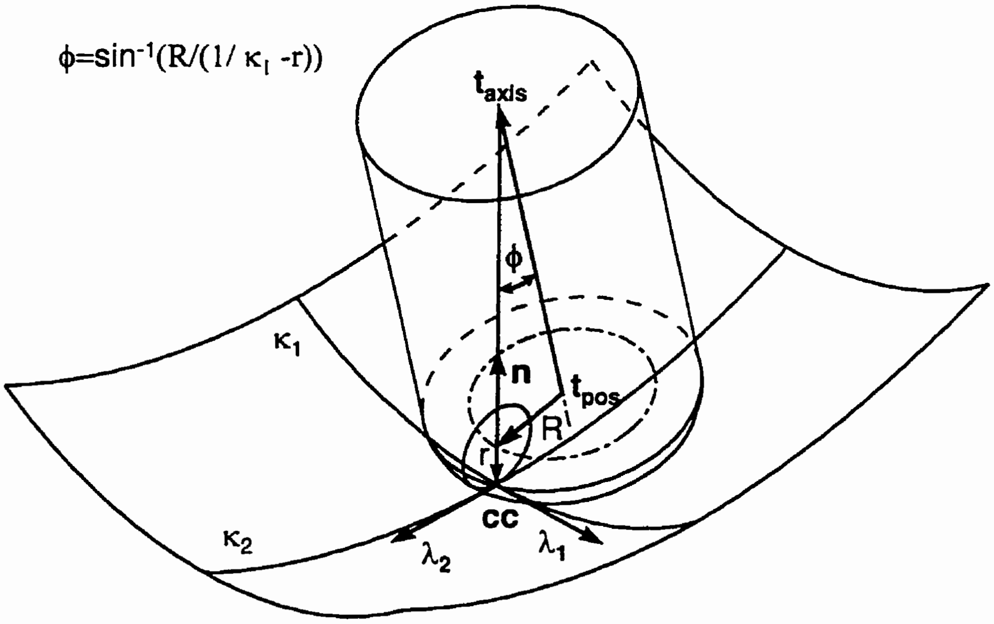


Figure 7.22 Principle axis method.

The tool axis plane shown in Figure 7.20 can also be used for the principle axis method with some slight modifications. The tool axis plane will contain the surface normal, **n**, as in the inclined tool method but it will not contain the feed direction. Instead the plane will include the direction of minimum curvature, λ_2. Once again, a coordinate system must be created in this plane at the insert center, **c**, which is located by:

$$\mathbf{c} = \mathbf{cc} + r\mathbf{n} \tag{7.6}$$

The coordinate axes will be the surface normal, **n**, and the direction of minimum curvature, λ_2.

$$\mathbf{e} = \lambda_2 \tag{7.7}$$

The tool axis is then calculated by:

$$\mathbf{t}_{\mathbf{axis}} = \cos(\phi)\mathbf{n} + \sin(\phi)\mathbf{e} \tag{7.8}$$

and the tool position is given by:

$$\mathbf{t}_{\mathbf{pos}} = \mathbf{c} + R\sin(\phi)\mathbf{n} - R\cos(\phi)\mathbf{e} \tag{7.9}$$

All that remains is to calculate the inclination angle, ϕ. The minimum curvature of a toroidal cutter, κ_{t2}, is given by [44]:

$$\kappa_{t2} = \frac{\sin(\phi)}{R + r\sin(\phi)} \tag{7.10}$$

Therefore, the inclination angle can be found by substituting the maximum curvature of the surface, κ_1, for κ_{2t} and solving for ϕ.

$$\phi = \sin^{-1}\left(\frac{\kappa_1 R}{1 + \kappa_1 r}\right) \tag{7.11}$$

Figure 7.23 shows a section of the surface deviations produced in simulation by the principle axis method. The maximum scallop height and scallop volume were 23.5 μm and 14.8 mm^3 respectively; they are less than a quarter of those produced by the inclined tool for the same tool pass interval. Furthermore, the scallop size is much more uniform across the entire surface. Clearly, there is an advantage to incorporating curvature information into tool positioning.

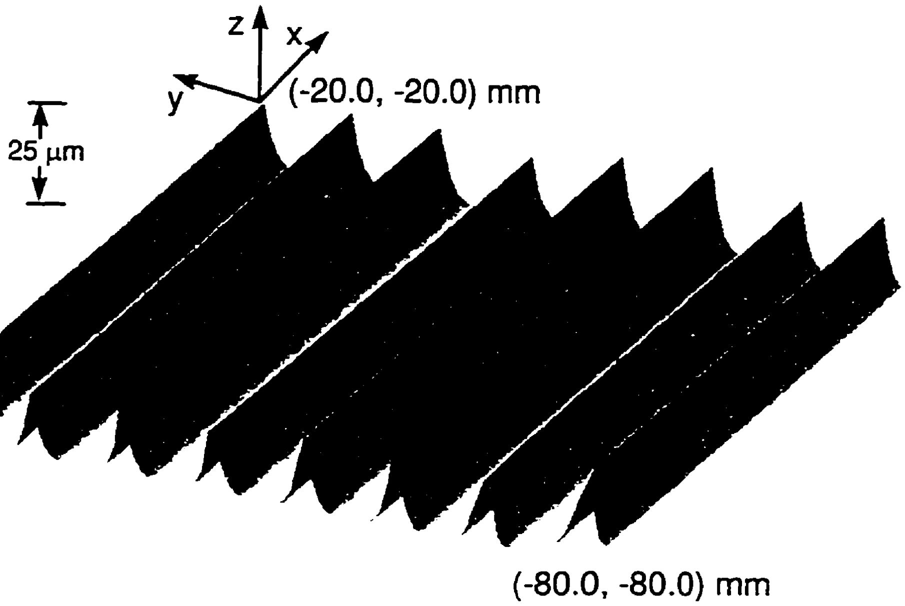


Figure 7.23 Surface deviations produced by the principle axis method. R = 5.0 mm, r = 3.0 mm, tool pass interval = 8.0 mm

7.2.3 Multi-Point Machining

The multi-point tool path was generated using the surface approach to tool position adjustment as discussed in Chapter 6. Figure 7.24 shows an example of the surface deviations produced by multi-point machining with the same tool and tool pass interval used for the inclined tool and principle axis methods. The separation ratio was 0.8. Once again, the two distinct scallop shapes are clearly visible. The sharp scallops are much smaller than those observed in Figure 7.13 because the separation ratio was larger. Note also that the scallop size is fairly even across the surface. The multi-point tool positioning strategy accounts for changes in surface curvature although curvatures are never calculated; a clear advantage when machining a poorly defined surface. The maximum scallop height was about half of that of the principle axis technique at 9.5 µm. The scallop volume was only slightly better than the principle axis method at 13.9 mm^3.

The comparison among the 5-axis tool positioning techniques was conducted for a wider range of tool pass intervals. The results are shown in figures 7.25 and 7.26. In these figures the scallop heights and volumes are graphed and tabulated for each of the methods for different tool pass intervals. In general, the maximum scallop heights for multi-point machining is about 400, 25 and 2 times smaller than ball, inclined tool and PAM scallop heights, respectively. The corresponding scallop volumes are about 350, 12 and 1.1 times smaller respectively.

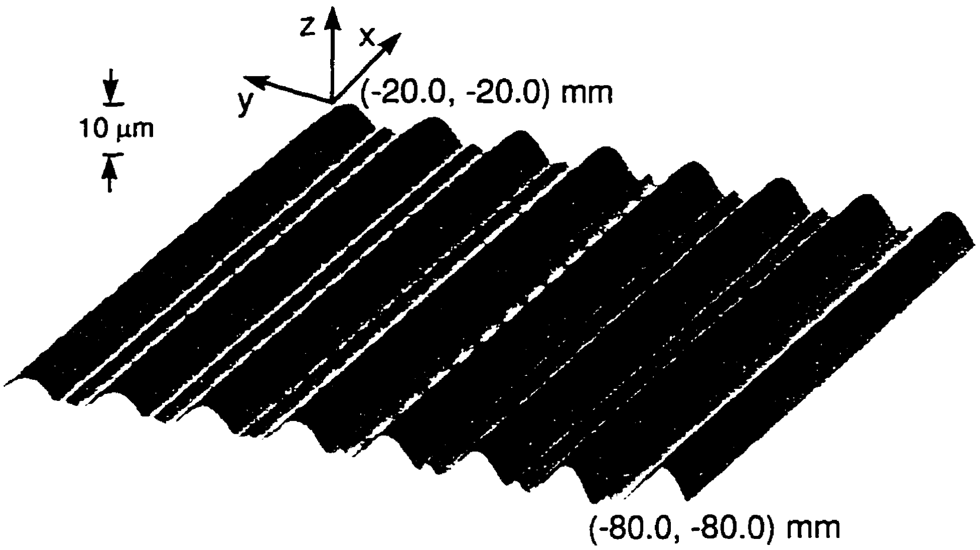
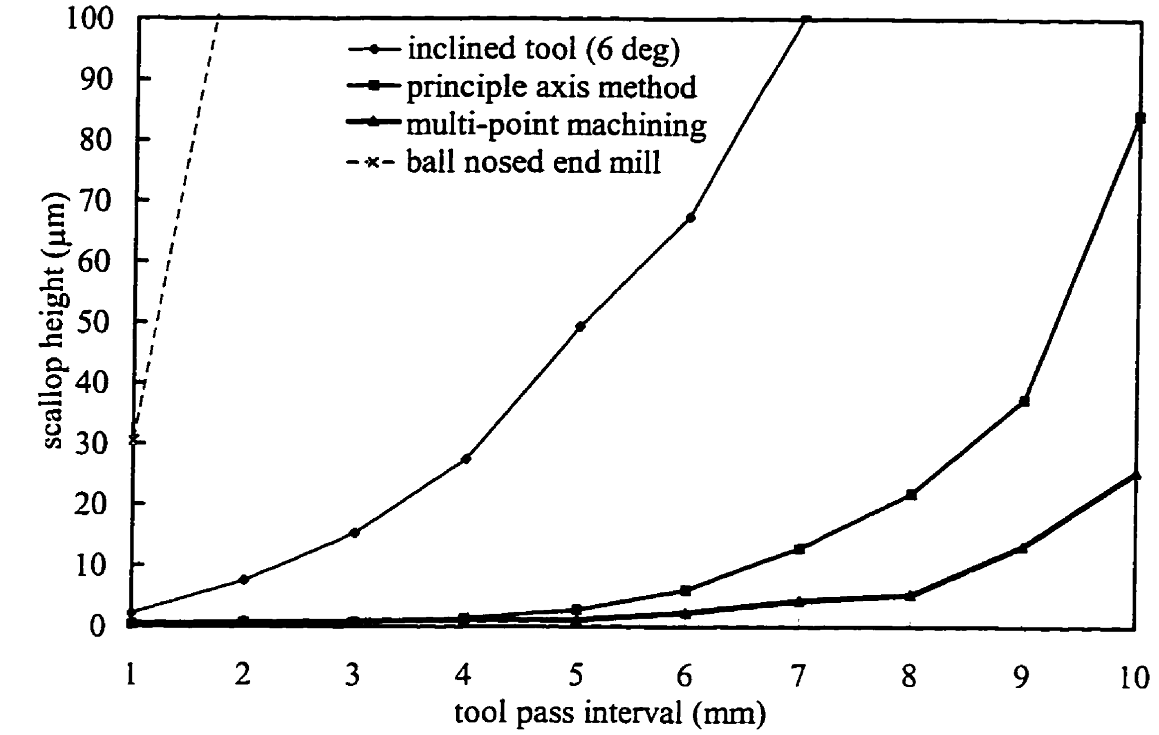


Figure 7.24 Surface deviations produced by multi-point machining. R = 5.0 mm, r = 3.0 mm, tool pass interval = 8.0 mm, separation ratio = 0.8

Increased performance comes at the expense of increased computational effort and surface requirements. Computational time was assessed by generating tool paths for each method that would result in a maximum scallop height of 0.1 mm. It took 121, 58 and 53 seconds on a 166 MHz Pentium to perform the tool positioning computations for the multi-point, principle axis and inclined tool methods. Ironically, tool positioning for the ball nose tool required 256 seconds due to the large number of tool passes required. The principle axis method requires surfaces for which the curvature must be calculated accurately while the multi-point, inclined tool and ball nosed techniques can be implemented with only surface normal information.

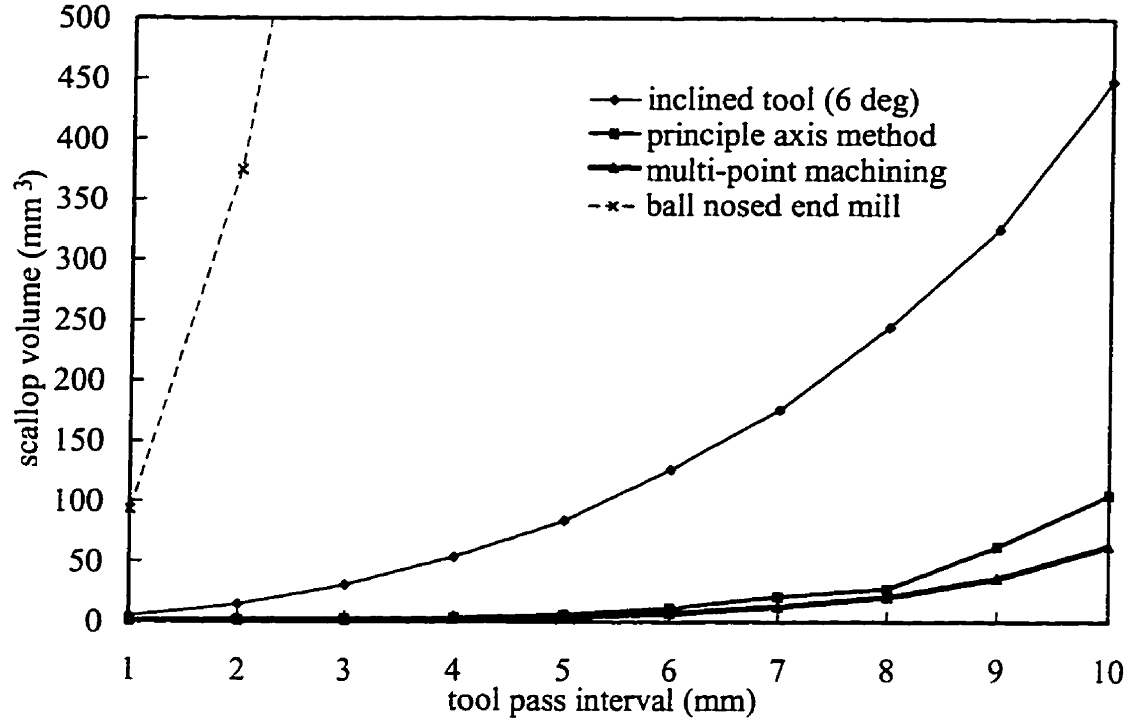


(a)

Tool pass interval (mm)	3-axis method	5-axis methods		
	ball	Inclined tool (6°)	Principle axis method	Multi-point machining
1	30.5	2.1	0.4	0.4
2	129.3	7.5	0.6	0.6
3	289.5	15.3	0.7	0.9
4	518.8	27.5	1.3	1.2
5	850.4	49.4	2.7	1.6
6	1353.8	67.5	6.0	3.0
7	2082.0	100.5	12.9	5.3
8	2705.6	132.7	21.9	9.5
9	-	162.2	37.5	16.2
10	-	282.1	84.3	27.0

(b)

Figure 7.25 Comparison of scallop height(μm) for different techniques. (a) graphed results, (b) tabulated results, torus dimensions: R = 5 mm, r = 3 mm, ball dimensions: r = 8 mm, separation ratio = 0.8 for MPM



(a)

Tool pass interval (mm)	3-axis method	5-axis methods		
	ball	Inclined tool (6°)	Principle axis method	Multi-point machining
1	93.5	4.8	1.1	1.0
2	374.7	14.4	1.6	1.5
3	853.2	30.4	2.1	2.1
4	1541.4	53.5	3.2	3.2
5	2427.2	83.8	5.9	5.6
6	3591.2	126.5	11.1	10.4
7	5113.8	176.1	20.8	18.7
8	6995.3	244.5	27.3	33.0
9	-	325.4	62.3	58.6
10	-	448.6	105.3	99.9

(b)

Figure 7.26 Comparison of scallop volume (mm^3) for different techniques. (a) graphed results, (b) tabulated results, separation ratio = 0.8 for MPM, torus dimensions: R = 5 mm, r = 3 mm, ball dimensions: r = 8 mm

7.3 Experimental Results

Workpieces were machined using the inclined tool, principle axis and multi-point methods in order to verify the simulated results. Cutting tests were conducted on a Rambaudi milling machined that had been retrofitted with a tilt-rotary table to provide 5-axis machining capability. Machining with a ball nose was not performed because it has already been shown on numerous occasions to be several folds inferior to the inclined tool method in the literature. For comparison purposes, each piece was machined with the same tool and tool pass interval. The 16.0 mm diameter Carboloy tool MM16-0.630-R7.6-MD07 with a torus radius R = 5 .0 mm and insert radius r = 3.0 mm was used for the cutting tests. The tool pass interval was 8.0 mm. The spindle speed and feedrate were 1200 RPM and 70.0 mm/min respectively.

Before machining the workpieces, some setup was required. This setup is described in Appendix D. The machined surface was measured on a Mitotoyo BHN305 Coordinate measuring Machine, CMM. Surface deviations were extracted from the resulting surface measurements using the algorithm described in Appendix D.

The photographs of the inclined tool, principle axis and multi-point workpieces are shown in figures 7.27, 7.28 and 7.29. They show that the surfaces are practically identical in appearance. This is because the tool pass interval was the same for each workpiece and each technique produced relatively small scallops. Yet, the surface finish produced by each technique could be distinguished by touch; the inclined tool workpiece felt the roughest and the multi-point workpiece felt the smoothest. Also, it can be noted in Figure 7.29, that the lines between the tool passes are slightly more jagged for the multi-point workpiece.

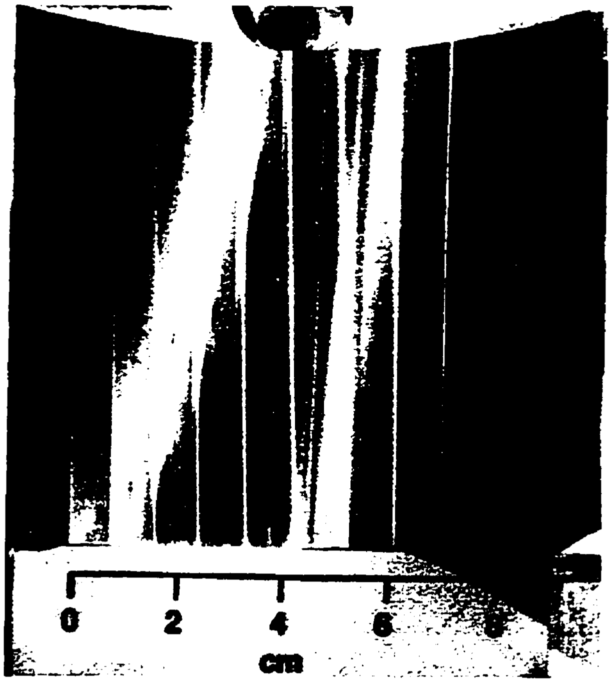


Figure 7.27 Test surface machined using inclined tool. R = 5.0 mm, r = 3.0 mm, tool pass interval = 8.0 mm, inclination angle = 6°

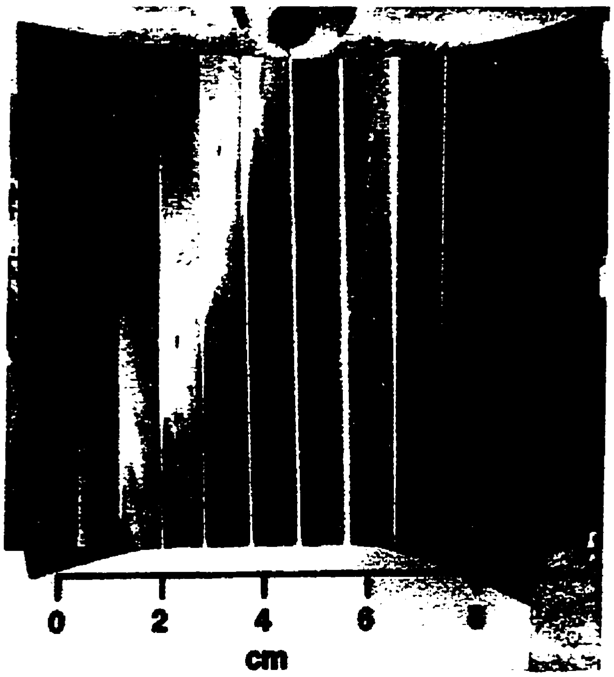


Figure 7.28 Test surface machined using principle axis method. R = 5.0 mm, r = 3.0 mm, tool pass interval = 8.0 mm

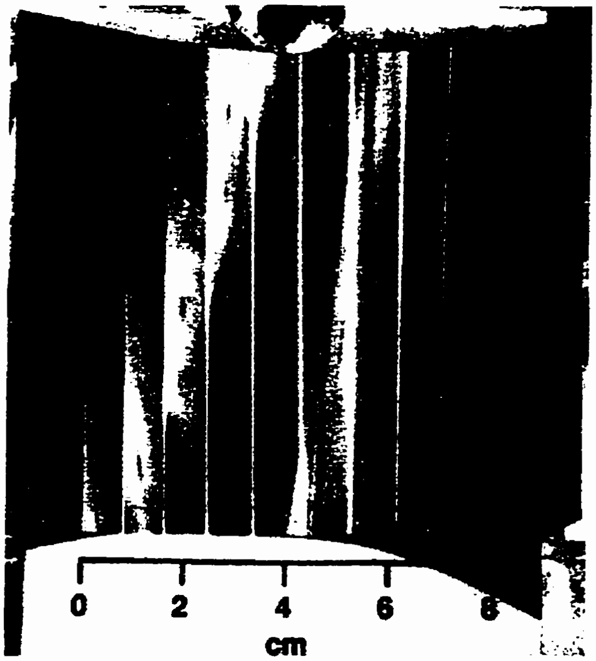


Figure 7.29 Test surface machined using multi-point method. R = 5.0 mm, r = 3.0 mm, tool pass interval = 8.0 mm, separation ratio = 0.8

The surface deviations for the three workpieces are shown in figures 7.30, 7.31 and 7.32 respectively. The thick lines represent the measured results and the thin lines represent the simulated results. The surface deviations are presented as four sections in the zy plane at x = -5.0, -30.0, -60.0 and -90.0 mm. Ideally the simulated and experimental results would match exactly. However, there are two distinct differences between the simulated and experimental results. First, the experimental deviations are slightly bow shaped whereas the simulated deviations are straight, and second, the height of the simulated and experimental scallops differ.

The bow shape, called form error, is largely due to errors locating the programmed coordinate system during workpiece setup. As noted in the discussion on experimental setup in Appendix D, this error is approximately ±12 μm in each of the x, y and z components of the vector describing the location of the programmed coordinate system. The propagation of this error onto the surface depends on the rotations of the A and C axes which results in the bow shape of the plots. Another source of error is due to the retrofitting of the Rambaudi, which included modification to the spindle and the addition of the tilt-rotary table. When the spindle was remounted, it had to be shimmed in order to align it with the z-axis. After shimming, the spindle was inclined by approximately 0.01° from the z-axis. The error in the table alignment was evaluated by measuring the flatness of the tilt-rotary table top when it was in the horizontal position. This was accomplished by taking dial indicator measurements on the tilt-rotary table top; they varied by ±12 μm over the entire surface.

If the scallops are considered without the form error, the agreement between the experimental and simulated results is excellent, with the experimental scallops being slightly smaller than the simulated scallops. These differences may be attributed to errors caused by the simulation technique, measurements and tool deflections. The discrete nature of the metal removal simulations may cause errors. Surface deviations are calculated every 0.1 mm. As a result features such as the sharp peaks of single point scallops may be missed and the resulting maximum scallop may be under-estimated. In addition, metal removal calculations are performed at discrete tool positions. In reality the tool removes metal as it moves along the tool path. This motion is not accounted for in the simulations. Therefore, more material is removed during actual machining than in simulated machining. Errors in the shape of the scallop also arise from the CMM measurements. As with the simulations, measurements were taken every 0.1 mm. Therefore the sharp peaks of the scallops may be missed. In addition, the coordinate systems used for the measurements, machining and the surface definition all vary slightly. The resulting measurements may contain some offset. Finally, the raw CMM data must be processed to extract surface deviation information. The algorithms used for this processing are described in Appendix D. The resulting processing may distort the shape of the actual scallops. Tool defections may be responsible for some of the differences between the experimental and simulated results. Tool deflections during machining were minimized by using a small depth of cut (0.5 mm) and a small feed/tooth (0.03 mm/tooth). The resulting cutting forces were on the order of 10-20 N which produced tool deflections of about 2-4 μm.

If the form error is ignored, the shape and size of the simulated surface deviations are close to the measured surface deviations of each machining method. For example, consider the inclined tool workpiece shown in Figure 7.30. The maximum scallop from the simulation was 133 μm; within 12% of the measured maximum of approximately 150 μm. The simulated and measured scallops both tended to have a parabolic shape which varied in the same proportions across the entire surface. This variation was due to the changing curvature of the surface. Regions of the surface with high curvature had larger scallops than regions with low curvature.

Figure 7.31 shows that simulated and measured results for the principle axis method are in good agreement. Surface deviations appear to be "U" shaped. The scallops are more evenly distributed across the surface because the principle axis method accounts for variations in curvature. Ignoring the form error, the maximum measured scallop was approximately 25 μm. The maximum simulated scallop was 22 mm, which is within 12% of the measured result.

The measured and simulated results for the multi-point workpiece also showed reasonable correlation. Both sets of surface deviations show the same distinctive shape of multi-point scallop; a large rounded scallop followed by a small sharp scallop.

The distribution of the scallop size is fairly even because multi-point machining accounts implicitly for curvature when the cutter contact points are selected. Information about the principle directions is incorporated into the tool position because the cutter contact points lie in the direction of maximum curvature. Therefore, when the tool is placed on these points the tool axis is approximately lined up with the direction of minimum curvature. The magnitude of the curvature effects the tool position because the angle between the surface normal vectors at cutter contact points influences the inclination angle of the tool.

If the form error is disregarded, the maximum measured scallop height for the multi-point workpiece was approximately 10 μm. The simulated maximum of 9.5 μm was within 5% of the measured result. Most importantly, both the measured and simulated traces demonstrate the superiority of the multi-point method. The multi-point scallop heights were approximately 15 and 2.5 times smaller than those produced by the inclined tool and principle axis methods.

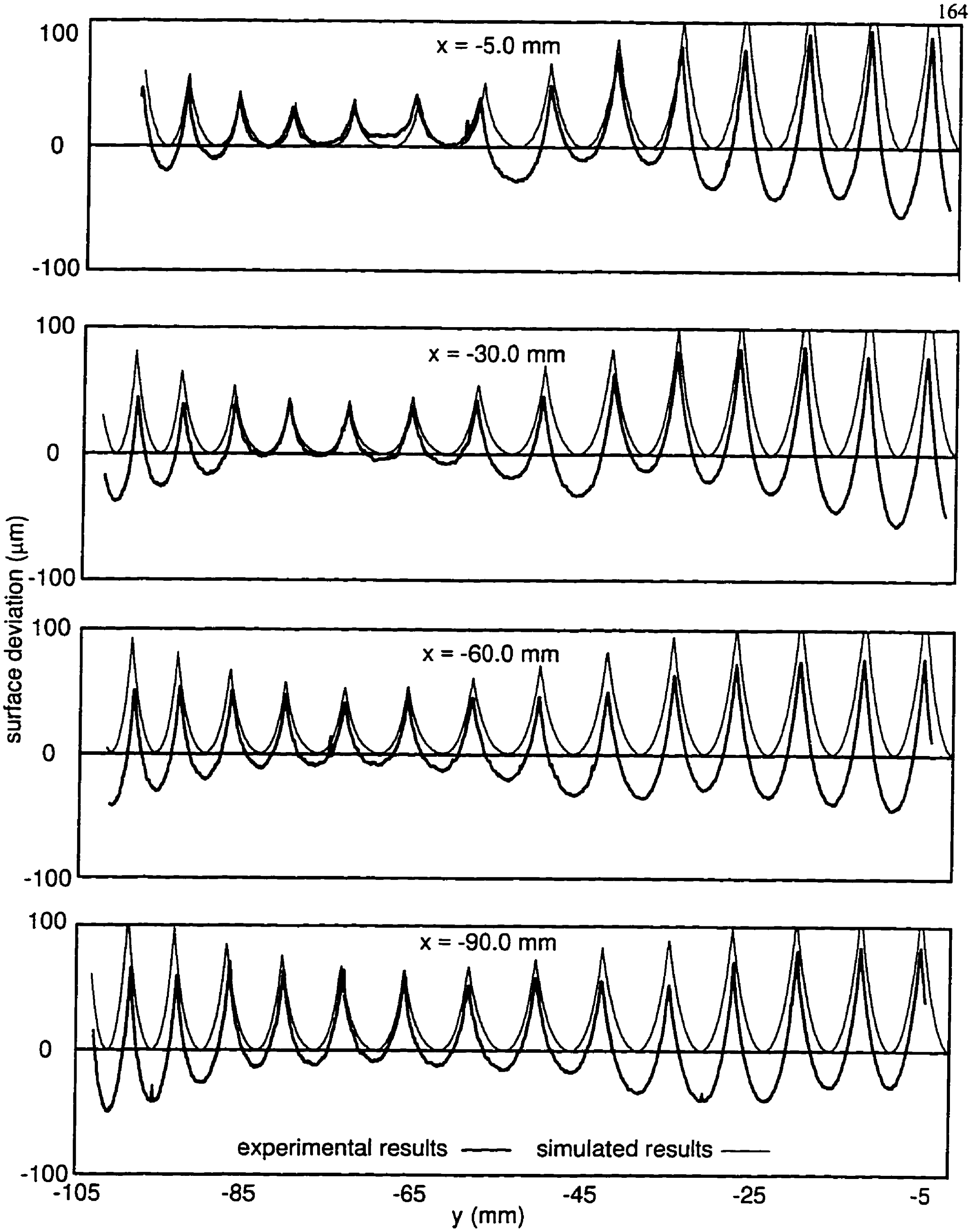


Figure 7.30 Comparison of experimental and simulated results for inclined tool method. R = 5.0 mm, r = 3.0 mm, tool pass interval = 8.0 mm, ϕ = 6°

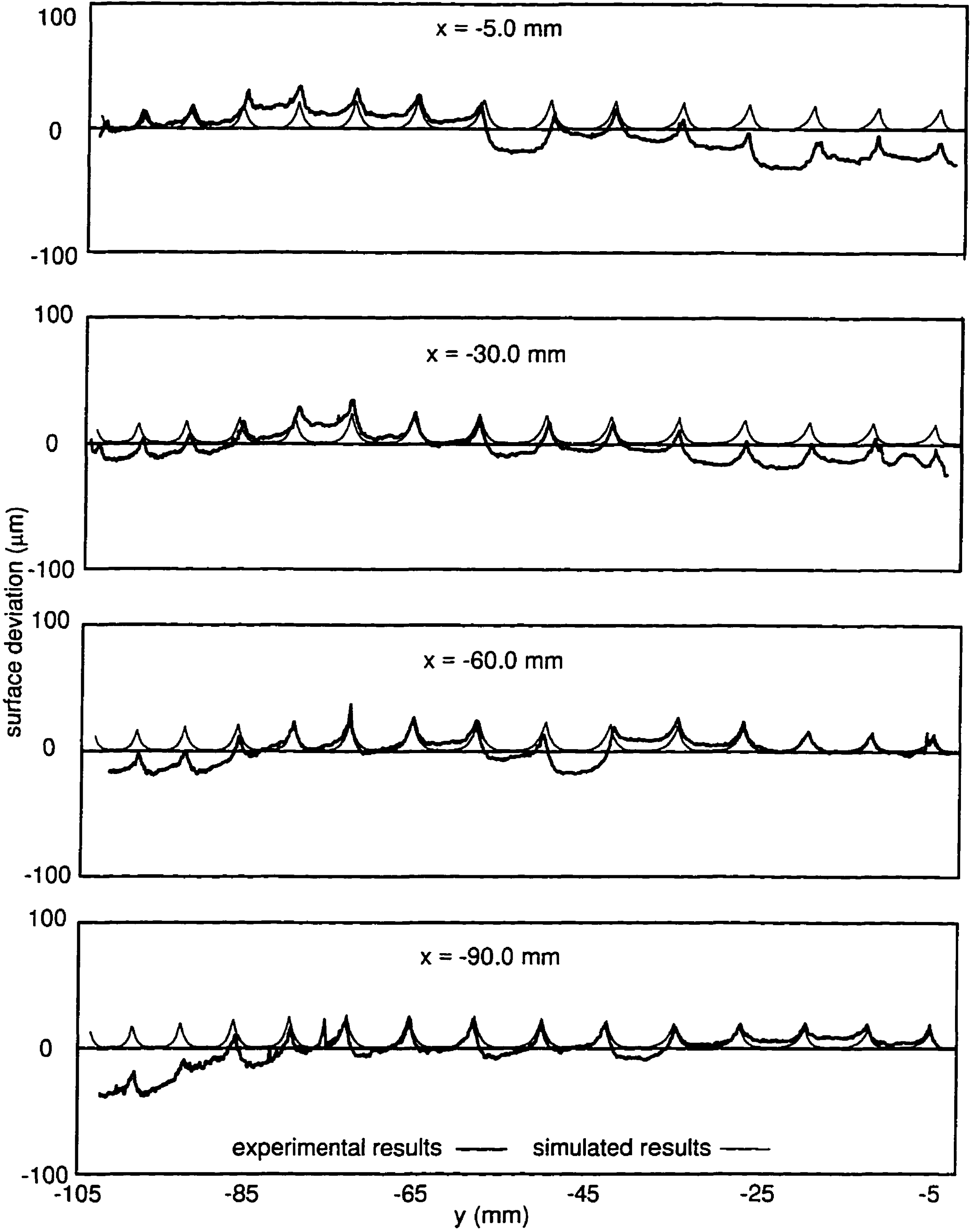


Figure 7.31 Comparison of experimental and simulated results for principle axis method. R = 5.0 mm, r = 3.0 mm, tool pass interval = 8.0 mm

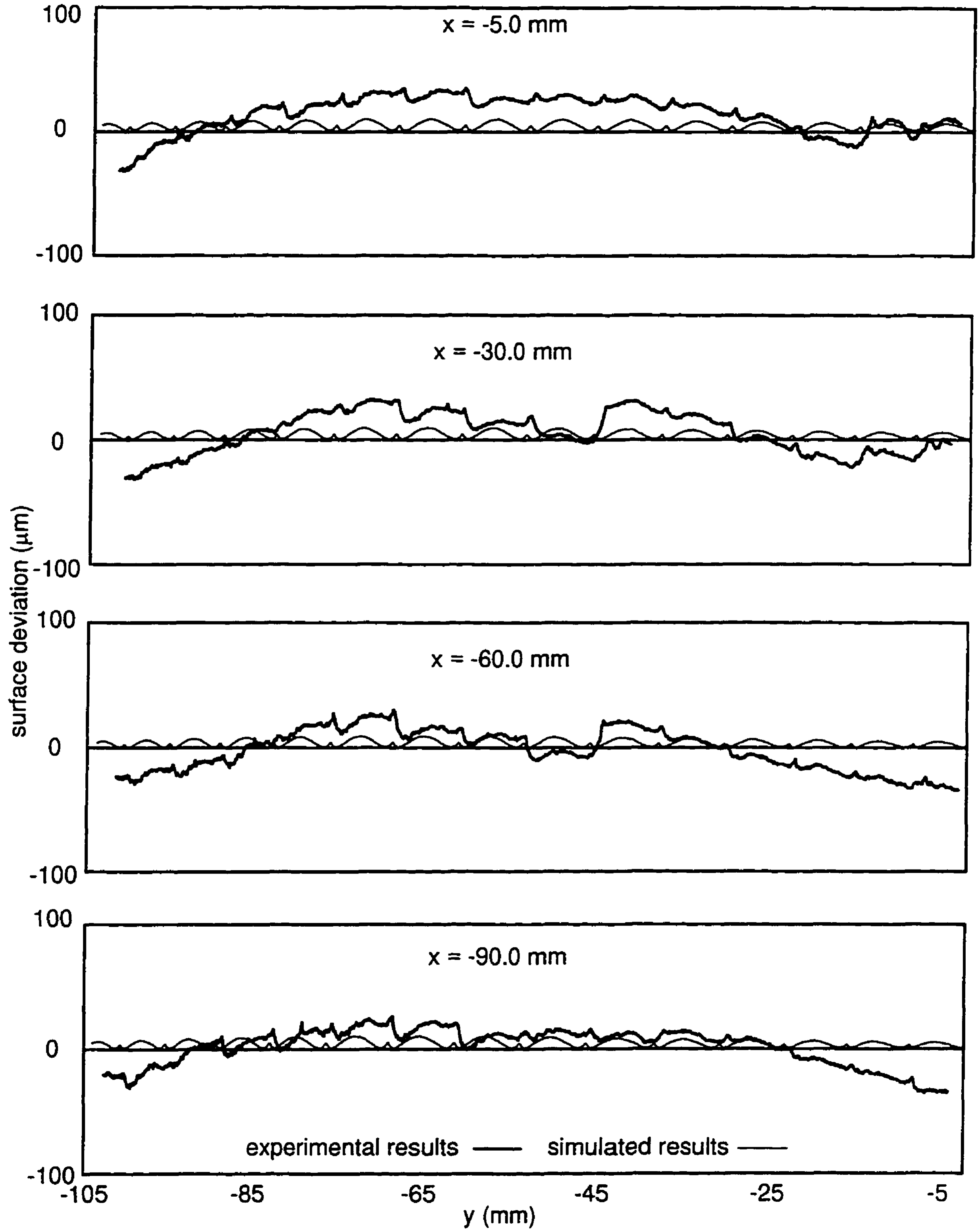


Figure 7.32 Comparison of experimental and simulated results for multi-point machining. R = 5.0 mm, r = 3.0 mm, tool pass interval = 8.0 mm, separation ratio = 0.8

7.4 Summary

Simulated and experimental results have been used to investigate the properties of multi-point machining and compare it with the currently used techniques. Multi-point scallops were found to be low and wide giving the machined surface a much smoother appearance than the competing techniques. The size of the scallops primarily depends on the size of the tool relative to the surface curvature. A big tool with small inserts will produce smaller scallops than a small tool with large inserts. The tool pass interval and separation distance also effect the scallop size. Finally, the feed direction affects the performance of multi-point machining. The best results occur when machining in the direction of minimum curvature. The worst results occur when machining in the direction of maximum curvature. In this latter case, multi-point machining behaves like single point machining.

The simulated results show that the scallop heights produced by multi-point machining are about 400, 25 and 2 times smaller than those produced by the ball nosed, inclined tool and principle axis methods respectively. Cutting tests were used to verify the simulated results.

Chapter 8

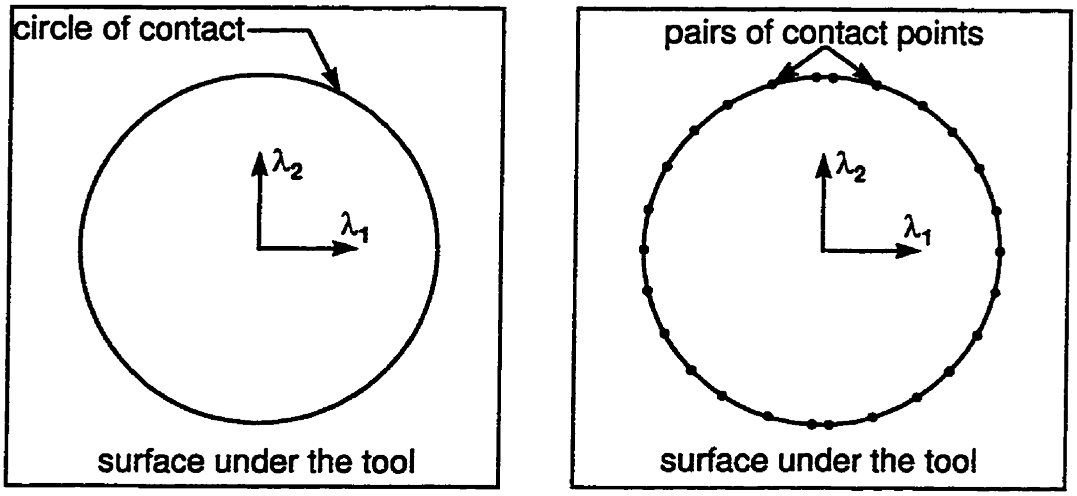
Conclusions and Recommendations

8.1 Conclusions

Sculptured surfaces are common in a wide variety of products such as automobiles, household appliances, water craft and aircraft components. Surface machining is often used directly or indirectly in the manufacture of these items. Unfortunately, surface machining is an extremely time consuming and costly process. A better surface machining process will lead to a reduction in product cost and an increase in the sophistication of product design. The goal of this work was to develop a 5-axis tool positioning strategy that would significantly reduce the time required for finish machining. The multi-point tool positioning strategy developed in this thesis has accomplished this goal.

One of the most important objectives of this work was to determine the nature of contact between a toroidal cutter and a surface. In other words, how many and in what configuration will cutter contact points be found? A multi-point tool positioning strategy could only be developed once these questions were answered. Therefore, a system of equations for modeling the contact between a tool and a design surface [64] was developed. They were used to study the nature of contact between the tool and the five quadratic surface forms: planer, parabolic, spherical, elliptic and hyperbolic. These surfaces can be used to approximate the topology of any surface under a tool. A study of the nature of multi-point contact with quadratic surfaces provided insight into the contact between a toroidal cutter and most surfaces.

This study concluded that two types of multi-point contact can exist between a quadratic surface and a toroidal cutter depending on the principle curvatures of the surface. The first type of contact occurs when the principle curvatures are equal as is the case of planar or spherical surfaces. For these surfaces a circle of contact can be achieved by offsetting the tool along the surface normal as illustrated by Figure 8.1(a). Planer surfaces have always been machined this way. However, prior to this work, spherical surfaces have not been machined in this fashion. A significant contribution to the field of surface machining was made by implementing multi-point machining for spherical surfaces [63]. Spherical surfaces can now be machined with the same finish as a planer surface in a fraction of the time required for conventional single point techniques.



(a) principle curvatures equal: *planar and spherical surfaces*

(b) principle curvatures different: *parabolic, elliptic and hyperbolic surfaces*

Figure 8.1 Configuration of multi-point contact for quadratic surfaces.

If the principle curvatures are different, as in the case of parabolic, elliptic and hyperbolic surfaces, a maximum of two points of contact can be achieved. These cutter contact points are arranged symmetrically about the direction of minimum curvature, λ_2, as shown in Figure 8.1(b). These important conclusions about the configurations of multi-point contact formed the basis of the multi-point tool positioning strategy.

After the investigation into the nature of multi-point contact, the next objective of this thesis was to develop a multi-point tool positioning strategy for quadratic surfaces. The resulting tool positioning strategy could be used to machine any quadratic surface without gouging. It was designed specifically for 2-point contact because this is by far the most likely occurrence of multi-point contact. One can still imagine rare instances when three or more point contact can occur. These rare cases were left for future studies. Multi point tool positions are calculated in three steps. In the first step, a pair of potential cutter contact points are located on the surface by assuming that the surface under the tool is one of the quadratic forms. Next, these points are used to calculate a multi-point tool position. If the potential cutter contact points have been located correctly, the tool position will result in tangential contact at both points. If the points have been located incorrectly, the tool will not be in tangential contact with the surface at either point. In the last step of the multi-point tool positioning strategy, the tool is moved so that tangential contact is achieved at at least one of the cutter contact points. This process ensures that, at worst, the resulting tool position will be comparable to a tool position calculated with a single point method.

The multi-point tool positioning strategy in this basic form has limited applicability; in most cases it will only produce multi-point contact for a quadratic surface. In reality, much more complex surfaces are used for the design of molds and dies. Therefore, two algorithms designated the tool approach and the surface approach, were developed to modify a multi-point tool position for this eventuality. These methods are not just gouge detection and correction algorithms, they can also detect and modify sub-optimal tool positions that are not gouging the surface. This is a major departure from the traditional gouge detection and correction approaches.

Both algorithms mentioned above use optimization codes to determine a multi-point tool position. The tool approach maintains tangential contact at the first cutter contact point while adjusting the tool position until the error at the second cutter contact point is minimized. The surface approach searches the surface for a pair of cutter contact points that will produce an error free tool position.

In terms of reliability, quality and speed the surface approach is far superior to the tool approach. The surface approach achieves multi-point contact 100% of the time verses 99% of the time for the tool approach. Although the scallop sizes achieved by both methods are about the same, the scallops produced by the tool approach tend to be slightly uneven. Most importantly, the computation associated with the surface approach take only a fraction of the time required by the tool approach. However, the tool approach has some advantages. It makes very few assumptions about the possible nature of contact between the tool and the surface. The surface approach assumes that the configuration of cutter contact points is similar to those produced by quadratic surfaces. This means that this approach may not work for rare and unusual surfaces. The tool approach only assumes that there are two points of contact. It does not assume anything about their locations. This approach may be more capable of dealing with unusual situations such as discontinuities or rapid changes in the curvature of a surface.

The last objective of this thesis was to evaluate multi-point machining. Simulations were used to examine the effect of the tool path parameters on the surface finish produced by multi-point machining. These parameters included: the tool geometry, tool pass interval, feed direction and separation distance. Once the effects of these parameters were understood, multi-point machining was compared with the ball nose, inclined tool and principle axis methods. This investigation was more demanding than those found in the literature because it compared the proposed technique to the best of the competing 5-axis techniques and not just the inferior 3-axis ball nose technique. Cutting tests were used to verify the simulated results.

Multi-point scallops were found to be low and wide giving the machined surface a much smoother appearance than the competing techniques. Among other factors, the size of the scallop depends on the size of the tool relative to the surface curvature. A big tool with small inserts will produce smaller scallops than a small tool with large inserts. The tool pass interval and separation distance also effects the scallop size. For best results, the separation distance should always be less than the tool pass interval. Finally, the feed direction also effects the performance of multi-point machining. The best feed direction is the direction of minimum curvature and the worst feed direction is the direction of maximum curvature. Most importantly, the results show that the scallop heights produced by multi-point machining are about 400, 25 and 2 times smaller than those produced by the ball nosed, inclined tool and principle axis methods respectively. Multi-point machining clearly offers the best performance of any tool positioning strategy.

8.2 Future Work

Multi-point machining has been shown to have great potential in the area of surface machining. Many small and large mold and die companies have shown interest in it. However, a great deal of work will be needed to make it a reality on the shop floor. The following paragraphs discuss this work.

Multi-point machining has only been applied to smooth low order concave surfaces consisting of a single patch. In reality, most parts consist of a multitude of different surface patches with convex as well as concave regions. For instance, a small turbine blade typically requires at least 50 different patches. Each patch may be as simple as a planar facet or as complicated as a NURBS surface.

Multi-point machining can be extended to convex surfaces by machining on the inner surface of a torus. This technique was used for machining spherical domes in Warkentin et al. [63] but has not yet been applied to other quadratics. A study of the contact between a torus and quadratic convex surfaces similar to the study in Chapter 4 should be undertaken. One can speculate that such a study will show that a maximum of two points of contact can be achieved with a convex surface. These points will lie in the direction of minimum curvature. If this premise is correct, then a multi-point tool positioning strategy similar to the one described in Chapter 5 can be developed. The tool adjustment algorithms of Chapter 6 will then require small modification for convex surface.

The applicability of multi-point machining to higher order surfaces must also be determined. Preliminary results show that the existing algorithm performs flawlessly for cubic Bèzier surfaces[64]. Nevertheless more case studies are needed for other commonly used surface patches such as third and fifth order B-splines and NURBS. Provided these surfaces can be reasonably approximated by quadratics, no difficulty is anticipated. If the quadratic approximation is unreasonable, the configuration and number of contact points may change. This could require the formulation of a higher order multi-point tool positioning strategy.

A surface may be designed with multiple surface patches, such as the previously mentioned turbine blade. In many instances, the boundaries between patches may be poorly matched or even discontinuous. Such composite surfaces may look fine on a graphics terminal but poorly designed from a machining perspective. Small irregularities in curvature can result in noticeable gouging of the final product. Multi-point machining has an advantage in this situation because the tool position is based on the information at more than one point. However, small irregularities may present problems for the tool adjustment algorithms because they assume the surface to be smooth. These algorithms may require enhancements to deal with non-smooth surfaces.

Multi-point machining must be fully integrated into the tool path planning process in order to achieve its full potential. The resulting tool paths should produce uniformly distributed scallops of a predefined size across the entire surface. These scallops were shown to depend on: tool dimensions, tool pass interval, separation distance, feed direction and surface curvature. These factors must all be accounted for in a multi-point tool positioning strategy.

The best way to incorporate feed direction into multi-point machining is to ensure that the tool is machining in the direction of minimum curvature at all times because this direction helps minimize scallop size. It will also reduce the number of points required for a tool pass because the tool will always be moving along the surface in the flattest direction. Thus, the direction of minimum curvature, λ_2, should point to the position of the next point as shown in Figure 8.2. For this tool path the cutter contact points, cc_1 and cc_2, and tool pass interval will lie in the direction of maximum curvature, λ_1. The tool pass interval and location of cc_2 will depend on the desired maximum scallop height and will vary along the tool pass. Complications to the proposed method will occur because tool passes based on the directions of curvature may diverge as the curvature of the surface changes. At the same time the tool pass calculation may require the tool passes to converge. A compromise solution between the directions of the tool passes and the tool pass interval calculation may be required. Further research is needed to obtain that optimum solution.

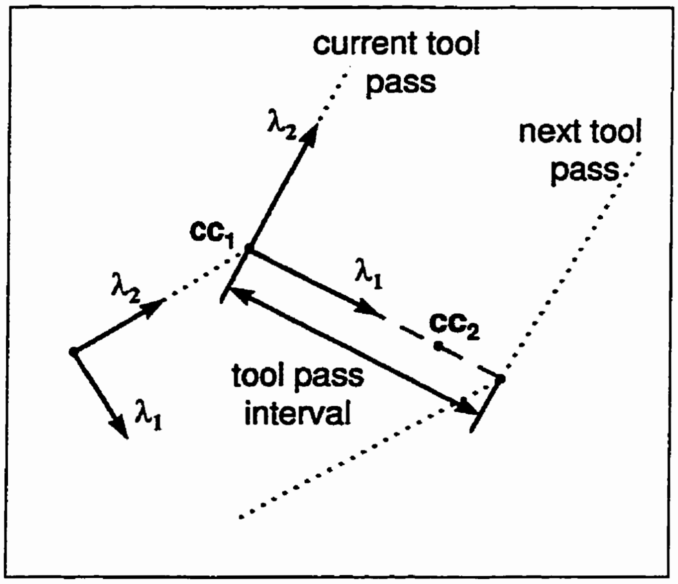


Figure 8.2 Proposed tool path direction.

REFERENCES

1. Altan, T., Lilly, B. W., Kuth J. P., Konig, W., Tonshoff, H. K., van Luttervet, C. A., Delft, T. U., Khairy, A. B., "Advanced techniques for die and mold manufacturing", *Annals of the CIRP*, 1993, 42(2), 707-716.

2. Anonymous, "Technology trends", *American Machinist,* 1994, 138(1), 16

3. Atherton, P. R., Earl, C., Fred, C., "A graphical simulation system for dynamic five-axis NC verification", Proc. Autofact, SME, Dearborn, MI, 1987, 2-12

4. Bedi, S., Ismail, F., Mahjoob, M. J. and Chen Y., "Toroidal versus ball nose and flat bottom end mills", International Journal of Advanced Manufacturing Technology, accepted Aug., 1996

5. Bobrow, J.E., "NC machine tool path generation from CSG part representations", *Computer-Aided Design*, 1985, 17(2), 69-76.

6. Cho, H. D., Jun, Y. T., and Yang M.Y., " Five-axis CNC milling for effective machining of sculptured surfaces", *International Journal of Production Research*, 1993, 31(11), 2559-2573.

7. Choi, B., Lee, C., Huang, J., and Jun, C., "Compound surface modeling and machining," *Computer-Aided Design*, 1988, 20(3), 127-136

8. Choi, B. K., and Jun, C. S., "Ball-end cutter interference avoidance in NC machining of sculptured surface", *Computer-Aided Design*, 1989, 21 (6), 371-378

9. Choi, B. K., Park, J. W., and June, C. S., "Cutter-location data optimization in 5-axis machining", Computer Aided Design, 1993, 25(6), 377-386

10. Chou, J. J., Yang D. C. H., "On the generation of coordinated motion of five axis CNC/CMM machines", *Journal of Engineering for Industry*, 1992, 114, 15-22.

11. Desrefani, J. D., "Mold and die making in transition", *Tooling and Production*, 1993, 4, 69-71.

12. Drysdale, R. L., Jerard, R. B., Schaudt, B., and Hauck, K., "Discrete simulation of NC machining", *Agorithmica*, 1889, 4(1), 33-60.

13. Farin, G., *Curves and Surfaces for Computer Aided Geometric Design*, 4th ed. 1997, Academic Press Inc., New York

14. Faux, I. D. and Pratt, M.J., *Computational Geometry for Design and Manufacture*, 1979, Ellis Horwood Limited, Chichester

15. Foley, J. D., van Dam, A., Feiner S. K., and Hughes, J. F., *Computer Graphics Principles and Practice*, 2nd ed. in C, 1996, Addison-Wesley Publishing Company Inc.

16. Fridshal, R., Cheng, K. P., Duncan, D., and Zucker, W., "Numerical control part program verification system", *Proc. Conf. CAD/CAM Technology in Mechanical Engineering*, 1982, MIT, Cambridge, MA

17. Groover, M. P., *Automation, Production Systems, and Computer-Integrated Manufacturing*, 1987, Prentice-Hall, New Jersey

18. Huang, Y. and Oliver, J. H., "NC milling error assessment and tool path correction", *Computer Graphics*, SIGGRAPH 94, Orlando, Florida, 24-29 July 1994 287-294

19. Huang, Y. and Oliver, J. H., "Non-constant parameter NC tool path generation on sculptured surfaces", *International Journal of Advanced Manufacturing Technology*, 1994, 9, 281-290.

20. Hui, K. C. "Solid sweeping in image space - application in NC simulation", *The Visual Computer*, 1994, 10(6), 306-316

21. Jerard, R. B., Drysdale, R. L., and Hauck, K., "Geometric simulation of numerical controlled machining", Proc. ASME Int. Computers in Engineering Conf. San Francisco, 1988, 2, 129-136

22. Jerard, R., Drysdale, R., Hauck, K., Schaudt, B. and Magewick, J., "Methods for detecting errors in numerically controlled machining of sculptured surface", *IEEE Computer Graphics and Applications,* 1989, 9(1), 26-39

23. Jerard, R. B., Hussaini, S. Z., Drysdale R. L., and Schaudt, B., " Approximate methods for simulation and verification of numerically controlled machining programs", *The Visual Computer*, 1989, 5, 329-348

24. Jensen, C.G. and Anderson, D. C., "Accurate tool placement and orientation for finished surface machining", *Journal of Design and Manufacture*, 3(4), 1993, 251-261.

25. Jensen, C.G., Anderson D. C., and Mullins S. H., "Scallop elimination based on precise 5-Axis tool placement, orientation, and step-over calculations", *Advances in Design Automation ASME*, 1993, 65(2), 535-544.

26. Kawabe, S. Kimura, F., Sata, T., "Generation of NC commands for sculptured surface machining from 3-coordinate measuring data," *CIRP Ann.*, 1981, 30(1), 369-372

27. Kim, C. B., Park, S. and Yang, M. Y., "Verification of NC tool path and manual and automatic editing of NC code", *International Journal of production Research*, 1995, 33(3), 659-673

28. Kiridena, V., and Ferreira, P.M., "Mapping the effects of positioning errors on the volumetric accuracy of five-axis machine tools", *International Journal of Machine Tools and Manufacture*, 1993, 33(3), 417-436.

29. Krieziz, G. A., "Algorithms for rational spline surface intersections", 1990, Ph.D thesis, Department of Ocean Engineering, MIT, U.S.A.

30. Kruth, J. P. and Klewais P., "Optimization and dynamic adaptation of cutter inclination during five-axis milling of sculptured surfaces ", *Annals of the CIRP*, 1994, 43(1), 443-448.

31. Lee, Y. S. and Chang, T. C. "Machine surface error analysis for 5-axis machining", *International Journal of Production Research*, 1996, 34(1), 111-135

32. Li, S.X. and Jerard, R.B., "5-axis machining of sculptured surfaces with a flat-end cutter", *Computer-Aided Design*, 1994, 26(3), 165-178.

33. Li, F., Wang, X.C., Ghosh, S. K., Kong, D. Z., "Gouge detection and tool position modification for five-axis NC machining of sculptured surfaces", *Journal of Materials Processing technology*, 1995, 48, 739-745

34. Li, F., Wang, X.C., Ghosh, S. K., Kong, D. Z., "Tool-path generation for machining sculptured surfaces", *Journal of Materials Processing Technology*, 1995, 48,. 811-816

35. Lin, R. S. and Koren, Y., "Efficient tool-path planning for machining free-form surfaces", *Journal of Engineering for Industry*, 1996, 118(1), 20-28

36. Loney, G., and Ozsoy, T, "NC machining of free form surfaces," *Computer-Aided Design*, 1987, 19(2), 85-90.

37. Mäntylä, M., *Solid Modeling*, 1988, Computer Science Press, Inc., Rockville, Maryland

38. Markot, R. P. and Magedson, R. L., "Solutions of tangential surface and curve intersections", *Computer-Aided Design*, 1989, 21(7), 421-429

39. Martin, R. R., "The geometry of the helical canal surface", *The Mathematics of Surface IV*, 1994, 17-32

40. McLellan, E., Young, G. M., Goult, R. J. and Pratt, M. J., "Interference checking in the 5-axis machining of parametric surfaces", Computer-Aided Surface Geometry and Design, The Mathematics of Surfaces IV, 1994, 439-453, Clarendon Press, Oxford

41. Menon, J. P., and Voelker, H. B., "Towards a comprehensive formulation of NC verification and as mathematical and computational problem", *Concurrent Engineering*, 1992, PED59. 147-164

42. Oliver, J. H., "Graphical verification of numerically controlled milling programs for sculptured surfaces", 1986, Ph.D., Michigan State University, U.S.A

43. Patrikalakis, N. M., "Interrogation of surface intersections", *Geometry Processing for Design and Manufacturing*, Barnhill, R. E., ed., SIAM, Philadelphia, 1992, 161-185

44. Paul, R. P., *Robot Manipulators: Mathematics, Programming and Control*, 1981, The MIT press, Cambridge MA, and London UK.

45. Pegna, J. and Wolter, F.-E., "Designing and mapping trimming curves on surfaces using orthogonal projection", ASME *Advances in Design Automation*,

Vol. 1, Computer Aided and Computational Design, 1990, DE-Vol. 23(1), 235-245

46. Press, W. H., Teukolsky S. A., Vetterling W.T., and Flannery B. P, *Numerical Recipes in C The Art of Scientific Computing*, 1992, 2nd Ed. Cambridge University Press

47. Rao, N., Bedi S., and Buchal, R.,, "Implementation of the principal-axis method for machining of complex surfaces", *International Journal of Advanced Manufacturing Technology,* 1996, 11, 249-257

48. Rao, N., Ismail, F. and Bedi, I, "Tool path planning for five-axis machining using the principle axis method", *International Journal of Machine Tool and Manufacture*, accepted 1997

49. Ruegg, A., "A generalized kinematics model for three- to five-axis milling machines and their implementation in CNC", *Annals of the CIRP*, 1992, 41(1), 547-550.

50. Sakamoto, S., and Inasaki, I., "Analysis of generating motion for Five-Axis machining Centers", *Transactions of NAMRI/SME*, 1993, 21, 287-293.

51. Sata, T., Kimura, F., Okada, N., and Hosaka, M.,, "A new method of NC interpolator for machining the sculptured surface," *CIRP Ann.*, 1981, 30(1), 369-372

52. Suh, Y. S. and Lee, K., "NC milling tool path generation for arbitrary pockets defined by sculptured surfaces", *Computer-Aided Design*, 1990, 22(5), 371-378

53. Suresh K. and Yang. D., "Constant scallop-height machining of free-form surfaces", *Journal of Engineering for Industry*, 1994, 116, 273-283

54. Takeuchi, Y. and Idemura, T., "5-axis control machining and grinding based on solid model", *Annals of the CIRP,* 1991, 40(1), 455-458

55. Van Hook, T. "Real-time shaded NC milling display", Computer Graphics, Proc. SIGGRAPH, 20(4), 15-20

56. Vickers, G. W. and Bedi, S., "The definition and manufacture of compound curvature surfaces using G-Surf', *Computers in Industry*, 1985, 6, 173-183.

57. Vickers, G.W., and Quan K.W., "Ball-mills versus end-mills for curved

surface machining", *Journal of Engineering for industry*, 1989, 111(22), 22-26.

58. Voelcker, H. B., and Hunt, W. A., "The Role of solid modeling in machine process modeling and NC verification", *Proc. 1981 SAE International Congress and Exposition*, 1981, Detroit, MI

59. Wang, W. P., and Wang, K. K., "Real time verification of multiaxis NC programs with raster graphics", IEEE Proc. Of 1986 International Conference on Robotics and Automation, San Francisco, April 1986, 166-171

60. Wang, W. P., and Wang, K. K., "Geometric modeling for swept volume of moving solids", IEEE Computer Graphics Applications, 1986, 6(12), 8-17

61. Wang, X. C., Ghosh, S. K., Li, Y. B., Wu, X. T., "Curvature catering- a new approach in manufacturing of sculptured surfaces (part 1. Theorem)", *Journal of Materials Processing technology*, 1993, 38, 159-176

62. Wang, X. C., Ghosh, S. K., Li, Y. B., Wu, X. T., "Curvature catering- a new approach in manufacturing of sculptured surfaces (part 1. methodology)", *Journal of Materials Processing technology*, 1993, 38, 177-194

63. Warkentin, A., Bedi, S., and Ismail, F., "5-Axis milling of spherical surfaces, *International Journal of Machine Tools and Manufacture*, 1995, 36(2), 229-243

64. Warkentin, A., Ismail, F., and Bedi, S., "Intersection approach to multi-point machining of sculptured surfaces", *Computer Aided Geometric Design*, accepted for publication July 2, 1997

65. Wysocki, D. A., Oliver, J. H., and Goodman, E. D., "Gouge detection algorithms for sculptured surface NC generation", *ASME Computer-Aided Design and Manufacture of Cutting and Forming Tools*, 1989, PED 40, 39-44

Appendix A

Differential Geometry of Surfaces

Differential geometry will be used to calculate the local properties of a surface such as the normal, arc length and curvature. The normal and curvature are key factors in the proper placement and orientation of a cutting tool on a sculptured surface. Curvature can also be used to classify the form of a surface. For a more detailed explanation of this topic see for example Farin [12] or Faux and Pratt [13].

A.1 Parametric and Cartesian Coordinates

A surface can be defined parametrically as:

$$\mathrm{S} = \mathrm{S}(\mathrm{u},\mathrm{v}) = \begin{bmatrix} \mathrm{x}(\mathrm{u},\mathrm{v}) \\ \mathrm{y}(\mathrm{u},\mathrm{v}) \\ \mathrm{z}(\mathrm{u},\mathrm{v}) \end{bmatrix} \quad ; \quad \mathrm{u},\mathrm{v} \in [\mathrm{a},\mathrm{b}] \subset \mathbf{R} \qquad \mathrm{A.1}$$

where the Cartesian coordinates of a surface point [x, y, z] are differentiable functions of the parameters u and v and [a, b] denotes a rectangle in the u, v plane as shown in Figure A.1.

To avoid potential problems of undefined normal vectors, it is assumed that the parametrization is regular [12]. In other words the surface will contain no cusps or ridges.

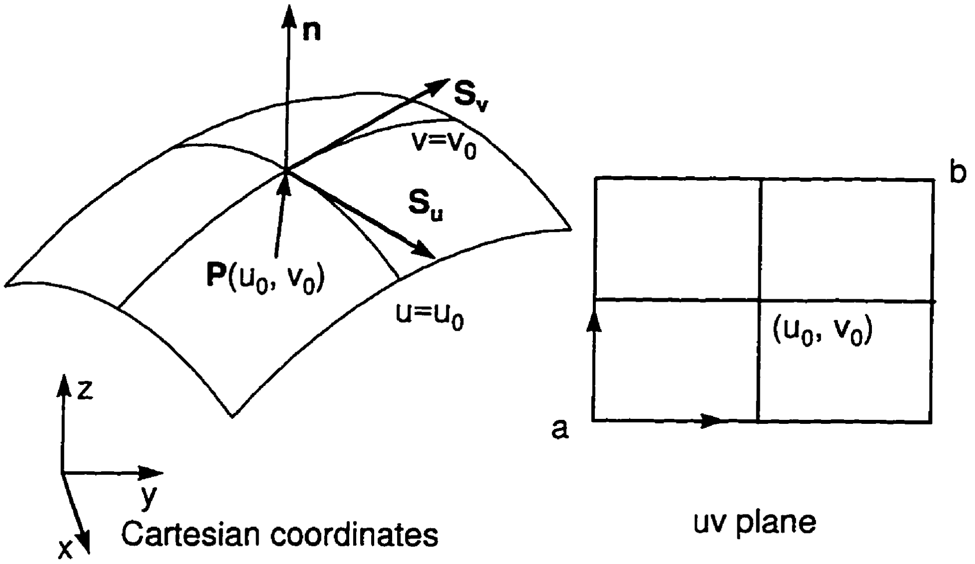


Figure A.1 Cartesian and parametric coordinates.

The partial derivatives S_u and S_v span the tangent plane at a point, **p**(u_0, v_0). Together with the surface normal, these vectors form a local coordinate system at the point.

$$S_u = S_u(u,v) = \begin{bmatrix} \frac{\partial x(u,v)}{\partial u} \\ \frac{\partial y(u,v)}{\partial u} \\ \frac{\partial z(u,v)}{\partial u} \end{bmatrix} \qquad S_v = S_v(u,v) = \begin{bmatrix} \frac{\partial x(u,v)}{\partial v} \\ \frac{\partial y(u,v)}{\partial v} \\ \frac{\partial z(u,v)}{\partial v} \end{bmatrix} \qquad \text{A.2}$$

$$\mathbf{n} = \frac{S_u \times S_u}{|S_u \times S_u|} \qquad \text{A.3}$$

A.2 Curvature

The curvature of a surface may be determined by considering the curvature of a curve formed by intersecting a plane containing the surface normal, **n**, at a point, **P**. This plane is known as an osculating plane. The normal curvature, κ_n, is the curvature of the resulting intersection curve as shown in Figure A.2. As this osculating plane rotates the value of the normal curvature changes. The directions that κ_n takes its maximum and minimum values are called the principle directions of curvature, λ_1 and λ_2. The values of the curvatures in these directions are principle curvatures κ_1 and κ_2. It can be shown that these directions are orthogonal. This discussion will consider the geometric properties of a space curve first, followed by the properties of curvature of the underlying surface itself.

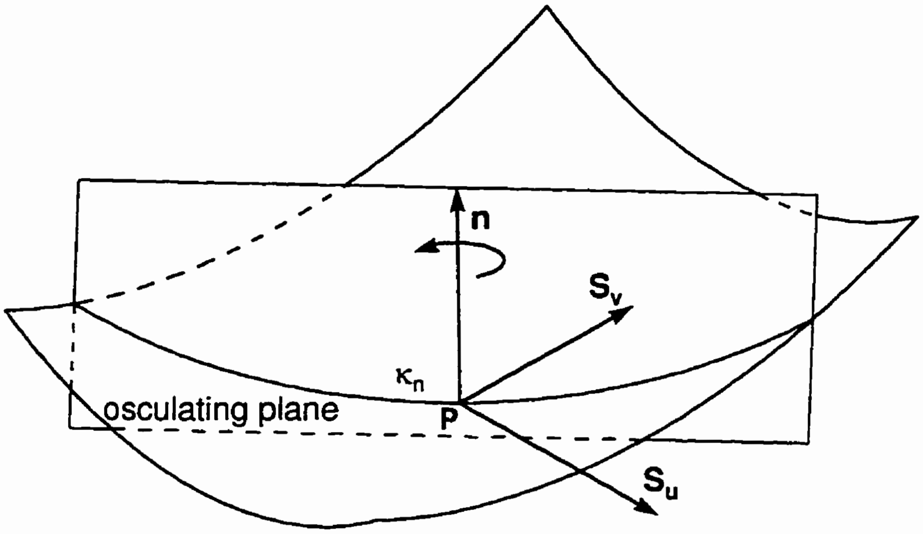


Figure A.2 Curve resulting from the intersection of a plane containing the surface normal.

A.3 Normal, Tangent and Curvature of a Space Curve

The parametric representation of a general space curve is given by equation A.4 and illustrated in Figure A.3.

$$\mathbf{r} = \mathbf{r}(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} \quad ; \quad t \in [a,b] \subset \mathbf{R} \qquad \text{A.4}$$

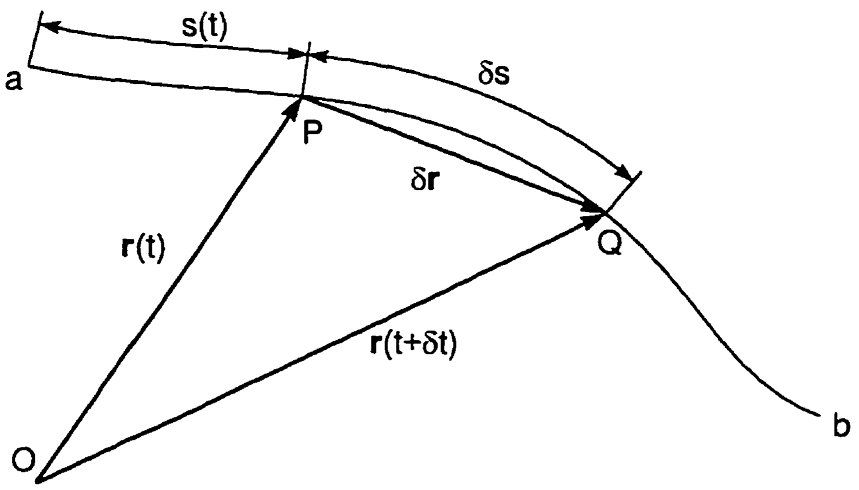


Figure A.3 Parametric curve in space.

The curve can also be expressed in terms of the distance, s, along the curve which is obtained by taking the line integral along the curve.

$$s = s(t) = \int_a^t \sqrt{\dot{\mathbf{r}}\cdot\dot{\mathbf{r}}}dt \qquad \text{A.6}$$

This type of parametrization is called arc length parametrization. The arc length parametrization is seldom used in calculation, but is important for the development of the theory. As δt→0 the cord $\delta\mathbf{r} = \mathbf{r}(t) - \mathbf{r}(t+\delta t)$ approaches the direction of the tangent vector **T** at P and δt→ δs. Therefore:

$$\frac{d\mathbf{r}}{ds} = \lim_{\delta s\to 0}\frac{\partial\mathbf{r}}{\partial s} = \mathbf{T} \qquad \text{A.7}$$

Applying the chain rule to equation A.7 and rearranging gives:

$$\dot{\mathbf{r}} = \frac{d\mathbf{r}}{dt} = \frac{ds}{dt}\mathbf{T} = \dot{s}\mathbf{T} \qquad \text{A.8}$$

The principle normal vector, **N**, is a unit vector perpendicular to **T** and can be obtained by differentiating equation A.8.

$$\ddot{\mathbf{r}} = \ddot{s}\mathbf{T} + \kappa\dot{s}^2\mathbf{N} \qquad \text{A.9}$$

where κ is the curvature of the curve. For example, a helix may be defined by,

$$\mathbf{r}(t)=\begin{bmatrix}a\cos(t) & a\sin(t) & bt\end{bmatrix}.$$

It follows that
$$\dot{\mathbf{r}}=\frac{d\mathbf{r}}{dt}=\begin{bmatrix}-a\sin(t) & a\cos(t) & b\end{bmatrix}$$

and
$$\dot{s}=\frac{ds}{dt}=\left|\dot{\mathbf{r}}\right|=\sqrt{a^2+b^2}\,.$$

Thus
$$\mathbf{T}=\frac{\dot{\mathbf{r}}}{\left|\dot{\mathbf{r}}\right|}=\frac{\begin{bmatrix}-a\sin(t) & a\cos(t) & b\end{bmatrix}}{\sqrt{a^2+b^2}},$$

and
$$\kappa\mathbf{N}=\frac{d\mathbf{T}}{ds}=\frac{d\mathbf{T}}{dt}\frac{dt}{ds}=\frac{\begin{bmatrix}-a\cos(t) & -a\sin(t) & 0\end{bmatrix}}{a^2+b^2}$$

where
$$\kappa=\left|\frac{d\mathbf{T}}{ds}\right|=\frac{a}{a^2+b^2}$$

and finally
$$\mathbf{N}=\begin{bmatrix}-\cos(t) & -\sin(t) & 0\end{bmatrix}.$$

A.4 Curvature of a Surface

A similar analysis can be used to find the curvature of a curve on a surface. A curve on the surface S(u, v,) may be represented parametrically by the equations **u** = u(t), v = v(t) which may be summarized as:

$$\mathbf{r}=\mathbf{r}(t)=\mathbf{S}\left[u(t),v(t)\right] \tag{A.10}$$

A tangent vector to this curve is found by taking the partial derivative with respect to the curve parameter, t.

$$\dot{\mathbf{S}}=\frac{\partial S}{\partial u}\frac{\partial u}{\partial t}+\frac{\partial S}{\partial v}\frac{\partial v}{\partial t}=\mathbf{A}\dot{\mathbf{u}} \tag{A.11}$$

where
$$\mathbf{A} = \begin{bmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial v} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{S}}{\partial u} \\ \frac{\partial \mathbf{S}}{\partial u} \end{bmatrix} \qquad \text{A.12}$$

The length of this tangent vector is given by:

$$\dot{s}^2 = \left|\dot{\mathbf{S}}\right| = \dot{\mathbf{S}}^T\dot{\mathbf{S}} = \dot{\mathbf{u}}^T\mathbf{A}^T\mathbf{A}\dot{\mathbf{u}} = \dot{\mathbf{u}}^T\mathbf{G}\dot{\mathbf{u}} \qquad \text{A.13}$$

where s is the arc length parameterization of the curve and

$$\mathbf{G} = \mathbf{A}^T\mathbf{A}\begin{bmatrix} \frac{\partial \mathbf{S}}{\partial u}\frac{\partial \mathbf{S}}{\partial u} & \frac{\partial \mathbf{S}}{\partial u}\frac{\partial \mathbf{S}}{\partial v} \\ \frac{\partial \mathbf{S}}{\partial v}\frac{\partial \mathbf{S}}{\partial u} & \frac{\partial \mathbf{S}}{\partial v}\frac{\partial \mathbf{S}}{\partial v} \end{bmatrix} = \begin{bmatrix} E & F \\ F & G \end{bmatrix} \qquad \text{A.14}$$

In classical differential geometry, the matrix **G** is known as the first fundamental form and will be of use for calculating curvatures. For the curve **r**(t) on the surface S(u, v) the curvature can be obtained by differentiating twice.

$$\ddot{\mathbf{S}} = \ddot{s}\mathbf{T} + \dot{s}^2\kappa\mathbf{N} = \frac{\partial^2\mathbf{S}}{\partial u^2}\dot{u}^2 + 2\frac{\partial^2\mathbf{S}}{\partial u\partial v}\dot{u}\dot{v} + \frac{\partial^2\mathbf{S}}{\partial v^2}\dot{v}^2 + \frac{\partial\mathbf{S}}{\partial u}\ddot{u} + \frac{\partial\mathbf{S}}{\partial v}\ddot{v} \qquad \text{A.15}$$

Note that the curve, S(u(t), v(t)) may be twisted so that the surface normal, **n**, may not be in the same direction as the normal to the curve, **N**. The surface normal component is obtained by:

$$\ddot{\mathbf{S}}\cdot\mathbf{n} = \dot{s}^2\kappa\mathbf{N}\cdot\mathbf{n} = \left(\frac{\partial^2\mathbf{S}}{\partial u^2}\dot{u}^2\right)\cdot\mathbf{n} + 2\left(\frac{\partial^2\mathbf{S}}{\partial u\partial v}\dot{u}\dot{v}\right)\cdot\mathbf{n} + \left(\frac{\partial^2\mathbf{S}}{\partial v^2}\dot{v}^2\right)\cdot\mathbf{n} \qquad \text{A.16}$$

Note that **n** is perpendicular to **T**, $\frac{\partial \mathbf{S}}{\partial u}$ and $\frac{\partial \mathbf{S}}{\partial u}$. In matrix notation the curvature is given by:

$$\ddot{\mathbf{S}}\cdot\mathbf{n} = \dot{s}^2\kappa\mathbf{N}\cdot\mathbf{n} = \dot{\mathbf{u}}\mathbf{D}\dot{\mathbf{u}} \qquad \text{A.17}$$

where

$$\mathbf{D}=\begin{bmatrix} \mathbf{n}\cdot\dfrac{\partial^2\mathbf{S}}{\partial u^2} & \mathbf{n}\cdot\dfrac{\partial^2\mathbf{S}}{\partial u\partial v} \\ \mathbf{n}\cdot\dfrac{\partial^2\mathbf{S}}{\partial u\partial v} & \mathbf{n}\cdot\dfrac{\partial^2\mathbf{S}}{\partial v^2} \end{bmatrix}=\begin{bmatrix} \mathbf{L} & \mathbf{M} \\ \mathbf{M} & \mathbf{N} \end{bmatrix} \tag{A.18}$$

The normal curvature can be obtained by rearranging A.17.

$$\kappa_n=\frac{\dot{\mathbf{u}}^T\mathbf{D}\dot{\mathbf{u}}}{\dot{s}^2}=\frac{\dot{\mathbf{u}}^T\mathbf{D}\dot{\mathbf{u}}}{\dot{\mathbf{u}}^T\mathbf{G}\dot{\mathbf{u}}} \tag{A.19}$$

As this osculating plane rotates the value of the normal curvature changes. The directions from which κ_n takes its maximum and minimum values are called the principle directions of curvature, λ_1 and λ_2. The corresponding principle curvatures κ_1 and κ_2 may be obtained by solving equation A.20 for κ_n.

$$(\mathbf{D}-\kappa_n\mathbf{G})\dot{\mathbf{u}}=\mathbf{0} \tag{A.20}$$

or

$$\begin{bmatrix} \mathbf{L}-\kappa_n\mathbf{E} & \mathbf{M}-\kappa_n\mathbf{F} \\ \mathbf{M}-\kappa_n\mathbf{F} & \mathbf{N}-\kappa_n\mathbf{G} \end{bmatrix}\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix}=\begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{A.21}$$

For a non-trivial solution, the extreme values of κ_n occur when:

$$\det\begin{bmatrix} \mathbf{L}-\kappa_n\mathbf{E} & \mathbf{M}-\kappa_n\mathbf{F} \\ \mathbf{M}-\kappa_n\mathbf{F} & \mathbf{N}-\kappa_n\mathbf{G} \end{bmatrix}=0 \tag{A.22}$$

The corresponding direction vectors can be found by substituting κ_1 and κ_2 into equation A.21 and solving for $[\dot{u},\dot{v}]^T$. These vectors correspond to the principle directions in parametric space.

Appendix B
The Virtual Rambaudi Simulator

B.1 Introduction

5-axis numerically controlled (NC) milling machines are used to machine automobile parts, turbine blades, molds and dies, etc. Programming these machines is a complex task requiring good 3D skills and a thorough understanding of the machine kinematics. The complexity of this task means that even experienced programmers can make mistakes. A programming error may result in the destruction of a workpiece, damage to the machine tool, or operator injury. Simulators allow the user to check that the CNC program is correct before incurring the expense and danger of running the program on a real machine. Machining through a simulated clamp is much cheaper than machining through a real one! The Virtual Rambaudi is a full featured 5-axis simulator created to debug NC programs written for the Mechanical Engineering department's new 5-axis milling machine at the University of Waterloo.

B.2 Features

This section describes the features of Virtual Rambaudi as illustrated in the following figure.

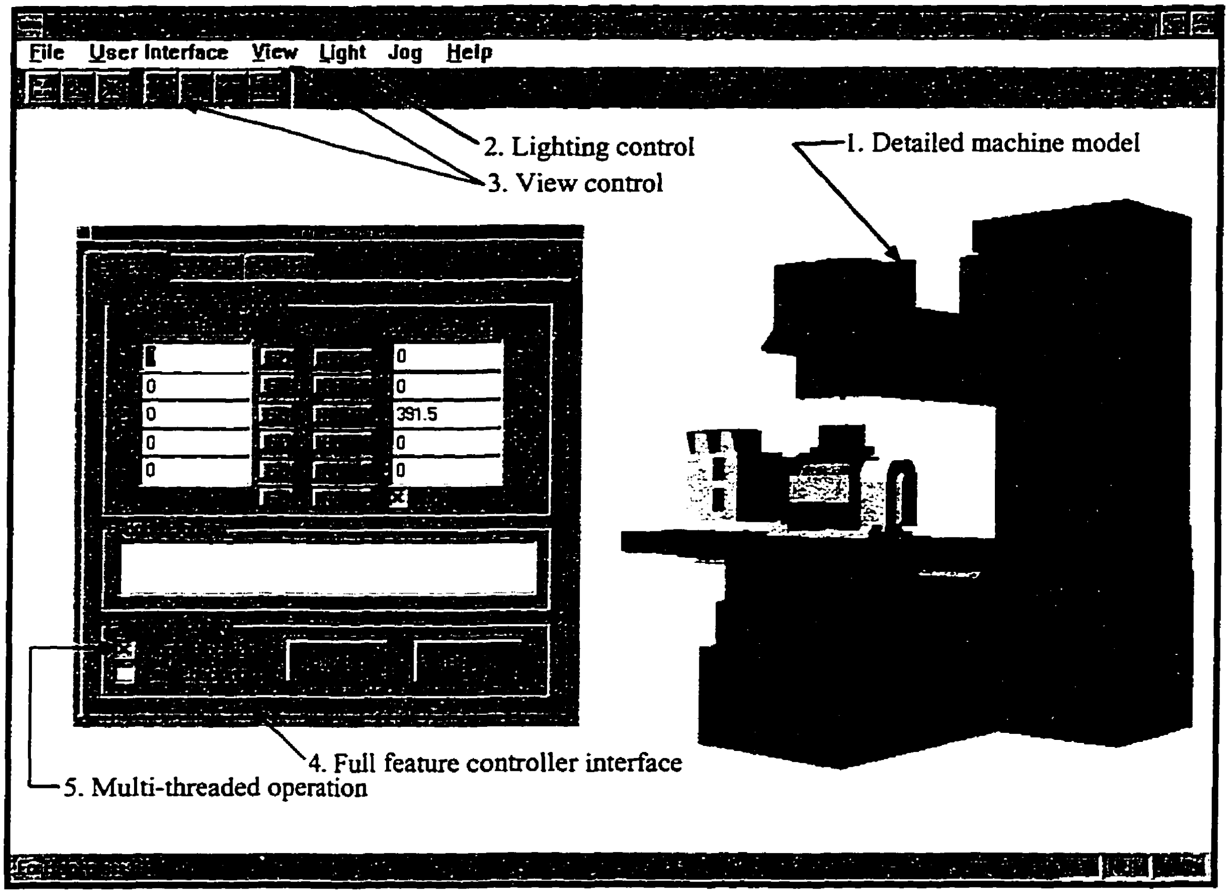

File User Interface View Light Jog Help
2. Lighting control
3. View control
1. Detailed machine model
0
0
0
0
0
0
0
391.5
0
0
4. Full feature controller interface
5. Multi-threaded operation

Figure B.1 Features of the Virtual Rambaudi

B.2.1 Machine model

A Rambaudi 5-axis NC milling machine has been retrofitted at the University of Waterloo. It is impossible to buy a commercial NC simulator for this machine. Most commercial simulators contain generic machine models that only approximate the target NC machine. This problem is aggravated by the fact that the Rambaudi is custom modified for research.

The machine model was produced in CorelCAD. CorelCAD is based on ACIS, which is an industry standard solid modeling C++ library. The model was exported from the CAD system in the ACIS solid modeling format. This file was read into a utility program, which converted the solid model file into a file containing a list of triangular polygons for each machine component. This file was read by the simulator and each part was stored in a separate display list.

Customization of the machine model was enhanced by making different tool, workpiece and fixture models. The user can select which machine conponent they want to use.

B.2.2 Lighting control

The appearance of the model can be altered by changing the lighting. The user has complete control over the color, intensity, and position of the lights. The position of the light can be displayed to make it easier to customize the illumination. Up to a maximum of two lights can be used to illuminate the scene. More lights would have provided only a small incremental improvement in illumination at the expense of application performance.

B.2.3 View control

View control allows the user to view any part of the 3D machine model. The view control uses a camera analogy. The user sees the machine by moving around the model and taking pictures from different directions. The user can interact with the model during a simulation by zooming, panning, or changing the view direction. Zoom buttons on the tool bar allow the user to zoom in or out. Panning buttons on the tool bar move the center of focus of the camera to the left, right , up or down. The magnification factors for zooming and panning can be altered by the user. The view direction is controlled by the view menu. The user can view the machine model from any direction by specifying the elevation and azimuth angle. Settings for front, side, top and other convenient views have also been provided.

B.2.4 Controller interface

The controller interface for the Virtual Rambaudi mimics that of a conventional NC controller. The controller interface consists of three overlapping pages on a standard windows property box. The operation of the simulator is controlled on the execution page. On this page the user can start and stop the simulator, manually position the machine, jog(incrementally move) each axis, and activate limit checking. This page also provides feedback on the current state of the machine and the current line of G-code being processed.

On the program page the user can load and view a G-code program. G-code is a programming language used to control the operation of NC machines. This code is parsed and used to control the simulator.

On the settings page the user can customize the machine for a particular simulation. The user can select different tools, workpieces, and fixtures. The user can also change the workpiece offsets. These offsets are used to relate the coordinate system of the workpiece to the coordinate system of the machine.

B.2.4 Multi-Threaded operation

An important feature of this simulation is its multi-threaded design. The machine simulation and the controller reside in separate threads. This means that the user can alter the controller, view, and lighting settings while the simulator is running a G-code program. Most importantly, the user can interrupt a G-code program by clicking on the "single block on" check box on the controller. If the simulator used only a single thread, the user would not be able to interact with the application while a G-code program was running.

B.3 Implementation

In the following sections the design of the Virtual Rambaudi software will be discussed.

B.3.1 The OpenGL skeleton application

The Virtual Rambaudi program was designed with modularity and code reuse in mind. One of the biggest challenges of this project was the integration of OpenGL into a Visual C++ application based on the Microsoft Foundation Class Library. Visual C++ provides a skeleton application for the programmer. The skeleton application usually consists of a AppMainframe, AppDocument, and a AppView class. Most OpenGL code is usually placed in the AppView class which is derived from the CView class. However, for the purposes of code reusability it was desirable to maintain a clean separation between the basic Visual C++ skeleton application and OpenGL. This was accomplished by creating a new class called the OpenGLView class which was derived from the CView class. This class contains most of the initialization code needed for an OpenGL application. To create a skeleton OpenGL application the user creates a skeleton Visual C++ application. The user then replaces all instances of "Cview" with "OpenGLView" in the AppView class. Effectively, the AppView class is now derived from the OpenGLView class. The only other change the programmer must make is to specify the initial size of the start up window in the OnCreate function in the AppMainFrame. The programmer then places his/her OpenGL code in the AppView class. Specifically: initialization code such as display list creation goes in the OnCreate function, drawing code goes in the OnDraw function; and viewing code goes in the OnSize function. The mainframe and document class typically contain no custom code in an OpenGL application.

B.3.2 Program Modules

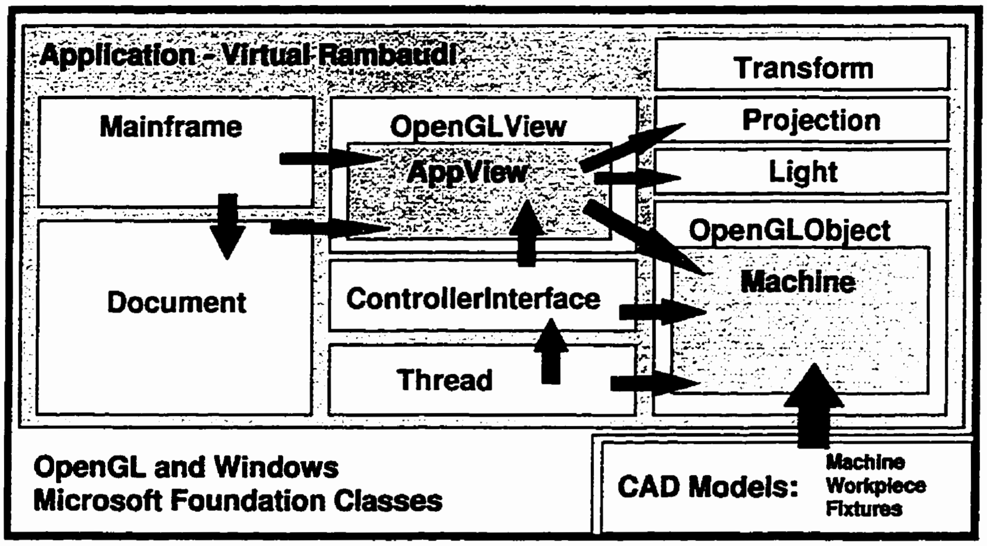


Figure B.2 Program modules.

The communication flow between the main program modules is shown in Figure B.2. The user interface and program execution are controlled by the AppView class, the ControllerInterface class and the thread function. The geometry, lighting, and view were handled by the Projection, Light, and Machine class. The Transform class contains utility functions for performing matrix and vector operations.

The AppView class contains the interfacing code for controlling the view and the lighting. The Controller Interface, Machine, Light and Projection classes are also initialized in this class. The view and lighting is controlled by calling the appropriate member functions of the Projection and Light classes followed by a forced screen refresh.

The ControllerInterface class consists of a modeless property sheet. This class is used to control the simulation, load and parse G-code, and load different machine components.

When the ControllerInterface class is instanced, a new thread is created to run G-code programs. The machine thread consists of an infinite loop. Each time through the infinite loop, the machine position is changed by executing a single line of G-code. The thread also checks to see if the "single block on" has been activated, and checks to see if the thread should be terminated. When "single block on" is off, the simulation will continue until all the G-code has been executed. If "single block on" is active only a single line of G-code will be executed. When the ControllerInterface is destroyed a message is sent to the thread to cease execution.

The Projection class contains the code needed to set and modify the view. The frustum is set using the gluPerspective and the gluLookAt functions. Different view directions are accomplished by moving the eye around the model center at a constant distance. Panning is implemented by moving the eye and model center location in the plane of the screen. Zooming is accomplished by scaling the object.

The Light class contains all the code needed to set and modify the location and color characteristics of a light. This class can be instanced repeatedly to produce more lights.

The Machine class is derived from the OpenGLObject class. The OpenGLObject class contains the code needed to initialize and set the material properties of an object. The geometry of the simulated machine is set by the Machine class. When the Machine class is instanced files containing tessellated machine conponent descriptions are read into display lists. Each machine conponent is stored in a separate display list. The machine component display lists are modified to simulate the motion of the NC machine.

Appendix C

Summary of Simulated Results

In this Appendix the data generated from the simulations is presented. This appendix is divided into sections on: 3-axis ball nosed machining, 5-axis inclined tool machining, 5-axis principle axis method, 5-axis multi-point method and comparison of results. In this appendix a number of symbols will be used in the headings of various tables. These symbols are defined as

χ tool pass interval

w cutter contact separation distance

R torus radius of tool

r insert radius of toroidal tool or radius of ball nosed end mill

ϕ inclination angle for 5-axis incline tool machining

C.1 3-Simulations of 3-axis Machining

Tool pass interval (mm)	Max scallop height (μm)	Max gouge (μm)	Volume of scallop material (mm^3)	Volume of gouged material (mm^3)
1	30.5	0.0	93.5	1.1
2	129.3	0.0	374.7	1.1
3	289.5	0.0	853.2	0.0
4	518.8	0.0	1541.4	0.0
5	850.4	0.0	2427.2	0.0
6	1353.8	0.0	3591.2	0.0
7	2082.0	0.0	5113.8	0.0
8	2705.6	0.0	6995.3	0.0

Table C.1 3-axis machining with a ball nosed end mill. r = 8.0 mm

C.2 Simulations of 5-axis Machining Using an Inclined Tool

Incline angle, ϕ (degree)	Max gouge (μm)	Volume of gouged material (mm^3)
1	222.6	469.7
2	144.0	150.8
3	80.7	31.2
4	36.4	3.3
5	6.3	0.1
6	0.1	0.0
7	0.1	0.0

Table C.2 Effect of inclination angle on 5-axis machining with an inclined toroidal cutter. r = 3.0 mm, R = 5.0 mm, 5mm cross-step

Tool pass interval (mm)	Max scallop height (μm)	Max gouge (μm)	Volume of scallop material (mm^3)	Volume of gouged material (mm^3)
1	2.1	0.3	4.8	0.0
2	7.5	0.4	14.4	0.0
3	15.3	0.4	30.4	0.0
4	27.5	0.1	53.5	0.0
5	49.4	0.5	83.8	0.0
6	67.5	0.2	126.5	0.0
7	100.5	0.1	176.1	0.0
8	132.7	0.1	244.5	0.0
9	162.2	0.2	325.4	0.0
10	282.1	0.2	448.6	0.0
11	392.3	0.1	601.0	0.0
12	461.6	0.1	803.1	0.0
13	956.3	0.0	1116.7	0.0
14	1430.3	0.1	1564.2	0.0

Table C.3 Effect of tool pass interval on 5-axis machining with an inclined toroidal cutter. r = 3.0 mm, R = 5.0 mm, ϕ = 6°

C.3 Simulations of 5-axis Machining Using Principle Axis Method, PAM

Tool pass interval (mm)	Max scallop height (μm)	Max gouge (μm)	Volume of scallop material (mm^3)	Volume of gouged material (mm^3)
1	0.4	0.9	1.1	0.0
2	0.6	0.9	1.6	0.0
3	0.7	0.9	2.1	0.0
4	1.3	0.3	3.2	0.0
5	2.7	0.9	5.9	0.0
6	6.0	0.8	11.1	0.0
7	12.9	0.3	20.8	0.0
8	21.9	0.9	27.3	0.0
9	37.5	0.7	62.3	0.0
10	84.3	1.0	105.3	0.0

Table C.4 Effect of tool pass interval on 5-axis machining using principle axis method. r = 3.0 mm, R = 5.0 mm

C.4 Simulations of 5-axis Machining Using Multi-point Machining, MPM

C.4.1 Effect of Tool Geometry

tool pass interval (mm)	Max scallop height (μm)	Max gouge (μm)	Volume of scallop material (mm^3)	Volume of gouged material (mm^3)
1	0.3	1.0	0.9	0.0
2	0.5	0.5	1.8	0.0
3	1.3	0.1	5.2	0.0
4	3.9	0.1	16.0	0.0
5	4.7	0.1	46.2	0.0

Table C.5 r = 5.0 mm, R = 3.0 mm, w = tool pass interval

Tool Pass interval (mm)	Max scallop height (μm)	Max gouge (μm)	Volume of scallop material (mm^3)	Volume of gouged material (mm^3)
1	0.4	0.7	1.1	0.0
2	0.6	1.0	1.6	0.0
3	1.4	0.8	2.9	0.0
4	1.6	0.1	6.4	0.0
5	3.5	1.5	14.3	0.0
6	7.5	0.1	30.1	0.0
7	14.6	0.1	59.7	0.0
8	28.3	0.1	115.4	0.0
9	56.6	0.0	228.7	0.0
10	170.8	0.0	532.9	0.0

Table C.6 r = 3.0 mm, R = 5.0 mm, w = tool pass interval

Tool Pass interval (mm)	Max scallop height (μm)	Max gouge (μm)	Volume of scallop material (mm^3)	Volume of gouged material (mm^3)
1	0.9	1.0	1.6	0.0
2	1.3	1.5	2.6	0.0
3	1.5	1.2	3.9	0.0
4	2.2	1.2	6.1	0.0
5	2.9	1.2	10.3	0.0
6	4.7	0.4	18.0	0.0
7	8.0	0.1	31.3	0.0
8	13.4	0.1	53.7	0.0
9	22.6	0.3	88.6	0.0
10	36.4	0.7	145.5	0.0
11	59.6	0.0	236.1	0.0
12	95.8	0.0	385.4	0.0
13	172.4	0.1	660.1	0.0

Table C.7 r = 1.0 mm, R = 7.0 mm, w = tool pass interval.

Tool pass interval (mm)	Max scallop height (μm)	Max gouge (μm)	Volume of scallop material (mm^3)	Volume of gouged material (mm^3)
1	0.4	0.5	1.2	0.0
2	1.0	0.1	2.2	0.0
3	1.6	0.1	6.0	0.0
4	4.4	0.1	18.0	0.0
5	12.9	0.1	52.4	0.0
6	61.3	2.1	190.7	0.0
7	95.7	0.1	407.7	0.0
8	217.8	0.1	642.2	0.0
9	388.4	1.3	977.2	0.0
10	1574.2	0.6	1735.7	0.0

Table C.8 r = 3.0 mm, R = 3.0 mm, w = tool pass interval.

Tool pass interval (mm)	Max scallop height (μm)	Max gouge (μm)	Volume of scallop material (mm^3)	Volume of gouged material (mm^3)
1	0.4	1.2	0.9	0.0
2	0.5	1.3	1.3	0.0
3	0.7	0.8	2.1	0.0
4	1.1	0.3	3.9	0.0
5	2.0	0.9	7.6	0.0
6	3.7	0.4	14.7	0.0
7	6.6	0.5	26.9	0.0
8	11.4	0.1	47.2	0.0
9	19.7	0.4	78.8	0.0
10	31.9	0.8	129.9	0.0

Table C.9 r = 3.0 mm, R = 7.0 mm, w = tool pass interval.

Tool pass interval (mm)	Max scallop height (μm)	Max gouge (μm)	Volume of scallop material (mm^3)	Volume of gouged material (mm^3)
1	0.3	1.2	0.8	0.0
2	0.4	1.2	1.2	0.0
3	0.7	0.5	2.4	0.0
4	1.7	0.2	5.6	0.0
5	3.1	0.1	12.8	0.0
6	6.6	0.3	27.2	0.0
7	13.1	0.3	53.9	0.0
8	24.7	0.1	103.7	0.0
9	48.3	0.1	197.8	0.0
10	105.9	0.1	428	0.0

Table C.10 r = 5.0 mm, R = 5.0 mm, w = tool pass interval.

Tool pass interval (mm)	Max scallop height (μm)	Max gouge (μm)	Volume of scallop material (mm^3)	Volume of gouged material (mm^3)
1	0.4	1.4	0.9	0.0
2	0.5	1.3	1.3	0.0
3	0.7	0.6	2.1	0.0
4	1.1	0.8	3.9	0.0
5	2	0.7	7.6	0.0
6	3.7	0.3	14.7	0.0
7	6.6	0.3	26.9	0.0
8	11.4	0.2	47.2	0.0
9	19.7	0.4	78.8	0.0
10	31.9	0.6	129.9	0.0

Table C.11 r = 7.0 mm, R = 5.0 mm, w = tool pass interval.

C.4.2 Effect of Cutter Contact Separation Distance

Cutter contact separation (%tool pass interval)	Max scallop height (μm)	Max gouge (μm)	Volume of scallop material (mm^3)	Volume of gouged material (mm^3)
40.0	0.8	0.8	1.6	0.0
60.0	0.9	1.0	1.5	0.0
80.0	0.5	0.9	1.5	0.0
100.0	0.6	0.9	1.6	0.0
120.0	0.6	0.7	1.7	0.0
140.0	0.6	0.5	1.8	0.0
160.0	0.6	1.2	1.9	0.0
180.0	0.8	0.8	2.4	0.0

Table C.12 Effect of cutter contact separation distance on multi-point machining.
r = 3.0 mm, R = 5.0 mm, tool pass interval = 2.0 m

Cutter contact separation (%tool pass interval)	Max scallop height (μm)	Max gouge (μm)	Volume of scallop material (mm^3)	Volume of gouged material (mm^3)
20	3.8	0.6	5.8	0.0
40	2.5	0.5	4.4	0.0
60	1.6	0.4	3.6	0.0
80	1.7	0.3	5.7	0.0
100	3.5	0.4	14.3	0.0
120	7.5	0.6	26.5	0.0
140	15.0	0.2	36.4	0.0
160	21.0	0.1	42.3	0.0
180	36.4	0.1	88.3	0.0

Table C.13 Effect of cutter contact separation distance on multi-point machining. r = 3.0 mm, R = 5.0 mm, tool pass interval = 5.0 mm

Cutter contact separation (%tool pass interval)	Max scallop height (μm)	Max gouge (μm)	Volume of scallop material (mm^3)	Volume of gouged material (mm^3)
40.0	55.6	0.5	26.0	0.0
60.0	10.5	0.8	18.5	0.0
80.0	9.7	0.1	34.4	0.0
100.0	28.3	0.1	116.4	0.0
120.0	98.2	0.1	359.8	0.0
140.0	250.9	0.1	991.1	0.0
160.0	332.7	0.1	1275.4	0.0
180.0	409.3	0.1	1497.1	0.0

Table C.14 Effect of cutter contact separation distance on multi-point machining. r = 3.0 mm, R = 5.0 mm, tool pass interval = 8.0 mm

C.4.3 Feed direction on Multi-Point Machining

Angle to direction of minimum curvature (degree)	Max scallop height (μm)	Max gouge (μm)	Volume of scallop material (mm^3)	Volume of gouged material (mm^3)
-70	56.9	782.72	23.5	1439.9
-60	42.0	54.5	81.6	18.9
-50	23.4	0.1	52.6	0.0
-40	17.7	0.1	31.0	0.0
-30	12.4	0.4	21.4	0.0
-20	9.2	0.1	16.5	0.0
-10	5.5	0.2	14.1	0.0
0	3.0	0.3	13.5	0.0
10	3.5	0.0	14.3	0.0
20	5.1	0.1	15.8	0.0
30	13.2	0.1	21.5	0.0
40	25.3	0.1	32.7	0.0
50	42.8	0.1	55.5	0.0
60	51.3	13.8	89.4	3.4
70	54.6	43.9	475.1	443.9

Table C.15 Effect of feed direction on multi-point machining.
r = 3.0 mm, R = 5.0 mm, tool pass interval = 5.0 mm
cutter contact separation distance = 5.0 mm

C.5 Comparison of machining methods

Tool pass interval (mm)	3-axis method	5-axis methods		
	ball	Inclined tool (6°)	Principle axes method	Multi-point machining
1	30.5	2.1	0.4	0.4
2	129.3	7.5	0.6	0.6
3	289.5	15.3	0.7	0.9
4	518.8	27.5	1.3	1.2
5	850.4	49.4	2.7	1.6
6	1353.8	67.5	6.0	3.0
7	2082.0	100.5	12.9	5.3
8	2705.6	132.7	21.9	9.5
9	-	162.2	37.5	16.2
10	-	282.1	84.3	27.0

Table C.16 Comparison of scallop height(μm) for different techniques. torus dimensions: R = 5 mm, r = 3 mm, ball dimensions: r = 8 mm.

Tool pass interval (mm)	3-axis method	5-axis methods		
	ball	Inclined tool (6°)	Principle axes method	Multi-point machining
1	93.5	4.8	1.1	1.0
2	374.7	14.4	1.6	1.5
3	853.2	30.4	2.1	2.1
4	1541.4	53.5	3.2	3.2
5	2427.2	83.8	5.9	5.6
6	3591.2	126.5	11.1	10.4
7	5113.8	176.1	20.8	18.7
8	6995.3	244.5	27.3	33.0
9	-	325.4	62.3	58.6
10	-	448.6	105.3	99.9

Table C.17 Comparison of scallop volume(mm^3) for different techniques. torus dimensions: R = 5 mm, r = 3 mm, ball dimensions: r = 8 mm.

Appendix D

Experimental Procedures

Before machining the workpieces, some setup of the CNC machine and workpiece was required. This setup consisted primarily of determining the coordinate systems used by the NC controller and post processor. These coordinate systems are shown in Figure D.1. Specifically, the machine, programmed and workpiece coordinates labeled in the figure as $\mathbf{C_m}$, $\mathbf{C_p}$ and $\mathbf{C_w}$ must be located.

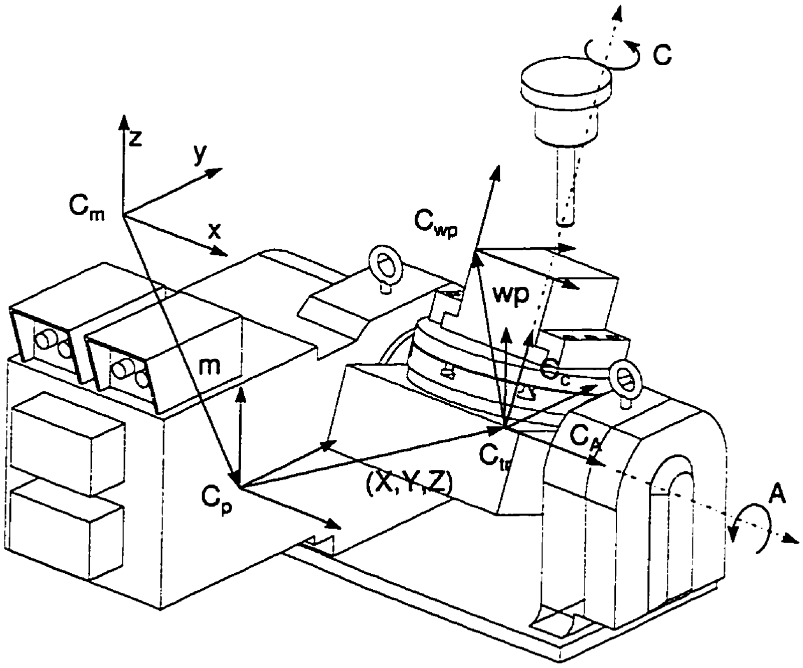


Figure D.1 Coordinate system of Ranboudi.

The machine coordinate system is the most positive position in the machine's work volume. It is the frame of reference used by the controller for joint motion. The machine coordinate system is located by homing all the axes of the machine. In other words each axis is moved to its most positive position. A limit switch attached to the frame of the machine informs the controller when this point has been reached. The home position is determined by the hardware of the system and is handled entirely by the CNC controller.

The controller assumes that all motion commands are given in terms of the programmed coordinate system. In 5-axis machining it is most convenient to place this coordinate system at the intersection point of the rotary axes. This point is usually called the rotation point. In other words a command to move to location (0, 0, 0) would place the tool at the rotation point as illustrated in Figure D.2. Note that a command to move to this position would damage the machine. The location of the programmed coordinate system is determined by specifying a machine offset vector, **m**, relative to the machine coordinate system.

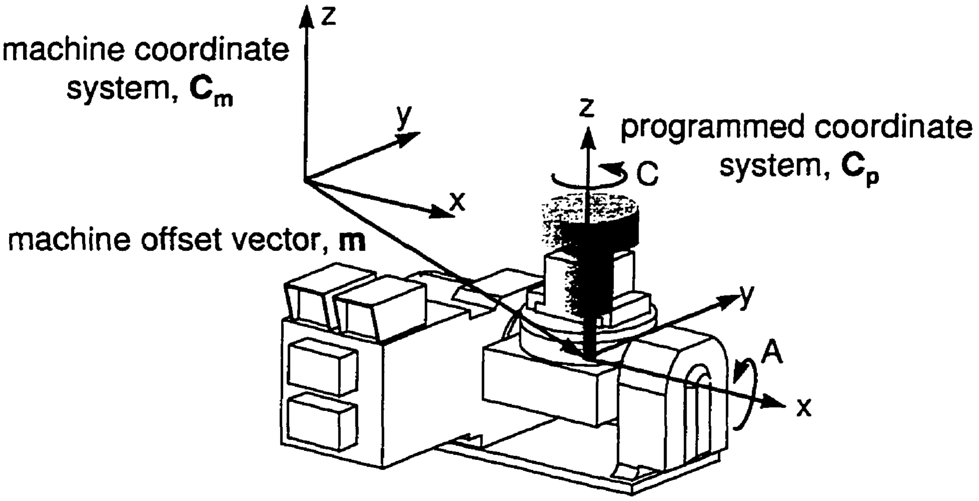

z
machine coordinate system, $\mathbf{C}_m$
y
x
machine offset vector, **m**
z
programmed coordinate system, $\mathbf{C}_p$
C
y
A
x

Figure D.2 Location of programmed coordinate system.

The correct location of the programmed coordinate system has the single largest impact on the accuracy of the final surface and must be accomplished with extreme care. It is determined in two stages. First of all, the spindle is placed directly above the (x, y) location point using the procedure illustrated in Figure D.3. A dial indicator is placed in the spindle of the milling machine. The axes of the machine are then moved so that the dial indicator stylus is in contact with the center hole in the top of the tilt-rotary table top. The dial indicator is then rotated in the spindle. If the spindle axis is not aligned with the rotation point, the dial indicator will change its reading as it rotates inside the center hole. The x and y axes are jogged incrementally until the dial indicator reading does not change. The position of the x and y axis at this point forms the x and y components of the machine offset vector, **m**. This procedure can locate the (x, y) components of the machine offset vector to within approximately 25 μm.

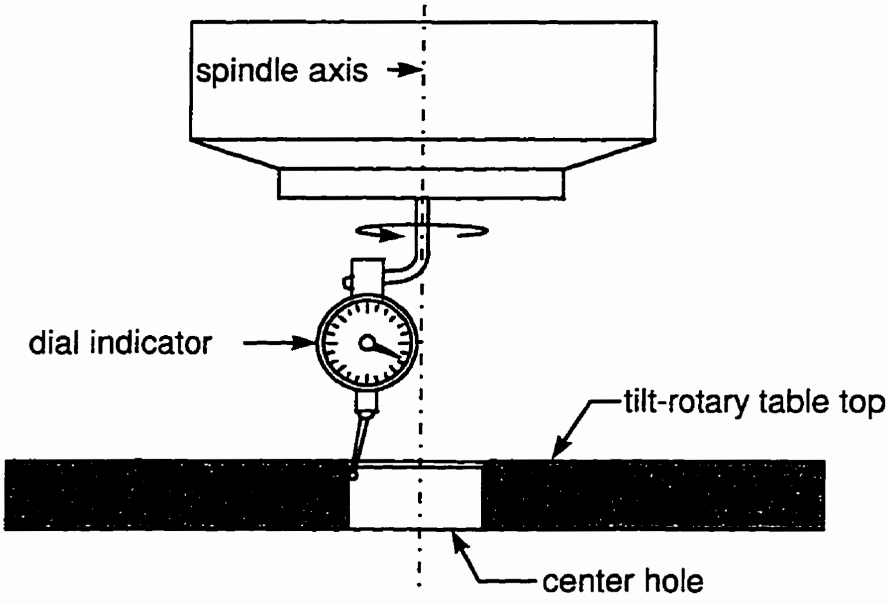


Figure D.3 Procedure to determining (x, y) location of rotation point.

The next step in determining the programmed coordinate system is to determine the z component of the machine offset vector. This component is dependent on the tool. A different tool length will require a different z-offset. The procedure illustrated in Figure D.4 was used for this purpose. Two scrap blocks of metal are attached to opposite sides to the tilt-rotary table top. The table is then rotated 90° about the A axis until it is in the vertical position and the tool is moved in the yz plane until it touches one of the blocks as shown in the figure. The C axis then makes one complete rotation. As the C axis rotates, the two blocks are cut into opposite sections of a cylinder. The diameter of this cylinder is measured with a vernier caliper. The radius of this cylinder is subtracted from the z-position of the tool to determine the z-component of the machine offset vector. The accuracy of this procedure is approximately 25 μm.

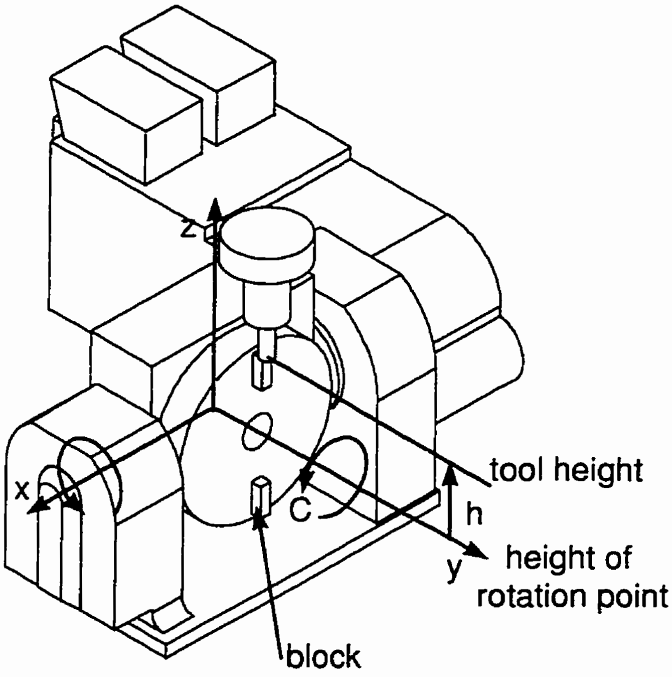


Figure D.4 Procedure for determining z-offset.

Once the programmed coordinate system is determined, the workpiece coordinate system can be established. It is specified by the workpiece offset vector, **w**, as shown in Figure D.5. This vector must be supplied to the post-processor so that it can convert the cutter location data into G-code commands specified in the programmed coordinate system. The workpiece offset vector is determined by placing the tool on a specified point on the workpiece relative to the programmed coordinate system. The difference between this tool position and the programmed coordinate system is the workpiece offset vector. Small errors in the location of the workpiece coordinate system will no doubt occur. This error will result in an error in the position of the surface on the workpiece. For this reason, prior to machining the surface, a coordinate system is machined into the workpiece. This coordinate system will share the same position error as the workpiece coordinate sytem, but relative to the surface it should be positioned correctly. It can then be used as a reference for measurement.

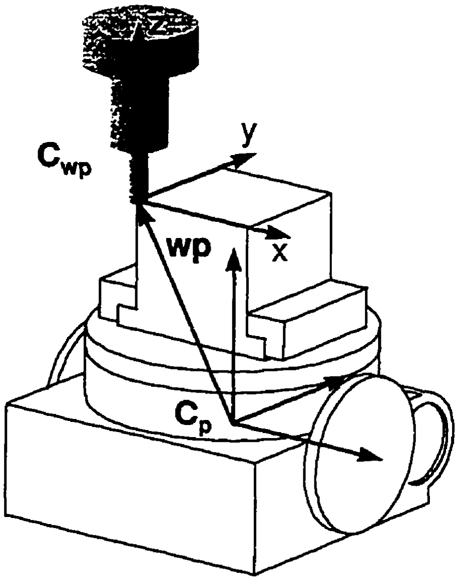


Figure D.5 Workpiece coordinate system.

Once a surfaces has been machined, it is measured on a coordinate measurement machine, CMM. This machine touches the surface with a spherical stylus and records the position of the center of the stylus. The CMM can be programmed to automatically measure a large number of points in the surface. The profile shown in Figure D.6 is an example of raw CMM data. Surface deviations must be extracted from the CMM data using the process illustrated in Figure D.7. In the figure the CMM stylus is in contact with the machined surface at point, $\mathbf{c}_{act}$. In order to determine the surface deviations the location of this point is approximated in the following manner. The measured point, $\mathbf{m}$, is offset from $\mathbf{c}_{act}$ the machined surface by some unknown vector whose magnitude is equal to the radius of the stylus, $\mathbf{r}_s$. This offset vector is in approximately the same direction as the surface normal, $\mathbf{n}_i$, at the point, $\mathbf{p}_i$, on the design surface. The approximate location of the contact point is found by offsetting the point, $\mathbf{c}_{app}$, in the direction of $-\mathbf{n}_i$ by the radius of the stylus, $\mathbf{r}_s$, from $\mathbf{m}$. The point $\mathbf{p}_{i+1}$ is then found by intersecting the surface with the line formed by $\mathbf{m}$ and $\mathbf{n}_i$. The surface deviation at $\mathbf{p}_{i+1}$ is then approximately $\left|\mathbf{p}_{i+1} - \mathbf{c}_{app}\right|$. Note that this result can be improved by repeating this process using the point, p_{i+1} and the surface normal n_{i+1}. There is little doubt that this procedure leads to inaccuracies in the surface deviation results. With CMM data, it is virtually impossible to determine the actual point which contacted the surface from the measured point. However, the results should be reasonable, provided the probe diameter and scallops are small. For this reason the smallest available probe, with a diameter of 1.0 mm, was used.

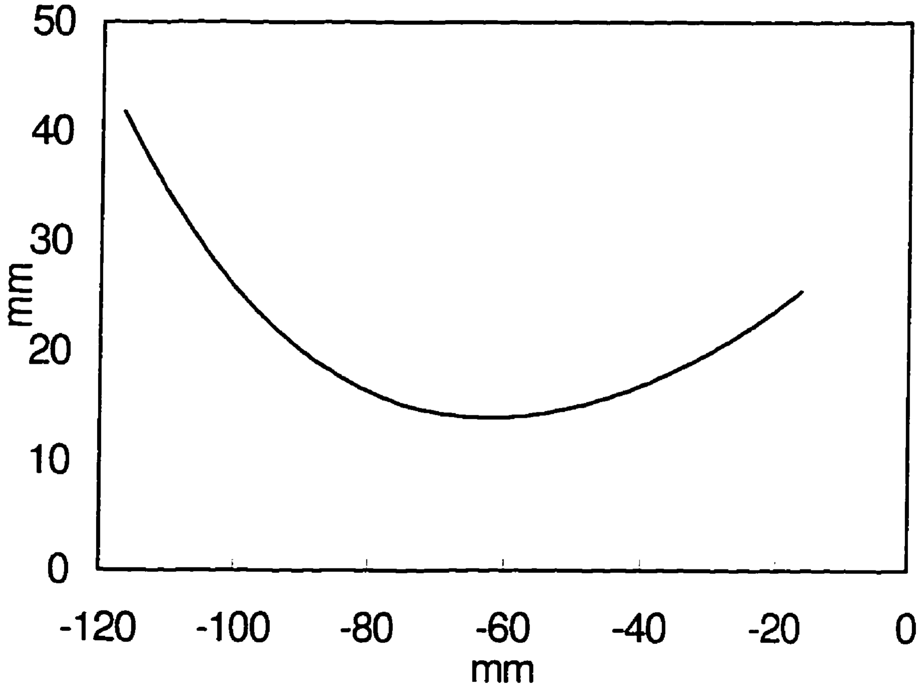


Figure D.6 Example of raw CMM data.

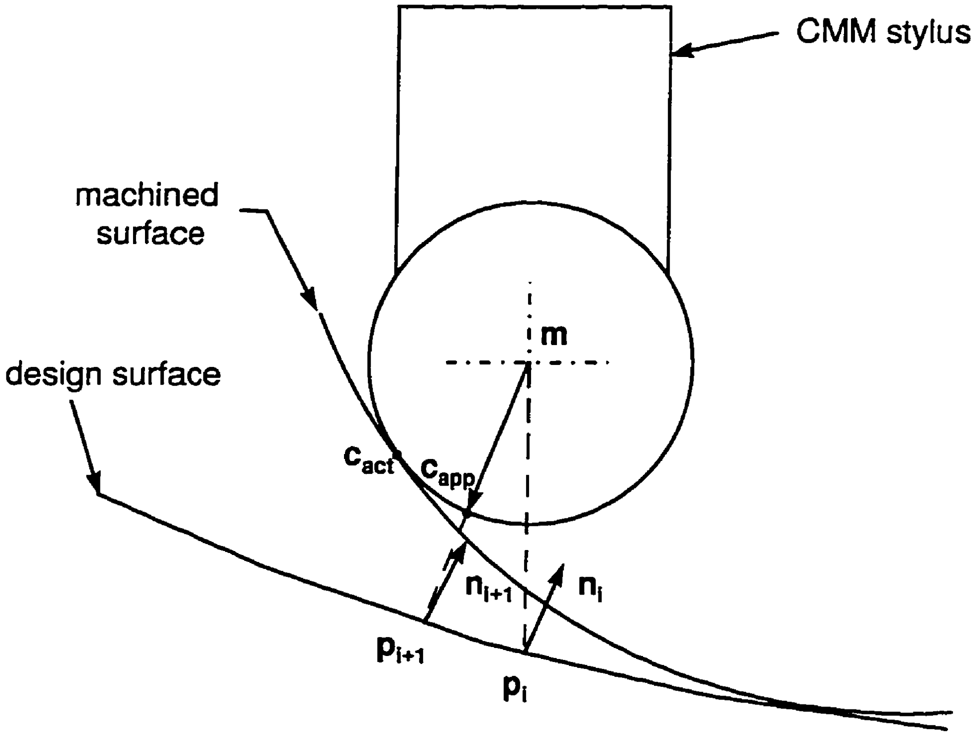


Figure D.7 Processing raw CMM data.