

## **Copyright Warning & Restrictions**

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

**Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation**

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

## **INFORMATION TO USERS**

**This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.**

**The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.**

**In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.**

**Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.**

**Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.**

# **UMI**

**A Bell & Howell Information Company  
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA  
313/761-4700 800/521-0600**



**UMI Number: 9730404**

**Copyright 1997 by  
Wang, Liping**

**All rights reserved.**

---

**UMI Microform 9730404  
Copyright 1997, by UMI Company. All rights reserved.**

**This microform edition is protected against unauthorized  
copying under Title 17, United States Code.**

---

**UMI**  
**300 North Zeeb Road**  
**Ann Arbor, MI 48103**

**MODELING OF 3D SWEPT VOLUMES USING SDE/SEDE  
METHODS AND ITS APPLICATION TO FIVE-AXIS NC  
MACHINING**

by  
**Liping Wang**

**A Dissertation  
Submitted to the Faculty of  
New Jersey Institute of Technology  
in Partial Fulfillment of the Requirements for the Degree of  
Doctor of Philosophy**

**Department of Mechanical Engineering**

**May 1997**

Copyright © 1997 by Liping Wang  
ALL RIGHTS RESERVED

## **APPROVAL PAGE**

### **MODELING OF 3D SWEEP VOLUMES USING SDE/SEDE METHODS AND ITS APPLICATION TO FIVE-AXIS NC MACHINING**

**Liping Wang**

---

Dr. Ming C. Leu, Dissertation Advisor NJ Sponsored Chair in Manufacturing/Productivity Professor of Mechanical Engineering, NJIT	Date
----------------------------------------------------------------------------------------------------------------------------------------	------

---

Dr. Denis Blackmore, Committee Member Professor of Mathematics, NJIT	Date
-------------------------------------------------------------------------	------

---

Dr. Reggie J. Caudill, Committee Member Professor of Industrial and Manufacturing Engineering, NJIT	Date
--------------------------------------------------------------------------------------------------------	------

---

Dr. Rajesh N. Dave, Committee Member Associate Professor of Mechanical Engineering, NJIT	Date
---------------------------------------------------------------------------------------------	------

---

Dr. Zhiming Ji, Committee Member Associate Professor of Mechanical Engineering, NJIT	Date
-----------------------------------------------------------------------------------------	------



## **ABSTRACT**

### **MODELING OF 3D SWEPT VOLUMES USING SDE/SEDE METHODS AND ITS APPLICATION TO FIVE-AXIS NC MACHINING**

by  
**Liping Wang**

This research falls in two important areas in solid modeling and manufacturing automation: (1) swept-volume modeling; (2) computer-based NC (Numerically controlled) machining simulation and verification. The swept volume is defined as the volume swept by an object undergoing an arbitrary motion. Modeling of 3D swept volumes includes the boundary computation and representation of a swept volume generated by a general object undergoing general motion in three dimensional space. The Sweep Differential Equation (SDE) and Sweep Envelope Differential Equation (SEDE) methods are two of the important swept volume modeling methods employed in this dissertation. They exploit differential equations to obtain the boundary points of a swept volume generated by a moving object. The application of SDE/SEDE methods is addressed to computer-based NC simulation and verification. Comparison of the SDE/SEDE approach with other swept volume modeling methods is conducted too. It has been shown that the SDE and SEDE methods have great benefits in calculating and representing general swept volumes and the research has substantially advanced existing manufacturing technologies.

The main contributions of the research are:

(1) The SDE method has been extended to three dimensional space to represent cutter swept volumes generated by moving five-axis NC milling tools. A SDE sweep generator, which can represent and analyze three-dimensional swept volumes generated by flat-end and ball-end tools for a typical five-axis NC milling machine graphically, has been developed. In the SDE sweep generator, a machine

control data based interpolation method is uniquely used to describe the interpolation motion equation of a five-axis NC milling tool.

(2) The SEDE method is derived for a more efficient swept volume calculation. A SEDE-based algorithm for the numerical boundary computation of swept volume is described and combined with some novel smooth approximation formulas in order to calculate the swept volume generated by a general 7-parameter APT (Automatic Programming Tool) tool for a large class of sweeps that includes the motions encountered in five-axis NC milling processes. The SEDE approach for the most part reduces the computation to the determination of SEDE trajectories at the initial grazing points (the main part of the boundary of a swept volume) of the tool, and therefore appears to reduce computational cost as well as providing a natural connectivity for most points on the swept volume boundary.

(3) An SEDE-based program has been integrated with Deneb Robotics's Virtual NC commercial software. The SEDE module is used to replace Virtual NC's convex hull sweep algorithm for a more accurate geometrical tool swept volume representation. By using the Boolean subtractor and verifier in Virtual NC, material removal of five-axis NC milling process is simulated and analyzed in an interactive machining environment. Furthermore, the SDE/SEDE approach has been integrated with a five-axis NC milling CAD/CAM system at NJIT to perform part design, tool path generation, Cutter Location (CL) and NC code simulation and verification, and actual machining on a FADAL VMC-20 five-axis NC milling machine. Several examples including machining of a turbine impeller are given to illustrate the effectiveness of this integration approach.

## BIOGRAPHICAL SKETCH

**Author:** Liping Wang  
**Degree:** Doctor of Philosophy  
**Date:** May 1997

### Undergraduate and Graduate Education:

- Doctor of Philosophy in Mechanical Engineering,  
New Jersey Institute of Technology,  
Newark, New Jersey, USA, 1997.
- Master of Engineering in Solid Mechanics,  
University of Science and Technology, Beijing, P.R. China, 1989.
- Bachelor of Science in Mechanics,  
Peking University, Beijing, P.R. China, 1984

**Major:** Mechanical Engineering

### Technical Presentations:

“Generating swept solids for NC verification using the SEDE method”, Fourth ACM Symposium On Solid Modeling and Applications, Atlanta, Georgia, May 16, 1997.

“Modeling of 3D swept volumes using SDE/SEDE methods and its application to NC machining”, Mechanical Engineering Department, City University of New York, New York, March 26, 1997.

“Determination of flat-end cutter orientation in 5-axis machining”, 1996 ASME International Mechanical Engineering Congress and Exposition, Atlanta, GA, Nov. 18, 1996.

“Implementation of SDE method to represent cutter swept volumes in 5-axis NC milling”, International Conference on Intelligent Manufacturing, Wuhan, China, June 15, 1995.

### Publications:

Blackmore, D., Leu, M.C., and Wang, L.P., “The sweep-envelope differential equation algorithm and its application to NC machining verification”, *Comput. Aided Des.*, 1996a. (accepted).

Blackmore, D., Leu, M.C., Wang, L., and Jiang H., "Swept volume: a retrospective and prospective view", *Journal of Neural, Parallel & Scientific Computation*, 1996b. (accepted).

Blackmore, D., Leu, M.C., Qin, D., and Wang, L., "An algorithm for computing swept volumes", *Proc. of Fourth SIAM Conference on Geometric Design*, Nashville, Tennessee, Nov. 6-9, 1995.

Deng, Z. and Wang, L.P., "An explicit formulation of the dynamic model for a chain of flexible links", *J. of Robotic Systems*, vol. 9, no. 8, pp. 1083-1094, 1992.

Deng, Z., Leu, M.C., Wang, L. and Blackmore, D., "Determination of flat-end cutter orientation in 5-axis machining", *Proc. of Manufacturing Science and Engineering-1996*, MED-Vol. 4, pp.73-80, 1996 ASME International Mechanical Engineering Congress and Exposition, Atlanta, Georgia, Nov.17-22, 1996.

Leu, M.C., Wang, L., and Blackmore, D., "A NC verification system for 5-axis NC machining with general APT tools", *Annals of the CIRP*, vol. 46/1, 1997. (accepted).

Leu, M.C., Blackmore, D., Wang, L. and Pak, K., "Implementation of SDE method to represent cutter swept volumes in 5-axis NC milling", *Proc. Int. Conf. on Intelligent Manufacturing*, pp. 211-220, Wuhan, China, June 14-18, 1995.

Wang, L., Leu, M.C., and Blackmore, D., "Generating swept solids for NC verification using the SEDE method", *Proc. of Fourth ACM Symposium On Solid Modeling and Applications*, Atlanta, Georgia, May 14-16, 1997.

Wang, L., Leu, M.C., and Blackmore, D., "Kinematics analysis of five-axis milling machine and its application to NC verification", *Proc. of International Conference on Manufacturing Automation*, Hong Kong, April 28-30, 1997.

Wang, L., Leu, M.C., and Blackmore, D., "Swept volume approach as an integral part of 5-axis NC milling CAD/CAM system", *Proc. of International Conference on Manufacturing Automation*, Hong Kong, April 28-30, 1997b.

Wang, L.P., "Semi-analytical finite element method in the process of direct cold drawing of section rods from round bars", M.Engr. Thesis, Department of Applied Mathematics and Mechanics, Univ. of Sci. & Tech. Beijing, 1989.

Wang, L.P., "Some methods on the solution of nonlinear finite element equations", B.Sc Thesis, Department of Mechanics, Peking University, 1984.

This dissertation is dedicated to  
my parents, Zhaocheng Wang and Zhongbin Zhu  
my husband, Zhongping Deng  
my son, Hongyi Deng

## ACKNOWLEDGMENT

I would like to express my sincere gratitude to Prof. Ming C. Leu, my advisor and chairman of my dissertation committee, for having introduced me to this project and his constant advice and support. Gratitude is due to Prof. Denis Blackmore of the Mathematics Department for his constant guidance, encouragement and patience. I owe a lot to them. The author would like to thank Prof. Zhiming Ji and Prof. Rajesh N. Dave of the Mechanical Engineering Department, Prof. Reggie Caudill of the Industrial and Manufacturing Engineering Department for serving as members on my dissertation committee. They took a keen interest in this research and provided valuable insights.

The author is grateful to Dr. Ronald Kane and Dr. R. Chen for graduate advice and assistance throughout these years.

The work of the author was partially supported by NSF grant DMS-9508808, ONR grant N00014-92-4093 and a Center for Manufacturing Systems grant from NJIT.

The technical support from Dr. David Perel of the Engineering Computing Laboratory at NJIT and Mr. David Lubliner from the Simulation and Modeling Labs at NJIT, who had an important role throughout out this work, is gratefully acknowledged.

Thanks are due to Jerry Eshbaugh and Rakesh Mahajan of Deneb Robotics for their help in the integration of our algorithm with their Virtual NC software.

Timely discussion, help and suggestions came from the members of the ONR research group, Dr. Kalin Liu, Mr. S. Kuan, and my friends at NJIT.

Last but not the least, the author would like to express her greatest and deepest gratitude to her husband, Dr. Zhongping Deng, for his constructive suggestions, love, faith, and patience and the many sacrifices he has made and for which he can never be fully compensated. Without him I would never have completed this dissertation.

## TABLE OF CONTENTS

Chapter	Page
1 INTRODUCTION .....	1
1.1 Motivation and Significance .....	1
1.2 Literature Review .....	2
1.2.1 Swept Volume Study and its Applications .....	2
1.2.2 NC Simulation and Verification .....	4
1.3 Objectives of this Work .....	8
1.4 Organization of this Dissertation .....	8
2 MATHEMATICAL MODELING OF SWEPT VOLUMES BY SDE/SEDE METHODS .....	10
2.1 Sweep Differential Equation (SDE) .....	10
2.2 Boundary Flow Formula (BFF) .....	13
2.3 Sweep-Envelope Differential Equation (SEDE) .....	14
2.4 Trimming Methods .....	20
2.5 Swept Volume for NC Simulation .....	20
2.5.1 General Tool Model and CL Data .....	20
2.5.2 Generalized Kinematics Model of Five-axis Milling Machine ..	22
2.5.3 Five-axis Motion .....	23
2.6 Remarks .....	24
3 SDE APPROACH TO MOVING TOOL IN FIVE-AXIS MOTION .....	35
3.1 Models of the Machine and Cutter .....	35
3.2 Sweep Equation .....	36
3.3 Sweep Vector Field .....	38
3.4 Tangency Function and Boundary Points .....	38
3.5 Boundary Trimming and Graphic Display of Swept Volume .....	40
3.6 Development of Computer Program .....	41

<b>Chapter</b>	<b>Page</b>
3.7 Illustrative Example . . . . .	42
3.8 Final Remarks . . . . .	42
4 GENERATING SWEPT SOLIDS FOR NC VERIFICATION USING THE SEDE METHOD . . . . .	50
4.1 SEDE Swept Volume Algorithm . . . . .	50
4.2 Decomposition of Tool into Closed Smooth Objects . . . . .	53
4.2.1 Lower Conical Part . . . . .	54
4.2.2 Toroidal Part . . . . .	55
4.2.3 Upper Conical part . . . . .	55
4.3 General Motion Equation . . . . .	56
4.4 Five-Axis Motion Equation . . . . .	57
4.5 Generating Initial Grazing Points . . . . .	58
4.6 Determining Grazing Set . . . . .	62
4.7 Determining Ingress and Egress Sets . . . . .	63
4.8 Boundary Trimming . . . . .	64
4.9 Swept Volume Representation and Graphic Display . . . . .	65
4.10 Complexity Analysis of the SEDE Algorithm . . . . .	66
4.11 SWEEP-SEDE Program: Implementation . . . . .	68
4.12 Illustrative Examples . . . . .	69
4.13 Error Analysis and Control . . . . .	72
4.13.1 Type 1: Error Caused by Tool Smoothing Approximation . . . .	73
4.13.2 Type 2: Error Caused by Tool Discretization . . . . .	74
4.13.3 Type 5: Error Caused by Polyhedral Approximation . . . . .	75
4.13.4 Conclusion . . . . .	75
4.14 Final Remarks . . . . .	75
5 APPLICATION OF SDE/SEDE APPROACH TO NC MACHINING AND COMPARISON WITH ENVELOPE METHOD . . . . .	92



Chapter	Page
5.1 Integration of SDE Approach with Research on Five-Axis NC Tool Path Generation . . . . .	92
5.1.1 Determination of Flat-End Cutter Orientation in 5-Axis Machining by Curvature Matching Method . . . . .	93
5.1.2 Verification of CL Data Using SDE-3D Sweep Generator . . . . .	93
5.1.3 Example . . . . .	94
5.2 Integration of SEDE Approach with Commercial NC Software . . . . .	94
5.2.1 About Virtual NC . . . . .	95
5.2.2 Integration Procedures . . . . .	95
5.2.3 Modeling and Kinematics Analysis of FADAL VMC-20 . . . . .	97
5.2.4 Illustrative Examples . . . . .	101
5.3 Integration of SEDE Sweep Module with 5-Axis NC Milling CAD/CAM System . . . . .	103
5.3.1 System Description . . . . .	103
5.3.2 Case Study : Turbine Impeller . . . . .	105
5.4 Comparison of SDE/SEDE Methods with Envelope Method . . . . .	106
5.4.1 General Comparison . . . . .	106
5.4.2 Numerical Calculation: Computer Program . . . . .	108
5.5 Final Remarks . . . . .	109
6 CONCLUSIONS AND FUTURE WORK . . . . .	122
APPENDIX A KINEMATICS ANALYSIS OF GENERAL 5-AXIS NC MILLING MACHINES . . . . .	124
APPENDIX B GENERAL 7-PARAMETER APT TOOLS . . . . .	126
REFERENCES . . . . .	131

## LIST OF FIGURES

Figure	Page
2.1 A typical swept volume of an object in space . . . . .	25
2.2 Decomposing object boundary . . . . .	26
2.3 Swept volume orbits of SDE . . . . .	27
2.4 Swept volume orbits of SEDE . . . . .	28
2.5 Local trimming procedure . . . . .	29
2.6 Global trimming procedure . . . . .	30
2.7 General 7-parameter APT tool and CL data . . . . .	31
2.8 Some tool geometries represented by 7-parameter APT tool model . . . .	32
2.9 Three types of five-axis NC machines and MCD data . . . . .	33
2.10 Six kinds of configurations for Type B five-axis machines . . . . .	34
3.1 A five-axis NC milling machine and reference coordinate frame $xyz$ . . . .	43
3.2 Tool geometry and moving coordinate frame $\xi\eta\zeta$ . . . . .	44
3.3 Relationship of the coordinate frames . . . . .	45
3.4 Geometric relationship between MCD data and CL data . . . . .	46
3.5 Denavit-Hartenberg representation for the machine . . . . .	47
3.6 Flow chart of the SDE-3D computer program . . . . .	48
3.7 Swept volume and some of the surface patches . . . . .	49
4.1 Flow chart of calculation of grazing points by SEDE . . . . .	77
4.2 General 7-parameter APT tool . . . . .	78
4.3 Approximation of a cone surface in a single smooth function . . . . .	79
4.4 Approximation of flat-end tool surface in a single smooth function . . . .	80
4.5 Moving cutters under one-block five-axis motions . . . . .	81
4.6 Graphical output for Example 4.1 . . . . .	82
4.7 Grazing points generated for Example 4.2 . . . . .	83

<b>Figure</b>	<b>Page</b>
4.8 Grazing curves generated for Example 4.2 . . . . .	84
4.9 Swept volume generated for Example 4.2 . . . . .	85
4.10 TCP path and tool configurations for Example 4.3 . . . . .	86
4.11 Grazing points generated for Example 4.3 . . . . .	87
4.12 Swept volume generated for Example 4.3 . . . . .	88
4.13 TCP path and tool configurations for Example 4.4 . . . . .	89
4.14 Grazing points generated for Example 4.4 . . . . .	90
4.15 Swept volume generated for Example 4.4 . . . . .	91
5.1 NC simulation for calculated inclination and yaw angles . . . . .	110
5.2 Integration of SWEEP-SEDE sweep module with commercial software . .	111
5.3 FADAL VMC-20 five-axis milling machine . . . . .	112
5.4 Simulated workcell of FADAL VMC-20 five-axis milling machine . . . . .	113
5.5 Denavit-Hartenberg representation of FADAL VMC-20 . . . . .	114
5.6 Graphical output for Example 5.1 . . . . .	115
5.7 Graphical output for Example 5.2 . . . . .	116
5.8 Integrated CAD/CAM system for five-axis NC milling process . . . . .	117
5.9 Designed part of turbine impeller . . . . .	118
5.10 NC program simulation in progress on Virtual NC . . . . .	119
5.11 Turbine machining simulation by SEDE and VNC . . . . .	120
5.12 Completed parts: SLA, wax, metal (from left to right) . . . . .	121
B.1 General 7-parameter APT tool . . . . .	127
B.2 Calculation of normals in tool coordinate frame . . . . .	128

# **CHAPTER 1**

## **INTRODUCTION**

In this chapter, the significance and main difficulties of this research and its current research and application status are presented followed by the project objectives.

### **1.1 Motivation and Significance**

This research falls in two important areas in solid modeling and manufacturing automation: (1) swept-volume study; (2) computer-based NC simulation and verification.

Development of fast and accurate sweep generators, which can represent and analyze swept volumes generated by a general three-dimensional object undergoing general motion has been an important research area in solid modeling and Computer Aided Design technology. This research has important applications in manufacturing automation such as NC machining, robot motion planning, mechanism synthesis and automated assembly. Research on swept volume has potential for advancing existing manufacturing technologies, thereby resulting in an increase of manufacturing productivity, product quality and cost reduction.

Automatically or manually generated NC programs often contain several types of errors. These errors range from simple syntax mistakes to subtle interference with fixtures. It therefore becomes necessary to 'verify' NC programs prior to execution on a CNC machine and this has evolved into an important field - computer-based NC program simulation and verification. In NC verification, the geometry of the machined part can be expressed as the difference between the stock and the swept volume of the cutter. Collisions between the cutter and the fixture or workpiece can be detected by checking whether the intersection between them is a null object. Therefore, fast and accurate sweep generators, Boolean subtractors, intersection

detectors and graphic displays play important roles in building an NC simulator and verifier.

In short, it is the importance of swept volume theory and its applications that inspired the author's interest on the modeling of 3D swept volumes and its application to NC machining.

## **1.2 Literature Review**

### **1.2.1 Swept Volume Study and its Applications**

The concept of swept volume was initiated in the late 1970's and 1980's to study manufacturing automation strategies. A great amount of effort has been devoted to the development of fast and accurate methods to represent and analyze the swept volume of a three-dimensional object under a general rigid-body motion. The most useful methods include the envelope theory, Z-buffer method, ray-casting method, or a combination of these methods (Wang & Wang, 1986a, 1986b; Narvekar, 1991; Sambandan, 1988, 1990; Leu et al., 1986; Weld & Leu, 1990).

Applications of swept-volume studies to NC machining have been proposed for the simulation of the material removal process in NC machining, detection of collisions, generation of a gouge-free tool paths, etc. Applications also include mechanism synthesis (Ling, 1995) and autonomous vehicle motion planning (Deng, et al., 1994a).

Leu et al. (Leu et al., 1986) developed a rather efficient method for representing translational swept volumes of objects. Weld & Leu (Weld & Leu, 1990) proved that the swept volume of an object can be obtained essentially from the swept volume of the boundary of the object. This theorem was used to study the change from a ruled surface to a developable surface in the sweep of a polygon.

The envelope technique (Wang & Wang, 1986a, 1986b) was one of the earliest attempts to compute the swept volumes generated by developable surfaces. Based on

this theory, Sambandan (Sambandan, 1988) developed a five-axis NC simulator for flat-end, ball-end and fillet-end cutters in image space and derived the parametric representation of the bounding surfaces of the cutter swept volume. The object's geometry also included any regular polyhedral objects (Sambandan, 1990).

Narvekar (Narvekar, 1991) derived the envelope equations for cylindrical, conical, spherical and toroidal surfaces of a general 7-parameter APT tool and also conducted intersection calculations. These representations were combined to model the swept surfaces of standard NC milling tools under general five-axis motions. Tool swept volumes were represented by these swept surfaces combined with the models of the tool in the initial and final positions.

Sambandan's and Narvekar's works rely on hidden line removal method in computer graphics to find graphically the exact boundary of a swept volume. This method may not be accurate enough for real machining simulation and verification.

There are other useful methods for representing and analyzing swept volumes. For example, fast and accurate renderings of general swept volumes may be obtained by using the Raycasting Engine (Menon & Robinson, 1993).

The recent work by Blackmore & Leu (Blackmore & Leu, 1990; Blackmore et al., 1992a, 1992b) introduced a new approach, called the sweep differential equation (SDE) method, for the study of swept volumes. This approach fully exploits the Lie group structure of the set of Euclidean motions, thereby enabling the problems of involving swept volumes to be reformulated in the context of differential equations. The potential of this approach for automated manufacturing applications has been discussed and demonstrated in robot motion planning (Blackmore et al., 1992c; Deng et al., 1994a) and NC verification (Deng et al., 1994b).

The SDE theory was implemented initially for two dimensional objects. A computer program was developed by Jiang (Jiang, 1993) to model the swept volumes of polygonal objects undergoing general planar sweeps. A set of candidate

points was calculated by the SDE method and trimmed to form the boundary of the swept volume. Qin et al. (Qin et al., 1994) modified Jiang's work by introducing an envelope differential equation method along with the sweep differential equation method to generate the grazing points for the swept-volume boundary more efficiently. When the grazing points, which are the candidate points to form the boundary of a swept volume, occur at the vertices of a planar polygon, the sweep differential equation method is relied on to determine which vertices are grazing points. When the grazing points are interior to the boundary edges of the object, they are obtained as the solutions of the envelope differential equation. The 2D computer program was also extended to include objects experiencing deformation (Blackmore et al., 1994), by enlarging the Lie group structure of the sweep. All of the usual results of rigid sweeps, including the boundary flow formula, were shown to have extensions for swept volumes with deformation. The difficulties associated with developing theories and algorithms for solving three dimensional motion verification and planning problems are much greater than those for planar problems.

Deng et al. (Deng et al., 1994b, 1996) established a model of 5-axis milling of a sculptured surface and applied the SDE approach to analyze the differential geometric characteristics of the machined surface. The full advantages of 5-axis milling were exploited by considering the tool inclination and yaw capabilities. The results demonstrate that the curvature matching of the machined surface can be improved by adjusting inclination and yaw angles, and that the machining error (i.g., cusp height) of the surface increases as the curvature of the machined surface increases.

### 1.2.2 NC Simulation and Verification

The most successful application of the swept volume research thus far is computer-based NC simulation and verification. In general, NC simulators and automatic

verifiers detect spatial and dynamic errors (Voelcker & Menon, 1989; Menon & Voelcker, 1993b).

**Spatial (geometric) errors** in NC machining are related to cutter collision and near miss machining. They are detected mainly by performing intersection tests with

$$P \cap V_i = \Phi?$$

where  $P$  is the desired part,  $V_i$  is the volume swept by the cutter on the  $i$ th command. The issues to address are : does the current cutter swept volume intersect the desired part  $P$  ? When and in which block of NC code a collision or near miss occurs?

**Machining dynamics errors** are related to cutter breakage, violation of tolerance constraints. The computation mainly involves tool deflection and machining force calculations. The verification of these kinds of errors is usually done using relevant models for predicting machining power, tool deflection and machining force (Menon, 1990).

An NC simulator or verifier can be classified as a visual simulator/verifier or automatic simulator/verifier as follows:

**Visual NC program simulators/verifiers** model the effects of machining via

$$W_i = W_{i-1} - V_i$$

where  $W_i$  is a representation of a workpiece after execution of the  $i$ th NC command,  $V_i$  is the 'operative spatial volume', i.e., volume swept by the cutter on the  $i$ th command and '-' is the Boolean subtractor. An NC simulator reads an NC program block-by-block and displays the workpiece after each command; a person watches the displays and tries to spot problems such as collisions, near miss, etc. These kinds of simulators are crude and rely on human experience and graphics display parameters. Gross errors can be detected, but small errors like small undercut may be missed.



**Automatic NC program verification** seeks to detect problems computationally, without recourse to human observers. This is considerably more difficult than graphic simulation because ‘problems’ must be modeled mathematically and tested by algorithms. In principle, automatic NC program verification does not need graphics display. Graphic display merely provides visual and diagnostic aids. Automatic NC program verifiers may test both spatial and machining dynamics errors.

Some graphical NC simulators/verifiers have been developed. These simulators are based initially on wireframe displays, and later, based on image-space, sculptured surface and solid-modeling techniques. The Vericut<sup>TM</sup> system from CG Tech, Pro/Engineer<sup>TM</sup> and Pro-Manufacture<sup>TM</sup> from Parametric Technology, I-DEAS<sup>TM</sup> from SDRC, Virtual NC<sup>TM</sup> from Deneb Robotics, Inc. are examples of commercially available NC simulators/verifiers. They mainly focus on spatial error detection. One of the examples in **wireframe** display was described by Anderson (Anderson, 1978). Using the image-space or ‘extended Z-buffer’ method, near real-time graphic displays are produced by evaluating Boolean operations via one-dimensional updates of pixel information (Wang & Wang, 1986a). Vericut<sup>TM</sup> is an example of commercial software based on this technique. Image-space methods support 5-axis machining simulation, but the approach is view-dependent and may not be accurate enough for real machining problems involving collision detection.

**Sculptured surface techniques** model a portion of the boundary of the workpiece (usually the ‘top’ surface) by a suitable set of points and associated vectors. They have been developed mainly for sculptured surface machining. Machining simulation is done by ‘cropping’ these vectors as they intersect the tool-sweep envelope. Various approaches in this category differ in their techniques for computing a suitable set of boundary points and associated vectors (Chappel,

1983; Drysdale et al., 1989; Jerard et al., 1989; Oliver & Goodman, 1990). These systems are used primarily as simulators, with some automatic verification functions.

**Solid modeling techniques** have been widely adopted in this area. Fridshal et al. (Fridshal et al., 1982) developed an early CSG-based (CSG=Constructive Solid Geometry) NC simulator in the TIPS-1 geometric modeling environment. It was very slow, but could handle some types of free-form parts. Pro/Engineer<sup>TM</sup>, I-DEAS<sup>TM</sup> and Virtual NC<sup>TM</sup> are the examples based on this technique. Deneb's Virtual NC<sup>TM</sup> (VNC) is basically for spatial error detection and has some functions on machining dynamics error detection. Powerful functions in VNC indicate the conditions that cause chatter and tool breakage, or gouges and undercuts. All tooling data is linked to each tool. Monitoring functions automatically record the time spent and the volume of material removed by a tool to track wear and predict failure.

So far, little work has been done on automatic NC verification, especially on testing machining dynamics related errors. Menon et al. described NC verification as a mathematical and computational problem (Menon & Voelcker, 1993) and have implemented these models in another paper (Menon & Robinson, 1993). A publicly available software package based on this approach is P2NC developed at Cornell University, operating primarily as a simulator but offering various levels of automatic assessment. Average estimates of cutting forces and deflections for each cutter move (NC block) have been computed in the CSG-based P2NC system. It is effectively limited to 2.5-axis machining simulation with simple cutter geometries and very slow in processing successive motions.

In short, fast and accurate sweep generators, Boolean subtractors, intersection detectors and graphics display environments play important roles in building NC simulators and verifiers to detect spatial/geometric errors. For automatic NC verifications, algorithms and conditions must be modeled mathematically. A motion

planner and machine models are needed for simulation and verification in a machining environment.

### **1.3 Objectives of this Work**

Based on the literature review and analysis, the objectives of this dissertation are as follows:

(1) Extend implementation of SDE theory to three dimensional space: Develop an SDE algorithm for swept volume computation and representation; construct an SDE sweep generator to represent and analyze three-dimensional swept volumes generated by moving tools in 5-axis NC milling graphically and; investigate the Machine Control Data (MCD) based interpolation method to approximate the tool motion in five-axis NC milling processes.

(2) Develop novel means of implementing a new differential equation approach called the sweep-envelope differential equation (SEDE) method for the study of the swept volume of a moving object and explore its properties.

(3) Construct SEDE algorithms and generate swept solids for NC verification using the SEDE method.

(4) Integrate the SEDE sweep module with commercial NC software and an integrated five-axis NC milling CAD/CAM system at NJIT to show the real utility of the SEDE method for solving problems in manufacturing automation.

### **1.4 Organization of this Dissertation**

This dissertation is organized as follows. In chapter 2, modeling of swept volumes mathematically by a differential equation approach and the basics of swept volumes for NC machining are presented. In chapter 3, an algorithm and implementation of the SDE method are described with some illustrative examples. In chapter 4, an algorithm and implementation of SEDE method are presented along with some novel

smooth approximation formulas for general 7-parameter APT tools and a large class of sweeps that includes the motions encountered in five-axis NC milling processes. Chapter 5 describes the application of the SDE and SEDE in five-axis NC machining in various aspects and the comparison of the SDE/SEDE methods with another swept volume modeling method. Conclusions and future work are given in Chapter 6.

## CHAPTER 2

### MATHEMATICAL MODELING OF SWEPT VOLUMES BY SDE/SEDE METHODS

As described in Chapter 1, modeling of swept volumes has been a challenging research topic and shown great potential in manufacturing automation. In this chapter, a new modeling method called the Sweep Envelope Differential Equation (SEDE) in the context of differential equation is derived and serves as an important contribution in solid modeling. The swept volume study for NC simulation is defined explicitly in terms of the tool object, five-axis motion interpolation and different types of five-axis NC milling machines. To begin with, basic definitions on swept volume, sweep, SDE, and Boundary flow Formula (BFF), which are already available from the work done by Blackmore and Leu (Blackmore et al., 1992a, 1992b), are summarized.

#### 2.1 Sweep Differential Equation (SDE)

Let  $\mathbb{R}$  represent real numbers.  $n$ -dimensional space  $\mathbb{R}^n$  is considered. The points in  $\mathbb{R}^n$  can be represented by an  $n$ -dimensional vector.  $n = 2$  and  $n = 3$  are primarily concerned. Hence a point in space  $P(x)$  can be identified with  $x$  in  $\mathbb{R}^3 = \{x = (x_1, x_2, x_3)^T : x_1, x_2, x_3 \in \mathbb{R}\}$ . This space carries a standard inner (dot) product

$$\langle x, y \rangle = \sum_{k=1}^3 x_k y_k$$

with norm  $|x|$ .

**Definition 1:** A *smooth sweep*  $\sigma$  in  $\mathbb{R}^n$  is a smooth ( $= C^\infty$ ) mapping

$$\sigma : [0, 1] \rightarrow \mathbb{R}^n \times SO(n)$$

such that  $\sigma$  is the identity mapping at  $t = 0$ , where  $SO(n)$  is the special orthogonal group in  $\mathbb{R}^n$  consisting of all real  $n \times n$  matrices  $\tau$  satisfying  $\tau\tau^T = I$  and  $\det\tau = 1$ .

From this definition, if  $\sigma_t$  represents a rigid-body motion at time  $t$ , it can be written as:

$$\sigma_t(x) = \Gamma(t) + \tau(t)x \quad (2.1)$$

where  $\Gamma: [0,1] \rightarrow \mathbb{R}^n$  and  $\tau: [0,1] \rightarrow SO(n)$  are smooth mappings satisfying  $\Gamma(0) = 0$  and  $\tau(0) = I$ .

It is assumed that the object  $M$  undergoing the sweep is a compact,  $n$ -dimensional submanifold of  $\mathbb{R}^n$  such that its boundary  $\partial M$  is piecewise smooth.

**Definition 2:** Let  $\sigma$  be a sweep in  $\mathbb{R}^n$ . The *swept volume* of object  $M$  generated by the sweep  $\sigma$  is

$$S_\sigma(M) = \{\sigma_t(x) : x \in M, t \in [0,1]\} \quad (2.2)$$

where the sets  $\sigma_t(M) = M(t) = \{\sigma_t(x) : x \in M\}$  are the  $t$ -sections of  $M$  generated by the sweep and  $\sigma(x) = \{\sigma_t(x) : 0 \leq t \leq 1\}$  is the  $\sigma$ -trajectory of  $x$ . A typical swept volume of an object in space is illustrated in Fig.2.1.

**Definition 3:** The *sweep vector field* (SVF) of a smooth sweep  $\sigma$  is given by

$$\dot{x} = X_\sigma(x, t) = \dot{\Gamma}(t) + \dot{\tau}(t)\tau^\top(t)(x - \Gamma(t)) \quad (2.3)$$

where dot denotes differentiation with respect to  $t$ .  $X_\sigma$  forms the sweep vector field (SVF) of  $\sigma$ .

It is easy to verify that there is a unique solution  $x(t) = \sigma_t(x^\circ)$  of Eqn.(2.3) satisfying the initial condition  $x(0) = x^\circ$ . As Eqn. (2.3) is linear with time varying coefficients, its solution is defined on all of  $\mathbb{R}^n$  if both  $\Gamma(t)$  and  $\tau(t)$  are also smooth on all of  $\mathbb{R}^n$ . In this case then, the sweep  $\sigma$  can be smoothly extended to all of  $\mathbb{R}^n$ .

**Definition 4:** Let  $M$  be a piecewise smooth object. On the smooth part of the  $\partial M$ , the *tangency function* for the sweep  $\sigma$  of object  $M$  is defined as

$$T(x, t) = \langle X_\sigma(x, t), N(x) \rangle \quad (2.4)$$

where  $\langle a, b \rangle$  denotes the standard inner product of  $a$  and  $b$  in  $\mathbb{R}^n$ ,  $N(x)$  is an outer normal vector field on the smooth part of the  $\partial M$  at point  $P(x)$ .

**Definition 5:** Let  $M$  be a piecewise smooth object as above and  $\sigma$  be a smooth sweep with SVF being  $X_\sigma$ . A point  $P(x)$  on  $\partial M$  is called an *ingress point* if the orbit of Eqn.(2.3) at  $P(x)$  is directed into the interior of  $M$ , it is an *egress point* if the orbit of Eqn.(2.3) at  $P(x)$  is headed out of the interior of  $M$ , and it is a *grazing point* if it is neither an ingress nor an egress point. The set of ingress, egress and grazing points of  $M(t)$  is denoted by  $\partial_- M(t)$ ,  $\partial_+ M(t)$  and  $\partial_0 M(t)$ , respectively. In general, for an  $n$ -dimensional  $M$ ,  $\partial_- M(t)$  and  $\partial_+ M(t)$  are  $(n - 1)$ -dimensional while  $\partial_0 M(t)$  is  $(n - 2)$ -dimensional.

Based on this definition,  $\partial M(t)$  is partitioned into three parts: ingress, egress and grazing points,

$$\partial M(t) = \partial_- M(t) \cup \partial_0 M(t) \cup \partial_+ M(t) \quad (2.5)$$

If  $P(x)$  belongs to the smooth part of  $\partial M$ , the set of ingress, egress and grazing points of  $M(t)$  can be decided when the tangency function  $T(x, t)$  in **Definition 4** is less than, greater than, or equal to zero, respectively, i.e.

$$\partial_- M(t) = \cup \{T(x, t) < 0\} \quad (2.6)$$

$$\partial_+ M(t) = \cup \{T(x, t) > 0\} \quad (2.7)$$

$$\partial_0 M(t) = \cup \{T(x, t) = 0\} \quad (2.8)$$

If  $P(x)$  belongs to the edge of two contiguous smooth portions of  $\partial M$ , we can assign two outer normal vectors to  $P(x)$ , say  $N_i$  and  $N_j$  for each of the abutting

smooth faces. If  $\langle X_\sigma, N_i \rangle$  and  $\langle X_\sigma, N_j \rangle$  are both negative,  $P(x)$  is an ingress point. If they are both positive,  $P(x)$  is an egress points; otherwise  $P(x)$  is a grazing point. The ingress, egress and grazing points are illustrated in Fig. 2.2 (Blackmore et al., 1992a).

This leads to a proof of the following result (Blackmore et al., 1992a) called the *boundary flow formula* (BFF).

## 2.2 Boundary Flow Formula (BFF)

**Definition 6:** The *candidate set* of points for the boundary of the swept volume of an object  $M$  undergoing a smooth sweep  $\sigma$ , denoted as  $C_\sigma(M)$ , is defined as:

$$C_\sigma(M) = \partial_- M(0) \cup \{\partial_0 M(t) : 0 \leq t \leq 1\} \cup \partial_+ M(1) \quad (2.9)$$

It can be shown that the boundary of the swept volume of  $M$ , represented by  $\partial S_\sigma(M)$ , is contained in the candidate set  $C_\sigma(M)$ , i.e.

$$\partial S_\sigma(M) \subseteq C_\sigma(M) \quad (2.10)$$

**Definition 7:** The *trimming set* for  $S_\sigma(M)$ , denoted by  $T_\sigma(M)$ , is composed of all points  $x$  in  $C_\sigma(M)$  such that  $x$  belongs to the interior of some  $M(t)$  and needs to be subtracted (or trimmed) from  $C_\sigma(M)$  to obtain the true boundary of the swept volume. So  $\partial S_\sigma(M)$  can be expressed as

$$\partial S_\sigma(M) = C_\sigma(M) \setminus T_\sigma(M) \quad (2.11)$$

**Definition 8:** The *local trimming set* (*global trimming set*) for  $S_\sigma(M)$ , denoted by  $LT_\sigma(M)$  ( $GT_\sigma(M)$ ), is composed of all points  $x$  in  $T_\sigma(M)$  such that  $x$  belongs to the interior of only one (more than one)  $M(t)$ . So  $T_\sigma(M)$  can be expressed as

$$T_\sigma(M) = LT_\sigma(M) \cup GT_\sigma(M) \quad (2.12)$$



**Theorem (BFF):** Let the object  $M$  and sweep  $\sigma$  be as described above. Then the boundary of the swept volume is given by the formula

$$\partial S_\sigma(M) = C_\sigma(M) \setminus T_\sigma(M) \quad (2.13)$$

where  $C_\sigma(M) = \partial_- M(0) \cup \partial_+ M(1) \cup G_\sigma(M)$ ,  $G_\sigma(M) = \cup \{\partial_0 M(t) : 0 \leq t \leq 1\}$  and  $T_\sigma(M)$  consists of those points of  $C_\sigma(M)$  that belongs to the interior of  $S_\sigma(M)$ .

The calculation of swept volume based on the SDE and BFF is illustrated in Fig.2.3. The major part of the calculation of a swept volume using the BFF is that of the grazing set and the trim set. For an autonomous sweep where  $X_\sigma$  is independent of  $t$  ( $\partial_t X_\sigma = 0$ ), the grazing set is just the set of all orbits of the SDE starting at  $\partial_0 M(0)$ . But for a non-autonomous sweep, swept volume calculation based on the SDE and BFF has shortcomings both in determining the grazing set and surface fitting. To overcome these shortcomings, a new method, called the sweep-envelope differential equation (SEDE) is developed. The development of the SEDE method is the culmination of a search for an ordinary differential equation that generates the grazing set for nonautonomous sweeps just as the SDE does for autonomous sweeps.

### 2.3 Sweep-Envelope Differential Equation (SEDE)

The sweep-envelope differential equation is derived and its main properties are discussed in this section.

As usual, it is assumed that the reference coordinate system is Cartesian and imbedded in the work piece. The goal is to find a differential equation of the form

$$\dot{x} = \Phi_\sigma(x, t) = X_\sigma(x, t) + Y_\sigma(x, t) \quad (2.14)$$

having the property that the set of all of its trajectories initiating from points on  $\partial_0 M(0)$  covers or foliates the grazing set  $G_\sigma(M)$ . To begin, it is observed that there is a smooth function  $f : \mathbb{R}^3 \rightarrow \mathbb{R}$  such that  $f$  is negative in the interior of  $M$ , zero on

$\partial M$  and positive in the exterior of  $M$ . The function  $F : \mathbb{R}^3 \times [0, 1]$  defined by

$$F(x, t) = f(\eta(x, t)) \quad (2.15)$$

where

$$\eta(x, t) = \tau^T(t)(x - \Gamma(t)), \quad (2.16)$$

has the property that it is negative in the interior of  $M(t)$ , zero on  $\partial M(t)$ , and positive in the exterior of  $M(t)$  for all  $0 \leq t \leq 1$ . It is noted that if  $x \in \partial M(t)$ , the vector  $\tau(t)\partial_x f(\eta(x, t))$  is an outward normal vector to  $\partial M(t)$ , where  $\partial_x$  denotes the partial differential operator  $(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3})^T$ . Therefore

$$N = |\tau(t)\partial_x f(\eta)|^{-1} \tau(t)\partial_x f(\eta) \quad (2.17)$$

is an outward unit normal to  $\partial M(t)$ . It follows from the definition of  $F$  that the grazing set can be characterized as the set of all points  $x$  in  $\mathbb{R}^3$  that satisfy the simultaneous equations

$$F(x, t) = 0 \quad (2.18)$$

and

$$\partial_t F(x, t) = 0 \quad (2.19)$$

for  $0 \leq t \leq 1$ , where  $\partial_t$  denotes partial differentiation with respect to  $t$ . But (2.18)-(2.19) are just the usual envelope equation for the family of surfaces  $F(x, t) = 0$ . In this context, the objective is to find a differential equation (2.14) with solutions  $x = x(t)$  such that  $x(0) \in \partial M(0)$  and

$$F(x(t), t) = 0 \quad (2.20)$$

and

$$\partial_t F(x(t), t) = 0 \quad (2.21)$$

for all  $0 \leq t \leq 1$ .

Differentiating equation (2.20) using the chain rule, Eqn.(2.15), Eqn.(2.16), and the fact that  $\tau\tau^T = \tau^T\tau = I$  and  $\tau\dot{\tau}^T = -\dot{\tau}\tau^T$ , one obtains

$$\begin{aligned}
 0 = \frac{d}{dt}F(x(t), t) &= \frac{d}{dt}f(\eta(x, t)) = \frac{d\eta}{dt} \cdot \partial_x f = \langle \partial_x f, \frac{d\eta}{dt} \rangle \\
 &= \langle \partial_x f(\eta), \dot{\tau}^T(x - \Gamma) + \tau^T(\dot{x} - \dot{\Gamma}) \rangle \\
 &= \langle \partial_x f(\eta), \tau^T[\tau\dot{\tau}^T(x - \Gamma) + \dot{x} - \dot{\Gamma}] \rangle \\
 &= \langle \partial_x f(\eta), \tau^T[\dot{x} - \dot{\Gamma} - \dot{\tau}\tau^T(x - \Gamma)] \rangle
 \end{aligned}$$

Hence it follows from (2.3) and (2.14) that

$$\begin{aligned}
 0 = \frac{d}{dt}F(x(t), t) &= \frac{d}{dt}f(\eta(x(t), t)) \\
 &= \langle \partial_x f(\eta), \tau^T(\dot{x} - X_\sigma(x, t)) \rangle \\
 &= \langle \tau\partial_x f(\eta), \Phi_\sigma(x, t) - X_\sigma(x, t) \rangle \\
 &= \langle \tau\partial_x f(\eta), Y_\sigma(x, t) \rangle, \tag{2.22}
 \end{aligned}$$

which means that  $Y_\sigma(x, t)$  must be tangent to  $\partial M(t)$  for each  $(x, t) \in \partial M(t)$ , since it is observed above,  $\tau\partial_x f(\eta)$  is an outward normal vector. It is obvious that  $Y_\sigma = 0$  satisfies (2.22), but equation (2.21) must also be taken care of.

The consequences of  $\partial_t F(x(t), t) = 0$  on the choice of  $Y_\sigma$  will be investigated next.

$$\begin{aligned}
 \partial_t F(x, t) &= \frac{\partial}{\partial t}f(\eta(x(t), t)) = \frac{\partial\eta}{\partial t} \cdot \partial_x f = \langle \partial_x f, \frac{\partial\eta}{\partial t} \rangle \\
 &= \langle \partial_x f(\eta), \dot{\tau}^T(x - \Gamma) + \tau^T(-\dot{\Gamma}) \rangle \\
 &= \langle \partial_x f(\eta), \tau^T[\tau\dot{\tau}^T(x - \Gamma) - \dot{\Gamma}] \rangle \\
 &= \langle \partial_x f(\eta), \tau^T[-\dot{\Gamma} - \dot{\tau}\tau^T(x - \Gamma)] \rangle.
 \end{aligned}$$

Since  $\tau\tau^T = I$  implies that  $\tau\dot{\tau}^T = -\dot{\tau}\tau^T$ , hence, it follows from (2.3) that

$$\partial_t F(x, t) = -\langle \tau(t)\partial_x f(\eta), X_\sigma(x, t) \rangle. \tag{2.23}$$

Rather than solving  $\partial_t F(x(t), t) = 0$ , it is convenient to solve the equivalent equation (assuming, of course, that the zeros of  $\partial_x f(\eta)$  are avoided, which must belong to the interior of  $M(t)$ ) obtained by dividing through by  $|\tau(t)\partial_x f(\eta)|$ :

$$P(x(t), t) = \langle N(x(t), t), X_\sigma(x(t), t) \rangle = 0 \quad (2.24)$$

It is noted that since  $\langle N, N \rangle = 1$ , differentiation totally or partially with respect to  $t$  reveals that the vectors  $\dot{N}$  and  $\partial_t N$  are both tangent to  $\partial M(t)$  for all  $0 \leq t \leq 1$ . Both of these vectors are indicators of the curvature of the sweep  $\sigma_t$ .

$$\begin{aligned} \partial_t N &= \frac{\partial}{\partial t} [|\tau(t)\partial_x f(\eta)|^{-1} \tau(t)\partial_x f(\eta)] \\ &= \frac{\partial}{\partial t} [\langle \tau(t)\partial_x f(\eta), \tau(t)\partial_x f(\eta) \rangle^{-1/2} \tau(t)\partial_x f(\eta)] \\ &= -|\tau\partial_x f(\eta)|^{-3} \langle \dot{\tau}\partial_x f(\eta) - \tau H(f)\tau^T X_\sigma, \tau\partial_x f(\eta) \rangle \tau\partial_x f(\eta) + \\ &\quad |\tau\partial_x f(\eta)|^{-1} [\dot{\tau}\partial_x f(\eta) - \tau H(f)\tau^T X_\sigma], \end{aligned}$$

where  $H(f)$  is the  $3 \times 3$  Hessian matrix of  $f$  defined by

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_3 \partial x_1} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial^2 f}{\partial x_3 \partial x_2} \\ \frac{\partial^2 f}{\partial x_1 \partial x_3} & \frac{\partial^2 f}{\partial x_2 \partial x_3} & \frac{\partial^2 f}{\partial x_3^2} \end{bmatrix};$$

a matrix that is clearly symmetric due to the equality of mixed partial derivatives.

Thus

$$\partial_t N = |\tau\partial_x f(\eta)|^{-1} (V_\sigma - \langle V_\sigma, N \rangle N), \quad (2.25)$$

where

$$V_\sigma = \dot{\tau}\partial_x f(\eta) - \tau H(f)\tau^T X_\sigma.$$

It can be seen directly from the above formula that  $\partial_t N$  is tangent to  $\partial M(t)$ .

A simple computation yields

$$\dot{N} = \frac{d}{dt} [|\tau(t)\partial_x f(\eta(x(t), t))|^{-1} \tau(t)\partial_x f(\eta(x(t), t))]$$

$$\begin{aligned}
&= \frac{d}{dt} \langle \tau(t) \partial_x f(\eta(x(t), t)), \tau(t) \partial_x f(\eta(x(t), t)) \rangle^{-1/2} \tau(t) \partial_x f(\eta(x(t), t)) \\
&= -|\tau \partial_x f(\eta)|^{-3} \langle \dot{\tau} \partial_x f(\eta) + \tau H(f) \tau^T Y_\sigma, \tau \partial_x f(\eta) \rangle \tau \partial_x f(\eta) + \\
&\quad |\tau \partial_x f(\eta)|^{-1} [\dot{\tau} \partial_x f(\eta) + \tau H(f) \tau^T Y_\sigma].
\end{aligned}$$

Hence

$$\dot{N} = |\tau \partial_x f(\eta)|^{-1} (W_\sigma - \langle W_\sigma, N \rangle N), \quad (2.26)$$

where

$$W_\sigma = \dot{\tau} \partial_x f(\eta) + \tau H(f) \tau^T Y_\sigma.$$

It can be seen directly from this formula that  $\dot{N}$  is tangent to  $\partial M(t)$  for any choice of  $Y_\sigma$ . Using the formula for  $\dot{N}$  and the definition of  $P$  in Eqn.(2.24), it is calculated that

$$\begin{aligned}
0 = \frac{d}{dt} P(x(t), t) &= \frac{d}{dt} \langle N(x(t), t), X_\sigma(x(t), t) \rangle \\
&= \langle \dot{N}(x(t), t), X_\sigma(x(t), t) \rangle + \langle N(x(t), t), \dot{X}_\sigma(x(t), t) \rangle \\
&= |\tau \partial_x f(\eta)|^{-1} \langle W_\sigma - \langle W_\sigma, N \rangle N, X_\sigma \rangle + \\
&\quad \langle N, \ddot{\Gamma} + \ddot{\tau} \tau^T (x - \Gamma) + \dot{\tau} \dot{\tau}^T (x - \Gamma) + \dot{\tau} \tau^T (\dot{x} - \dot{\xi}) \rangle.
\end{aligned}$$

Observe that  $\dot{\tau} \tau^T = -\tau \dot{\tau}^T$  implies that  $\dot{\tau}^T = -\tau^T \dot{\tau} \tau^T$ , which in turn implies that  $\dot{\tau} \dot{\tau}^T = -(\dot{\tau} \tau^T)^2$ , and so the above formula can be simplified as follows:

$$\begin{aligned}
0 = \frac{d}{dt} P(x(t), t) &= |\tau \partial_x f(\eta)|^{-1} \langle W_\sigma - \langle W_\sigma, N \rangle N, X_\sigma \rangle + \\
&\quad \langle N, \ddot{\Gamma} + \ddot{\tau} \tau^T (x - \Gamma) + \dot{\tau} \tau^T [\dot{x} - \dot{\Gamma} - \dot{\tau} \tau^T (x - \Gamma)] \rangle \\
&= |\tau \partial_x f(\eta)|^{-1} \langle W_\sigma - \langle W_\sigma, N \rangle N, X_\sigma \rangle + \\
&\quad \langle N, \ddot{\Gamma} + \ddot{\tau} \tau^T (x - \Gamma) + \dot{\tau} \tau^T Y_\sigma \rangle.
\end{aligned}$$

Then using the fact that the symmetry of the Hessian  $H(f)$  implies that

$$\langle H(f)Y, X \rangle = \langle H(f)X, Y \rangle,$$

The equation can be rewritten as (assuming  $\langle N, X_\sigma \rangle = 0$  and  $\langle N, Y_\sigma \rangle = 0$ ):

$$0 = \frac{d}{dt}P(x(t), t) = -\langle \partial_t N, Y_\sigma \rangle + \langle N, \tilde{\Gamma} + \tilde{\tau}\tau^T(x - \Gamma) - \dot{\tau}\tau^T X_\sigma \rangle, \quad (2.27)$$

Solving Eqn.(2.27) for  $Y_\sigma$ , the equation is obtained; namely,

$$\dot{x} = \Phi_\sigma(x, t) = X_\sigma(x, t) + \langle N, -\dot{\tau}\tau^T X_\sigma + \tilde{\Gamma} + \tilde{\tau}\tau^T(x - \Gamma) \rangle |\partial_t N|^{-2} \partial_t N.$$

Notice that if

$$Q = V_\sigma - \langle V_\sigma, N \rangle N, \quad (2.28)$$

the differential equation can be rewritten as

$$\dot{x} = \Phi_\sigma(x, t) = X_\sigma(x, t) + \langle \tau \partial_x f(\eta), -\dot{\tau}\tau^T X_\sigma + \tilde{\Gamma} + \tilde{\tau}\tau^T(x - \Gamma) \rangle |Q|^{-2} Q.$$

It is also easy to see that this can be rewritten in the more concise form

$$\dot{x} = \Phi_\sigma(x, t) = X_\sigma(x, t) + \langle \tau \partial_x f(\eta), \partial_t X_\sigma(x, t) \rangle |Q|^{-2} Q, \quad (2.29)$$

which shows, as expected, that the differential equation reduces to the SDE in the autonomous case when  $\partial_t X_\sigma = 0$ .

**Definition.** Equation (2.29) is called the *sweep-envelope differential equation* (SEDE) of the swept volume  $S_\sigma(M)$ , and  $\Phi_\sigma$  is called the *sweep-envelope vector field* (SEVF). Note that the SEDE reduces to the SDE when the sweep is autonomous ( $\partial_t X_\sigma = 0$ ), just as expected. To summarize, the following has been proved:

**Theorem.** The grazing set of the swept volume is foliated by the trajectories of the SEDE starting at  $\partial_0 M(0)$ .

The properties of the SEDE are illustrated in Fig.2.4, including a triangulation of the grazing set associated with the foliation by trajectories. It provides an automatic connectivity for the computed boundary points that facilitates integration with standard algorithms for visual realization and Boolean operations.

## 2.4 Trimming Methods

There are two types of trimming required in representing a swept-volume boundary based on either SDE or SEDE: local trimming and global trimming.

**Local Trimming Procedure** (Fig. 2.5) : suppose  $x$  belongs to the smooth portion of  $\partial M$  and is a grazing point at time  $t$ , then  $x$  is not a valid boundary point if for a positive and arbitrarily small  $\Delta t$

$$\langle X_\sigma(x, t - \Delta t), N \rangle > 0$$

and

$$\langle X_\sigma(x, t + \Delta t), N \rangle < 0 \quad (2.30)$$

**Global Trimming Procedure** (Fig. 2.6) : a grazing point  $x$  in the sweep  $\sigma_t(x) = \Gamma(t) + \tau(t)x$  of an object  $M = \{x : f(x) \leq 0\}$  is not a valid boundary point if  $\exists t \in [0, 1]$  such that  $f(\tau^T(x - \Gamma)) \leq 0$

The above two trimming algorithms work fairly well for two dimensional problems (Jiang, 1993; Deng et al., 1994a) and may be applied to three dimensional problems also.

## 2.5 Swept Volume for NC Simulation

The basic object  $M$  and sweep  $\sigma$  in NC simulation are defined in this section. It is assumed that the cutting tool and the stock are rigid solids.

### 2.5.1 General Tool Model and CL Data

Fig. 2.7 shows the  $\xi\zeta$  cross section of a typical 7-parameter APT tool (Kral, 1986) and local coordinate frame  $C - \xi\eta\zeta$ . The tool geometry is the revolution of this curve about the  $\zeta$  axis for  $360^\circ$ . In every APT part program, it is described by a statement which contains 7 parameters as follows:

$$CUTTER/d, r, a, b, \alpha, \beta, h$$

where

$d$  = twice the distance from tool-end to the projection on the position  $\xi$  axis of the intersection of the bottom and upper segments.

$r$  = radius of the corner circle.

$a$  =  $\xi$  coordinate value of a point at the center of the corner circle.

$b$  =  $\zeta$  coordinate value of a point at the center of the corner circle.

$\alpha$  = angle between the positive  $\xi$  axis and the bottom segment (measured in counter-clockwise direction) and in the range  $0 \leq \alpha \leq \frac{\pi}{2}$ .

$\beta$  = the angle between the positive  $\zeta$  axis and the upper segment (measured in clockwise direction) and in the range  $-\frac{\pi}{2} < \beta < \frac{\pi}{2}$ .

$h$  = the height of the tool (measured from the tool-end along the positive  $\zeta$  axis).

$C$  is called the tool control point (TCP). Generally, a general 7-parameter APT tool is composed of four faces: the lower conical face, toroidal face, upper conical face and top face.

The 7-parameters  $d, r, a, b, \alpha, \beta, h$  are not independent in terms of the need for geometry representation. But they are necessary for the classification of different tools. There is a wide range of tool geometries that can be described by the above seven parameters. Fig. 2.8 shows some of the tool geometries which can be represented by the 7-parameter APT tool model. For example:

(i) When  $r = 0, a = 0, b = 0, \alpha = \beta = 0$ , it represents a flat-end cylindrical tool (Fig. 2.8). An example is given by Cutter/4,0,0,0,0,0,7. Only two parameters are needed to define the geometry of the tool. The other five parameters are used to identify it as a flat-end tool.

(ii) When  $r = b = d/2, a = 0, \alpha = \beta = 0$ , it represents a ball-end cylindrical tool (Fig. 2.8). An example is given by Cutter/4,2,0,2,0,0,7. Two parameters are needed to define the geometry of this tool.



(iii) When  $r = a = b = d/4$ ,  $\alpha = \beta = 0$ , it represents a fillet-end tool (Fig. 2.8). An example is given by Cutter/4,1,1,1,0,0,7.

In NC simulation or verification, Cutter Location (CL) Data is defined as

$$\text{CL} = (x_c, y_c, z_c, i, j, k)$$

where  $(x_c, y_c, z_c)$  (called the control point) are the coordinates of tool end  $C$  in the reference frame.  $i, j, k$  are the normalized directional cosines of the moving tool axis ( $\zeta$  - axis) in the reference coordinate frame. The CL data defines the tool tip location and tool orientation in the workpiece coordinate frame. Since the cutting tool is axisymmetric, CL data completely defines the tool configuration.

### 2.5.2 Generalized Kinematics Model of Five-axis Milling Machine

Machine Control Data (MCD) for a 5-axis milling machine is defined as

$$\text{MCD} = (X, Y, Z, A, B)$$

where  $X, Y$  and  $Z$  represent three translational axes, and  $A$  and  $B$  are two rotational axes. Considering only the possible arrangements of the rotation axes, commonly used five-axis milling machines are characterized into three main families (Leslie, 1970) which are illustrated in Fig. 2.9:

**Type A** Machines with a rotary table and a tilting spindle head, rotation of the table and head being restricted to one plane.

**Type B** Machines with a fixed spindle head and a table capable of rotation in two perpendicular planes, e.g., the FADAL VMC-20 vertical milling machine.

**Type C** Machines with a fixed table and a spindle head capable of rotation in two perpendicular planes.

Certain types of five-axis machines may have different configurations with different tilting and rotation directions (Fig. 2.10). For example, the FADAL VMC-20 vertical milling machine has at least four configurations. Taking both translational

and rotational axes into account, there are 12 possible arrangements: 3 for rotation axes and 4 for the translation axes (Ruegg, 1992).

The possible arrangements for the rotational and translational axes, tool and workpiece form the basis of a generalized kinematics model. The general machine can be specified as a sequence of transformations of coordinates.

### 2.5.3 Five-axis Motion

The motion of a multi-axis NC milling machine can be assumed generally that each axis has a constant speed and all the axes start and end at the same time, i.e. the normal form of operation is joint-interpolated motion. For a five-axis machine, both the position of the control point and the orientation of the tool-axis have to be interpolated. The machine displacements are assumed to be linearly interpolated between two successive configurations of the machine, called MCD interpolation, i.e.,

$$\begin{pmatrix} X(t) \\ Y(t) \\ Z(t) \\ A(t) \\ B(t) \end{pmatrix} = \begin{pmatrix} X(0) \\ Y(0) \\ Z(0) \\ A(0) \\ B(0) \end{pmatrix} + t \begin{pmatrix} X(1) - X(0) \\ Y(1) - Y(0) \\ Z(1) - Z(0) \\ A(1) - A(0) \\ B(1) - B(0) \end{pmatrix} \quad (2.31)$$

where  $X(0), Y(0), Z(0), A(0), B(0)$  and  $X(1), Y(1), Z(1), A(1), B(1)$  represent the initial and final configurations of the machine respectively, and  $t \in [0, 1]$  is the normalized time variable.

The motion from one configuration to the next configuration by such an interpolation will be referred to as *MCD interpolation based five-axis motion* in our study. Apart from the linear interpolator, circular, parabolic and cubic interpolators are also available in advanced systems (Pressman, 1977).

In NC simulation, the linear interpolation of five-axis motion between two tool configurations may be approximated mathematically based on CL interpolation, i.e.

$$\begin{pmatrix} x_c(t) \\ y_c(t) \\ z_c(t) \\ i(t) \\ j(t) \\ k(t) \end{pmatrix} = \begin{pmatrix} x_c(0) \\ y_c(0) \\ z_c(0) \\ i(0) \\ j(0) \\ k(0) \end{pmatrix} + t \begin{pmatrix} x_c(1) - x_c(0) \\ y_c(1) - y_c(0) \\ z_c(1) - z_c(0) \\ i(1) - i(0) \\ j(1) - j(0) \\ k(1) - k(0) \end{pmatrix} \quad (2.32)$$

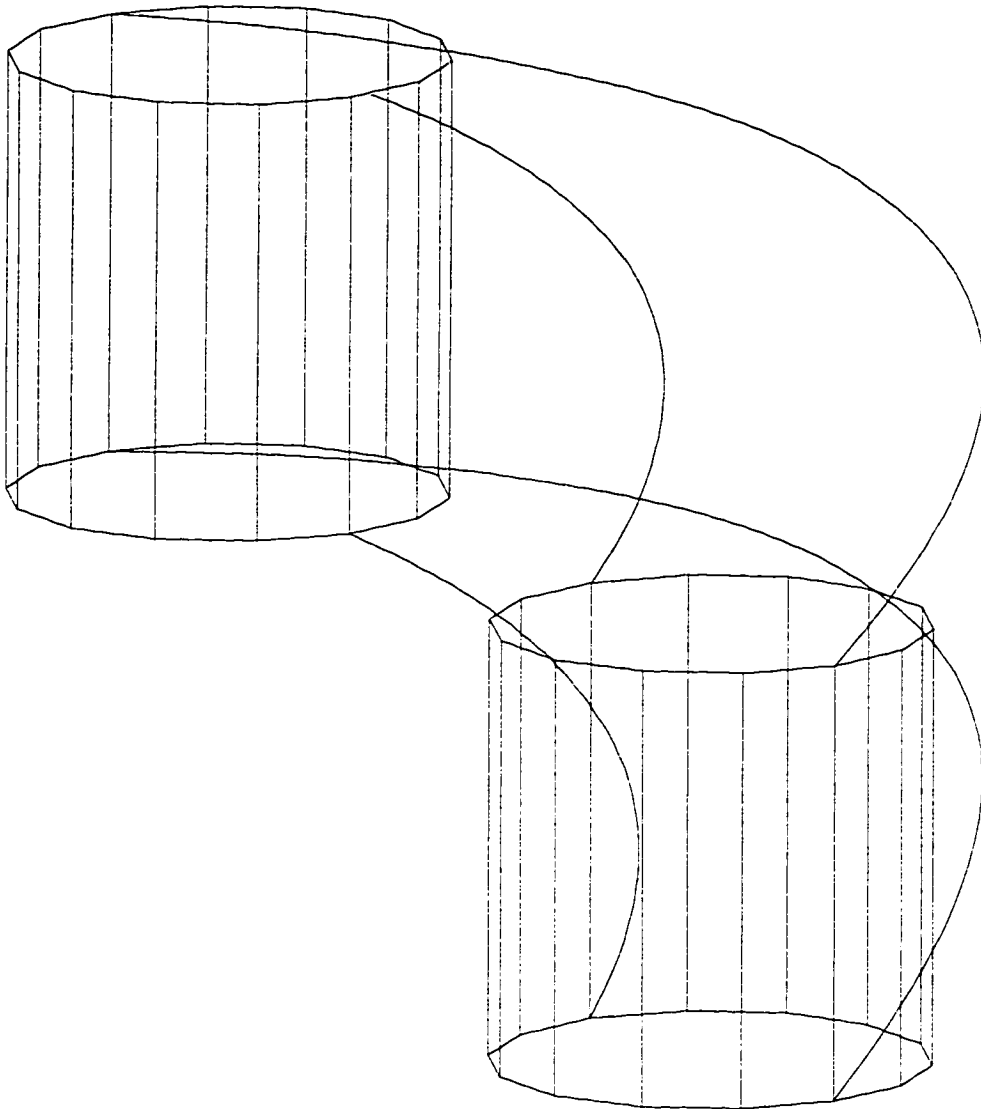
where  $x_c(0), y_c(0), z_c(0), i(0), j(0), k(0)$  and  $x_c(1), y_c(1), z_c(1), i(1), j(1), k(1)$  represent the initial and final configurations of the tool respectively, and  $t \in [0, 1]$  is the normalized time variable. This interpolation is referred to as *CL interpolation based five-axis motion*. This method does not involve the machine configuration and is convenient in applications.

Sweep equations of five-axis motion may be derived by conducting a kinematics analysis (Appendix A) and using either CL interpolation or MCD interpolation. It will be illustrated in the following chapters.

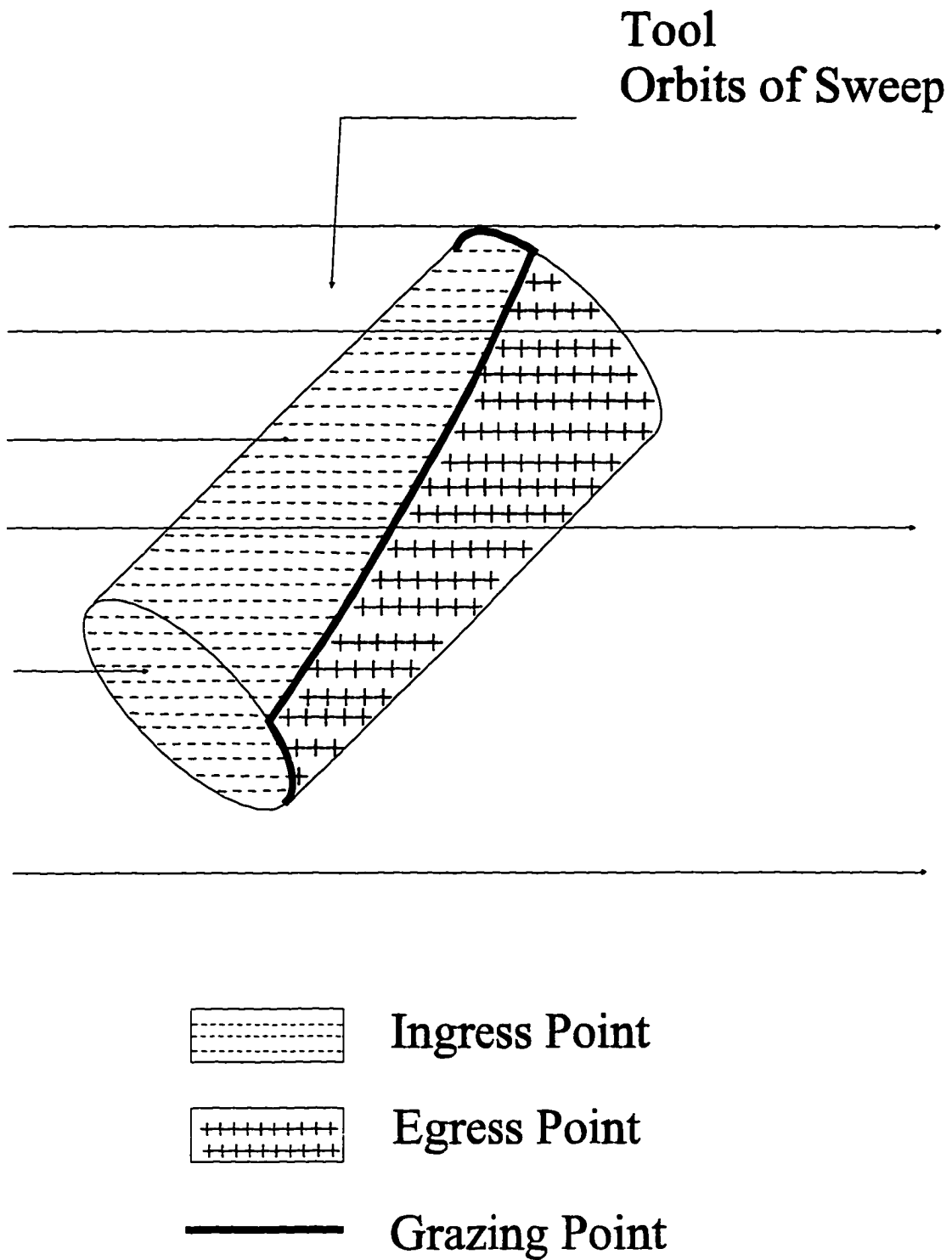
## 2.6 Remarks

In this chapter, some basic concepts related to swept volume such as smooth  $M$ ,  $\partial M$ , smooth sweep  $\sigma$ ,  $\sigma_t(x)$  and swept volume  $S_\sigma(M)$  have been summarized. The basic ideas of SDE and BFF analysis: SVF, tangency function, decomposition of a moving object, ingress point, egress point, grazing point, grazing set, candidate set and trimming set are described in detail. These, although already available in previous work, are necessary to proceed the generation of the new SEDE method. The SEDE equation is derived step by step. Trimming concepts and algorithms, which are important in SDE and SEDE study to obtain exact boundary, are described.

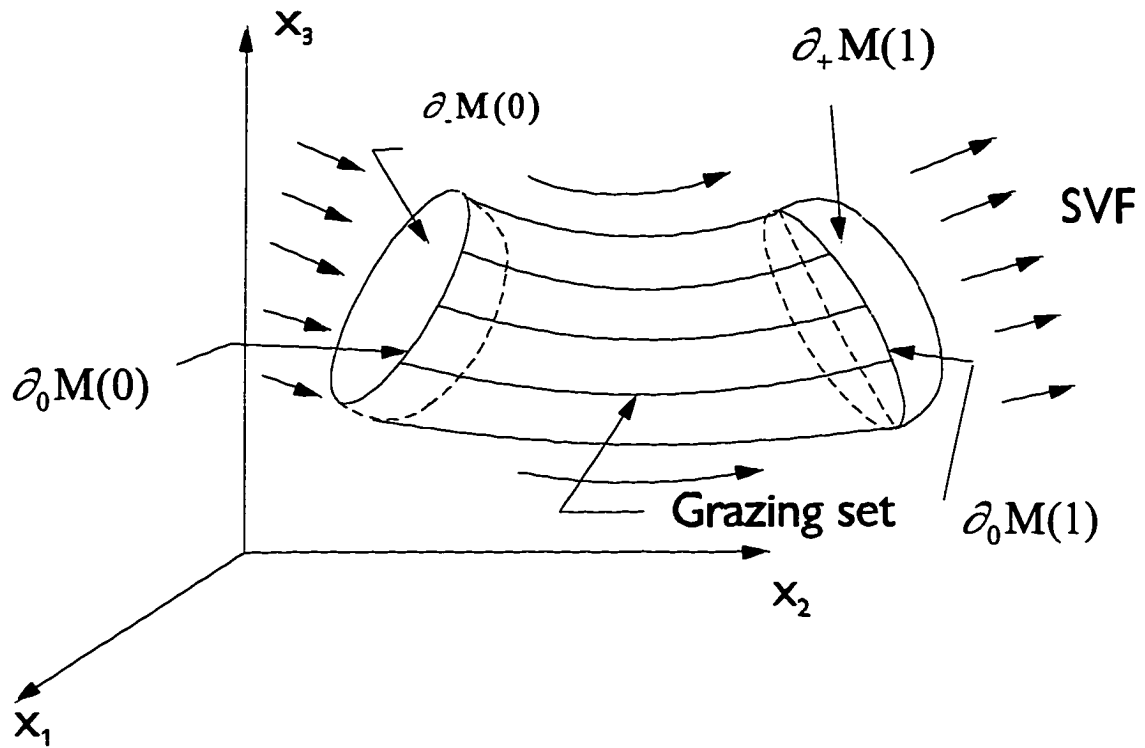
Having modeled swept volumes mathematically and defined the tool geometry and five-axis motion, the following chapters describe how the swept volumes for given tool model and tool motion are computed using the SDE/SEDE methods and how they are applied in NC machining problems.



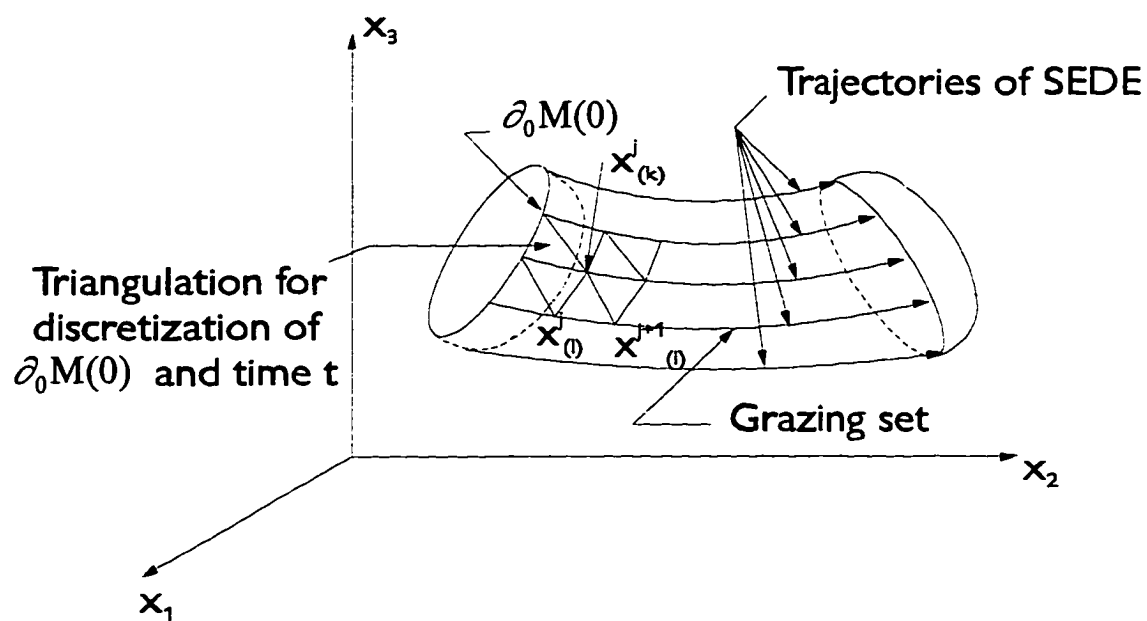
**Figure 2.1** A typical swept volume of an object in space



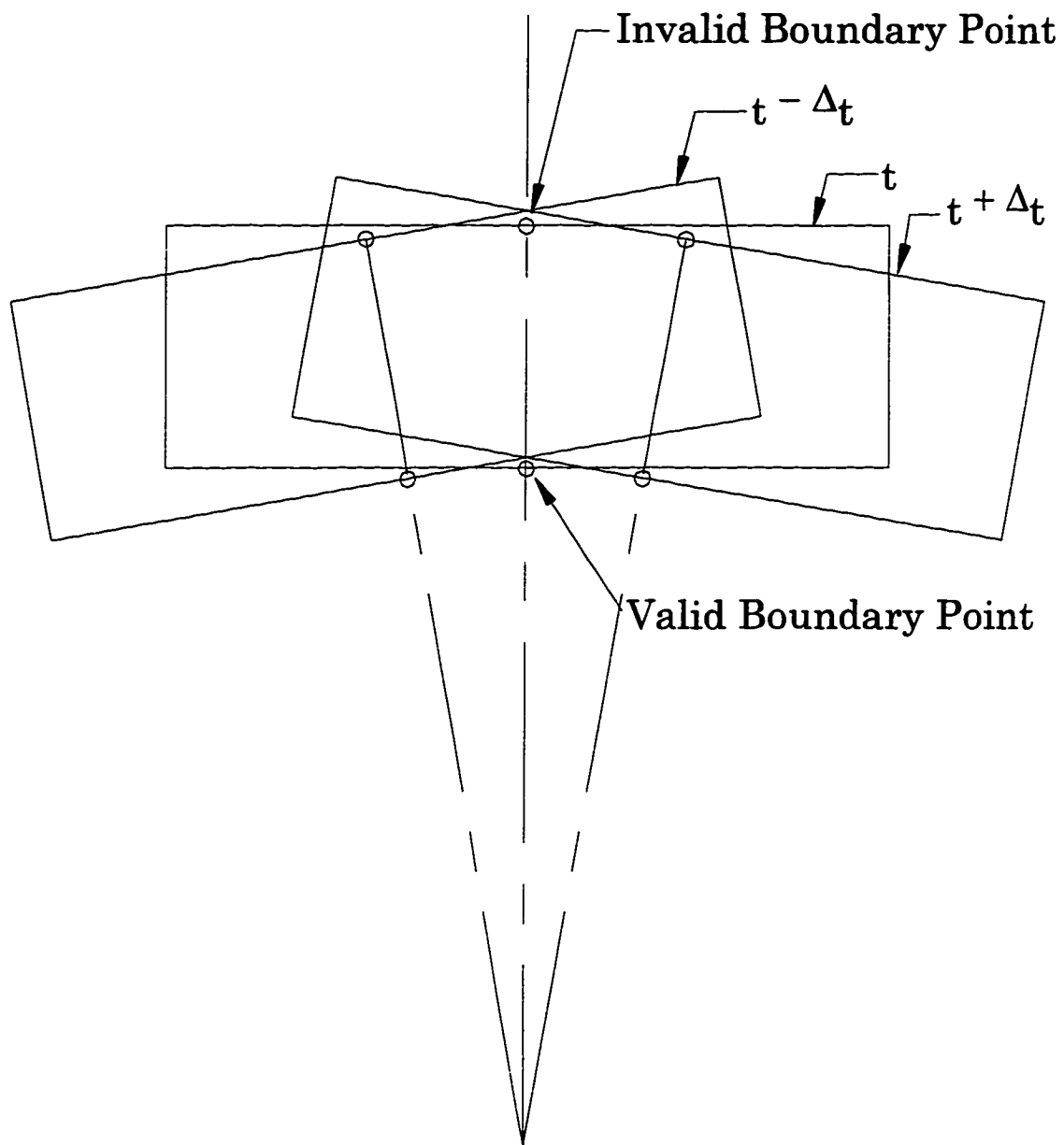
**Figure 2.2** Decomposing object boundary



**Figure 2.3** Swept volume orbits of SDE

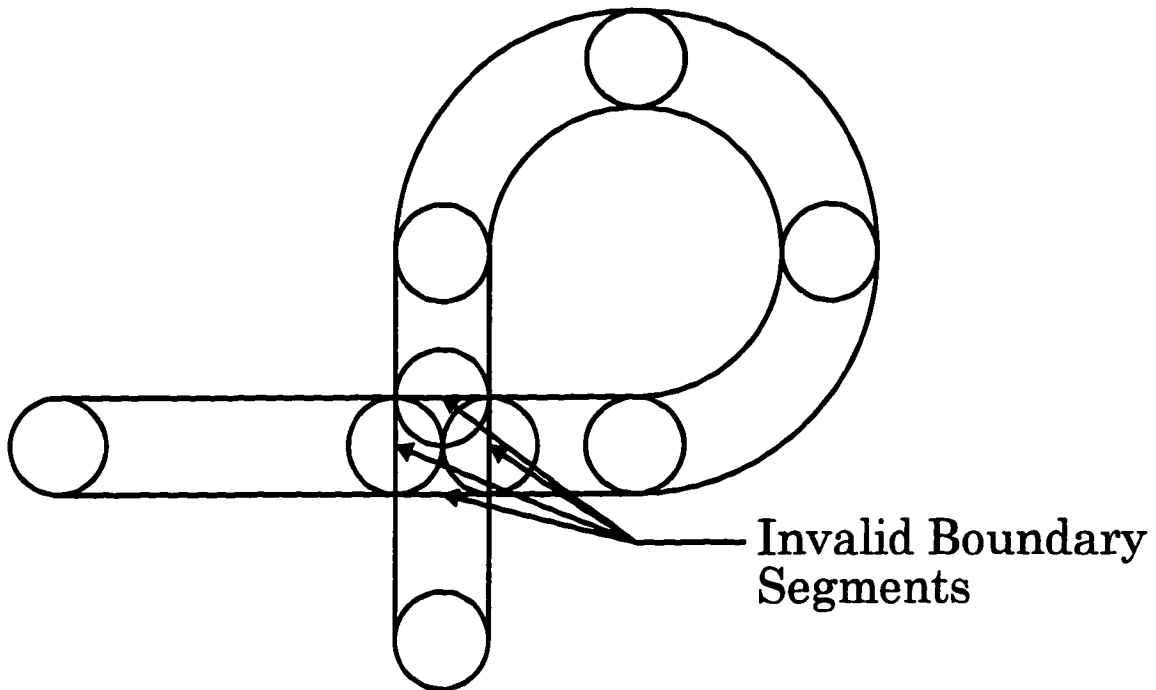


**Figure 2.4** Swept volume orbits of SEDE



**Figure 2.5** Local trimming procedure





**Figure 2.6** Global trimming procedure

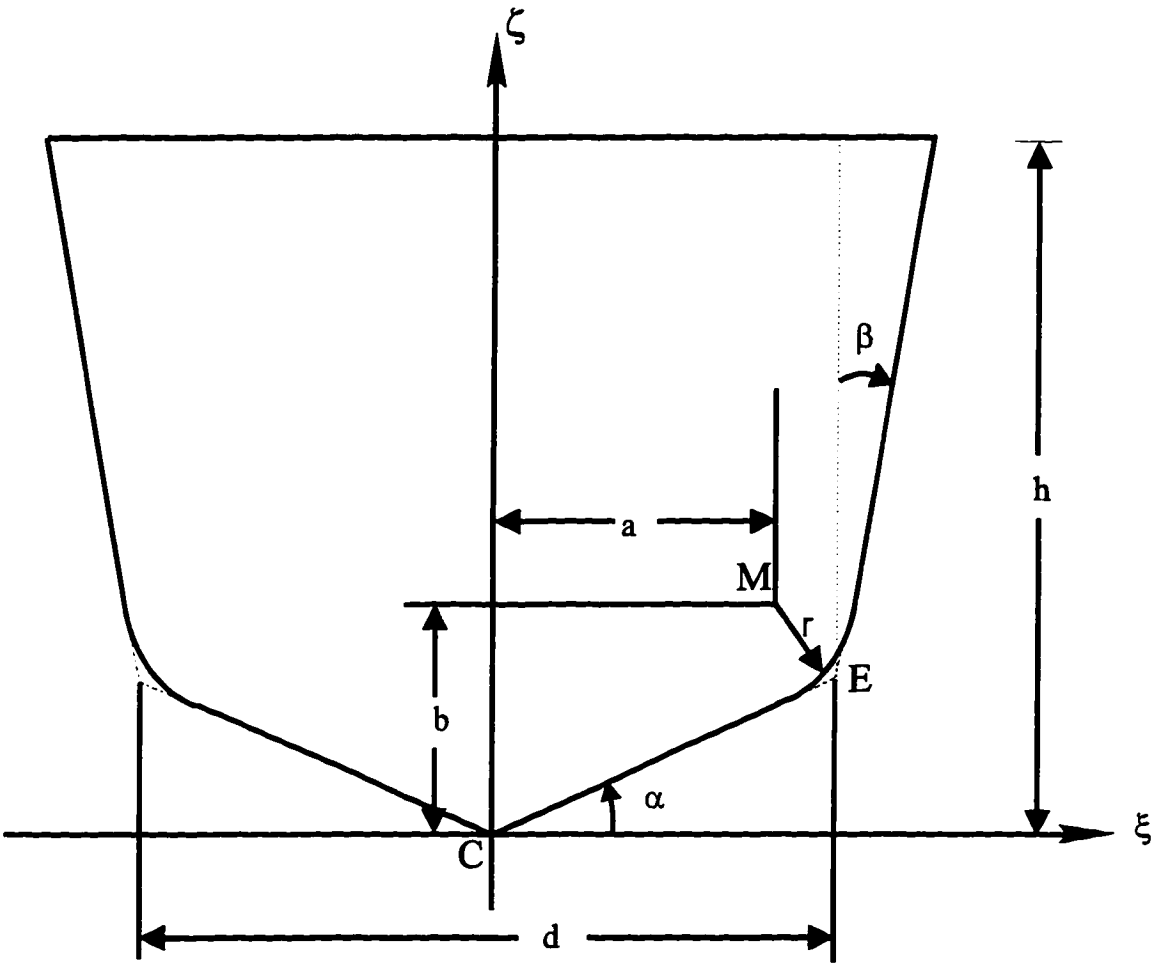

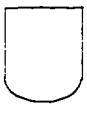
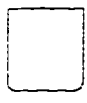
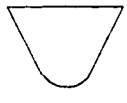
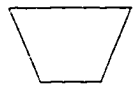
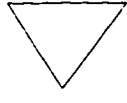
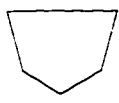



Figure 2.7 General 7-parameter APT tool

	Cross-section	Description	d	r	a	b	$\alpha$	$\beta$	h
1		flat-end	d	0	0	0	0	0	h
2		ball-end	d	d/2	0	d/2	0	0	h
3		fillet-end	d	d/4	d/4	d/4	0	0	h
4		ball-conical	d	r	0	r	0	$\beta$	h
5		flat-conical	d	0	0	0	0	$\beta$	h
6		cone	d	0	0	0	$\alpha$	0	h
7		conical-end	d	0	0	0	$\alpha$	$\beta$	h
8		general	d	r	a	b	$\alpha$	$\beta$	h

**Figure 2.8** Some tool geometries represented by 7-parameter APT tool model

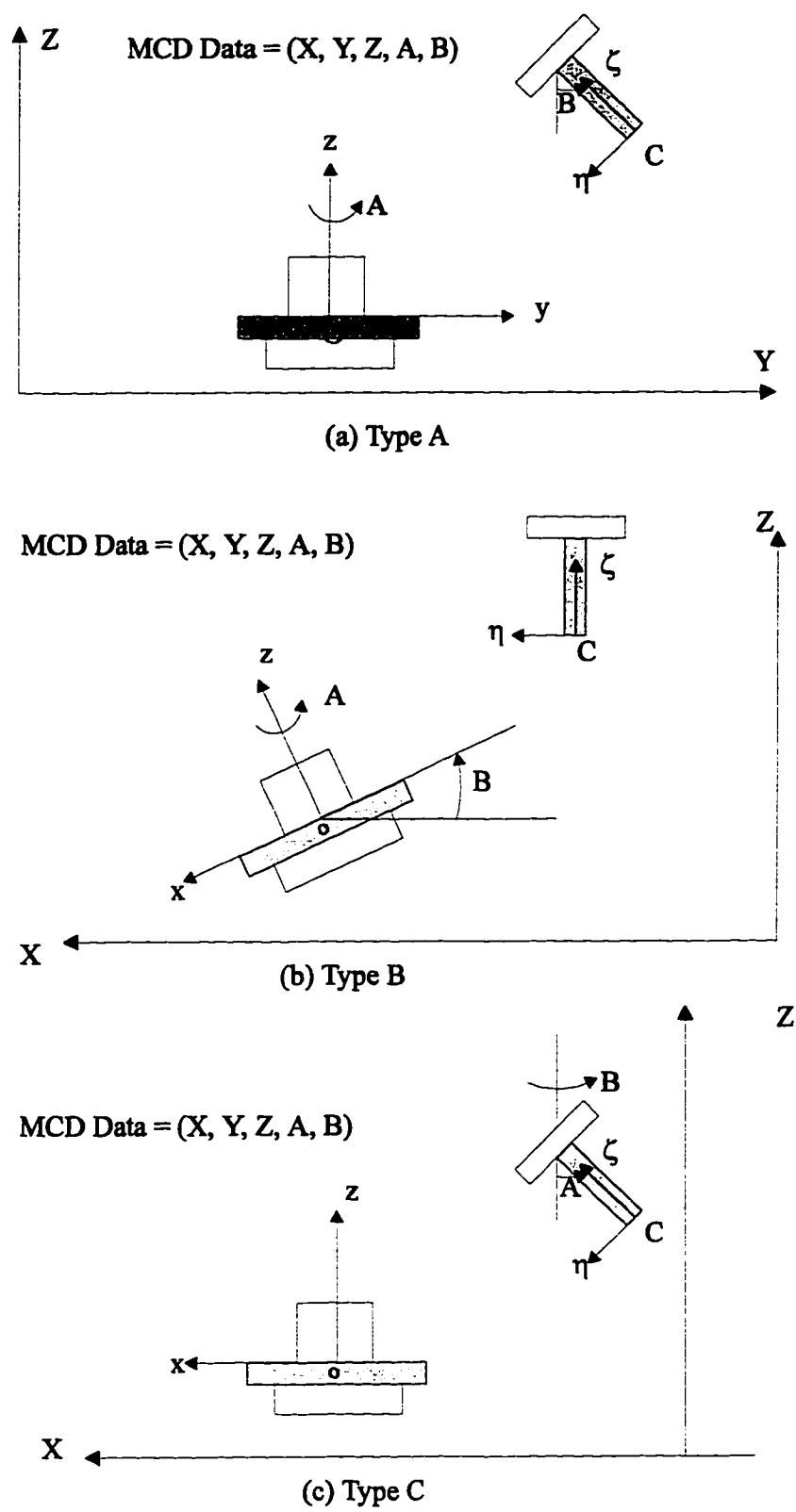
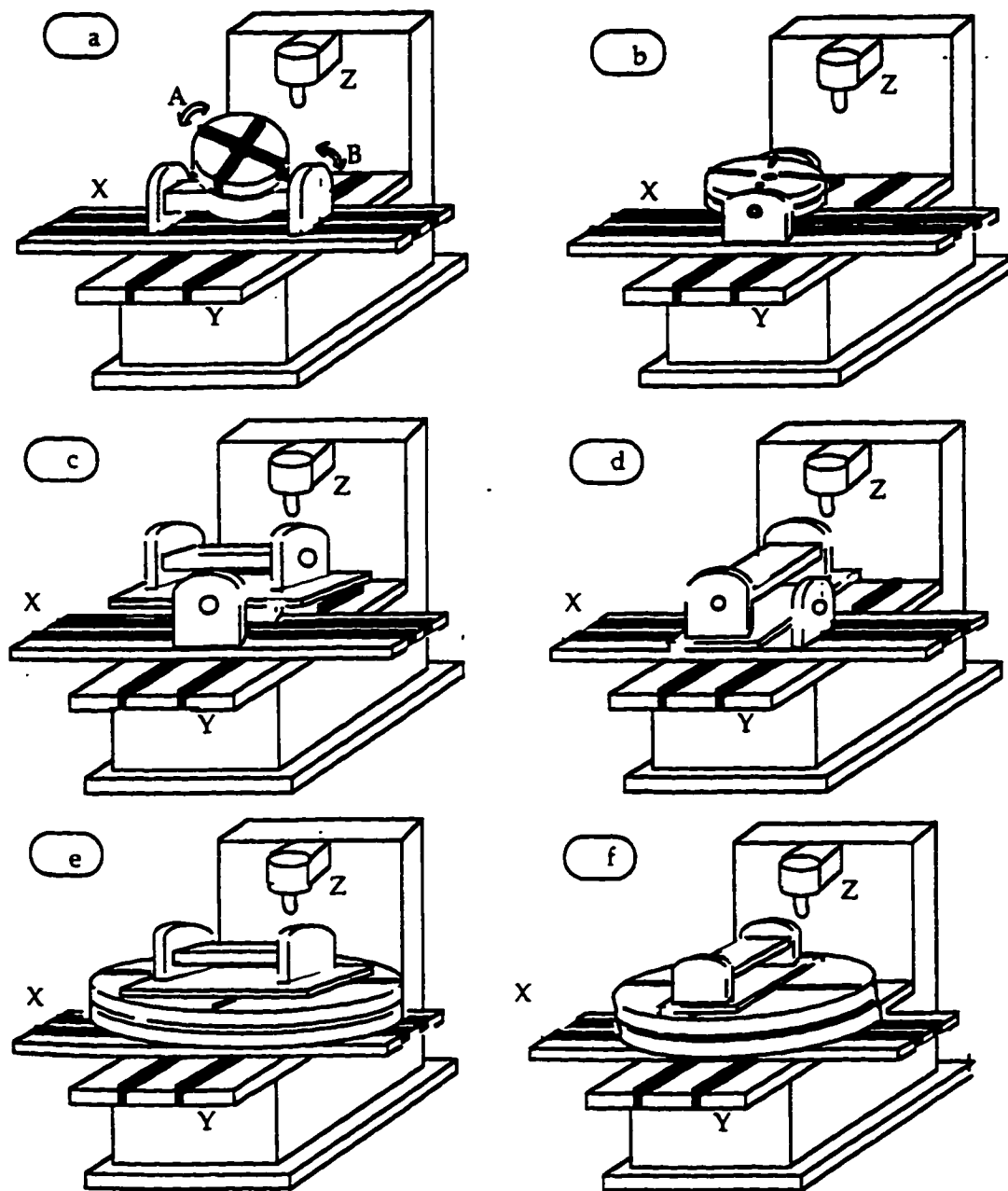


Figure 2.9 Three types of five-axis NC machines and MCD data



**Figure 2.10** Six kinds of configurations for Type B five-axis machines

## CHAPTER 3

### SDE APPROACH TO MOVING TOOL IN FIVE-AXIS MOTION

In this chapter, a sweep generator is developed using the SDE method for simple tool geometries (flat-end and ball-end) and a type A machine undergoing general MCD interpolation based 5-axis NC milling motion. An algorithm is developed and a software program called SDE-3D is written for the computation and display of the tool swept volumes. To obtain this, the tool geometry and motion have to be described parametrically.

#### 3.1 Models of the Machine and Cutter

Fig. 3.1 shows a typical five-axis NC milling machine with the reference coordinate frame  $xyz$ . The origin of the reference coordinate frame ( $O$ ) is fixed at the center of the top of the workpiece platform. The machine has three translational motions ( $X$ ,  $Y$  and  $Z$ ) and two rotational motions ( $A$  and  $B$ ).

Fig. 3.2 shows the tool geometry and moving tool coordinate frame  $\xi\eta\zeta$ . Commonly used ball-end and flat-end tools are considered here. Point  $C(x_c, y_c, z_c)$  is called the control point or the tool center point. The Cutter Location Data (CL Data) are defined by  $(x_c, y_c, z_c, i, j, k)$ , where  $i, j, k$  are the directional cosines of the moving tool frame relative to the  $xyz$  coordinate frame. The CL data and tool length  $L$  and radius  $R$  are input parameters in the SDE-3D computer program.

Fig. 3.3 shows the relationship of the coordinate frames. To define tool motion, CL data and parameters of the machine geometry are needed. Some researchers used only CL data as inputs and they generated the tool motion by linearly interpolating the two successive CL data inputs. The advantage in doing this is that the verification system is simpler to build. However, the tool path between the initial and final locations depends on the machine geometry. Two

different machines may produce two different tool paths even if the CL data are the same.

The transformation of CL data to MCD data can be done using geometrical properties of the machine. Fig. 3.4 shows the relations between the translational displacement of the machine and the reference coordinate frame, and between the rotational displacement and the generalized directional cosines. From this figure, the relation between the MCD data represented by  $(X, Y, Z, A, B)$  and the CL data represented by  $(x_c, y_c, z_c, i, j, k)$  can be derived as

$$\begin{aligned} X &= -x_c \cdot \cos A + y_c \cdot \sin A \\ Y &= -x_c \cdot \sin A - y_c \cdot \cos A + L \cdot \sqrt{1 - K^2} \\ Z &= z_c + L \cdot K \\ A &= \tan^{-1} \frac{I}{J} \\ B &= \tan^{-1} \frac{\sqrt{1 - K^2}}{K} \end{aligned}$$

where

$$I = \frac{i}{\sqrt{i^2 + j^2 + k^2}}, \quad J = \frac{j}{\sqrt{i^2 + j^2 + k^2}}, \quad K = \frac{k}{\sqrt{i^2 + j^2 + k^2}},$$

and  $(I, J, K)$  represents the normalized directional cosines.  $L$  is the tool length and  $x_c, y_c, z_c$  are the coordinates of the tool control point  $C$ .

### 3.2 Sweep Equation

For the five-axis NC milling machine shown in Fig. 3.1, the sweep equation of the moving tool can be described as follows.

Fig. 3.5 shows the Denavit-Hartenberg representation for the machine. The origin of the tool coordinate frame  $\xi\eta\zeta$  initially coincides with the origin of the reference coordinate frame  $xyz$ . The Denavit-Hartenberg transformation matrix for adjacent coordinate frames,  $x_i, y_i, z_i$  and  $x_{i-1}, y_{i-1}, z_{i-1}$  is

$$A_i = \begin{pmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.1)$$

By multiplying the transformation matrices of all links, the transformation matrix between the tool coordinate frame and the reference coordinate frame is obtained

$$T_i = \begin{pmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = A_1 A_2 \dots A_5 A_t \quad (3.2)$$

where

$$A_t = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & L \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.3)$$

Separating the transformation matrix into a translational component and a rotational component and using the regular (instead of homogeneous) representation, the equation representing the trajectory of any generic point in the tool coordinate frame is obtained as follows:

$$\mathbf{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} + \begin{pmatrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{pmatrix} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix} \quad (3.4)$$

where the vector  $\mathbf{r} = (x, y, z)^T$  represents any point in the reference coordinate frame  $xyz$  and  $\mathbf{r}' = (\xi, \eta, \zeta)^T$  represents any point in the tool coordinate frame  $\xi\eta\zeta$ . These vectors are shown in Fig. 3.3. For this particular machine, the sweep equation is:

$$\begin{aligned} \mathbf{r} &= \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -X \cos A - Y \sin A + L \sin A \sin B \\ X \sin A - Y \cos A + L \cos A \sin B \\ Z - L \cos B \end{pmatrix} \\ &+ \begin{pmatrix} \cos A & \sin A \cos B & -\sin A \sin B \\ -\sin A & \cos A \cos B & -\cos A \sin B \\ 0 & \sin B & \cos B \end{pmatrix} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix} \end{aligned} \quad (3.5)$$



### 3.3 Sweep Vector Field

By differentiating Eqn.(3.5) with respect to  $t$ , the sweep vector field of the moving tool is obtained:

$$\dot{\mathbf{r}} = V_\sigma = [u, v, w]^T \quad (3.6)$$

where

$$u = -\dot{X} \cdot \cos A - \dot{Y} \cdot \sin A + \dot{A}(\sin A \cdot X - \cos A \cdot Y) - \dot{A} \cdot \sin A \cdot \xi + (\cos A \cos B \cdot \dot{A} - \sin A \sin B \cdot \dot{B})\eta + (\sin A \cos B \cdot \dot{B} + \cos A \sin B \cdot \dot{A})(L - \zeta),$$

$$v = \dot{X} \cdot \sin A - \dot{Y} \cdot \cos A + \dot{A}(\cos A \cdot X + \sin A \cdot Y) - \dot{A} \cdot \cos A \cdot \xi - (\sin A \cos B \cdot \dot{A} + \cos A \sin B \cdot \dot{B})\eta - (\sin A \sin B \cdot \dot{A} - \cos A \cos B \cdot \dot{B})(L - \zeta),$$

$$w = \dot{Z} + \dot{B} \cdot (L \cdot \sin B + \cos B \cdot \eta - \sin B \cdot \zeta).$$

### 3.4 Tangency Function and Boundary Points

Let  $N_1, N_2, N_3$  and  $N_4$  denote the outer normal vectors of the ball-end and flat-end tools as shown in Fig. 3.2. They correspond to the top surface, cylindrical surface, bottom surface of the flat-end tool and the spherical surface of the ball-end tool, respectively. The tangency functions for these smooth parts of the tool are of the following form:

The top surface of either the ball-end or flat-end tool:

$$T_1 = \langle V_\sigma, N_1 \rangle = a_1(t)\xi + a_2(t)\eta + a_3(t) \quad (3.7)$$

where

$$a_1(t) = \dot{A} \sin B,$$

$$a_2(t) = \dot{B},$$

$$a_3(t) = \dot{Z} \cos B + \dot{Y} \sin B - \dot{A}X \sin B.$$

The cylindrical surface of either the ball-end or flat-end tool:

$$T_2 = \langle V_\sigma, N_2 \rangle = b_1(t)\xi + b_2(t)\eta + b_3(t)\xi\zeta + b_4(t)\eta\zeta \quad (3.8)$$

where

$$\begin{aligned} b_1(t) &= -\dot{X} - \dot{A}Y + \dot{A}L \sin B, \\ b_2(t) &= \dot{Z} \sin B - \dot{Y} \cos B + \dot{B}L + \dot{A}X \cos B, \\ b_3(t) &= -\dot{A} \sin B, \\ b_4(t) &= -\dot{B}. \end{aligned}$$

The bottom surface of the flat-end tool:

$$T_3 = \langle V_\sigma, N_3 \rangle = c_1(t)\xi + c_2(t)\eta + c_3(t) \quad (3.9)$$

where

$$\begin{aligned} c_1(t) &= -\dot{A} \sin B, \\ c_2(t) &= -\dot{B}, \\ c_3(t) &= -\dot{Z} \sin B - \dot{Y} \sin B + \dot{A}X \sin B. \end{aligned}$$

The spherical surface of the ball-end tool:

$$T_4 = \langle V_\sigma, N_4 \rangle = d_1(t)\xi + d_2(t)\eta + d_3(t)\zeta \quad (3.10)$$

where

$$\begin{aligned} d_1(t) &= -\dot{X} + \dot{A}L \sin B - \dot{A}Y, \\ d_2(t) &= -\dot{Z} \sin B - \dot{Y} \cos B + \dot{A}X \sin B + \dot{B}L, \\ d_3(t) &= -\dot{Z} \cos B + \dot{Y} \sin B - \dot{A}X \sin B. \end{aligned}$$

The candidate sets of points for the boundary of the tool swept volume that are due to the smooth portion of the tool can be obtained by letting the above tangency functions equal to zero.

To obtain the candidate sets of points due to the edge of two contiguous faces of  $\partial M$ , two outer normal vectors, say  $N_i$  and  $N_j$ , can be assigned for each of the

abutting smooth faces, and the candidate points which satisfy the following relation can be calculated:

$$\langle V_\sigma, N_i \rangle \cdot \langle V_\sigma, N_j \rangle \leq 0 \quad (3.11)$$

### 3.5 Boundary Trimming and Graphic Display of Swept Volume

As described in Chapter 2, some of the grazing points calculated are not actual boundary points of the swept volume and should be trimmed off. H. Jiang (Jiang, 1993) described local and global trimming procedures which are efficient for the sweeping of polygonal objects in a 2-D space (i.e. plane). The trimming process in the 3D case is the same in principle; however, it is computationally much more demanding. The local trimming procedure is applied during the process of generating grazing points. The global trimming procedure is applied after all the grazing points have been generated.

The trimming procedure can also be done in the image space using techniques in computer graphics. The surface patches are formed and displayed by the candidate boundary points calculated at successive times, with the hidden surfaces removed, using the HOOPS Graphics System in the X-window environment on a Sun workstation. This can be considered as a trimming process done in the image space. Since the candidate patches contain the total boundary of the swept volume, and the hidden surfaces are removed in the display, the image generated is the same as that of the actual swept volume boundary. It will be shown in an example that the candidate patches displayed indeed contain the boundary of the swept volume and thus serve as a visual proof of the correctness and validity of the SDE method.

The points on the swept-volume boundary for ball-end and flat-end cutters can be partitioned into six parts: (1) ingress points at time  $t = 0$ , (2) grazing points on the cylindrical face for  $0 < t < 1$ ; (3) grazing points on the spherical face for

$0 < t < 1$ ; (4) grazing points on the planar face for  $0 < t < 1$ ; (5) grazing points on the edge for  $0 < t < 1$ ; (6) egress points at time  $t = 1$ .

### 3.6 Development of Computer Program

Using the SDE approach for cutter swept volume representation described above, a software program called SDE-3D has been developed to calculate and display the cutter swept volume. Fig. 3.6 shows the flow chart of the computer program. The procedure is as follows:

**P<sub>1</sub> Input of data.** Input the tool geometry ( $L$  and  $R$ ), tool initial position  $((x_c, y_c, z_c, i, j, k)|_{t=0})$  and final position  $((x_c, y_c, z_c, i, j, k)|_{t=1})$ .

**P<sub>2</sub> Transformation of CL data to MCD data.**

**P<sub>3</sub> Calculation of grazing points.** Discretize the normalized time  $t=[0,1]$  into  $N$  intervals. At each  $t_i = i/N$  ( $i = 1, \dots, N - 1$ ), calculate the subset of the grazing points by checking the tangency function for the different parts of the tool: cylindrical face, ball face, planar faces and edges.

**P<sub>4</sub> Calculation of ingress and egress points.** Discretize the tool surface along its  $\zeta$  axis into  $N_\zeta$  circular sections. For each section, discretize the circular curve into  $N_s$  points. At each discretized point, calculate the ingress and egress points.

**P<sub>5</sub> Output of data for display.** Display the tool swept volumes by decomposition of the swept volume into patches for the ingress, egress, and grazing points. It should be noted that the trimming process of candidate boundary points has yet to be incorporated into the software, i.e., computing  $\partial S_\sigma(M)$  instead of displaying all subsets of  $C_\sigma(M)$ .

The SDE-3D program is implemented using C language and Hoops Graphical System under the Unix operating system on an SPARC 10 Sun Workstation.

### 3.7 Illustrative Example

For the ball-end tool shown in Fig. 3.2 with  $L=4$  and  $R=1$ , the tool initial position is  $(0,0,0,0,0,1)$  and its final position is  $(0,1,0,0,-1,1)$ . Fig. 3.7 shows the swept volume and some of the surface patches.

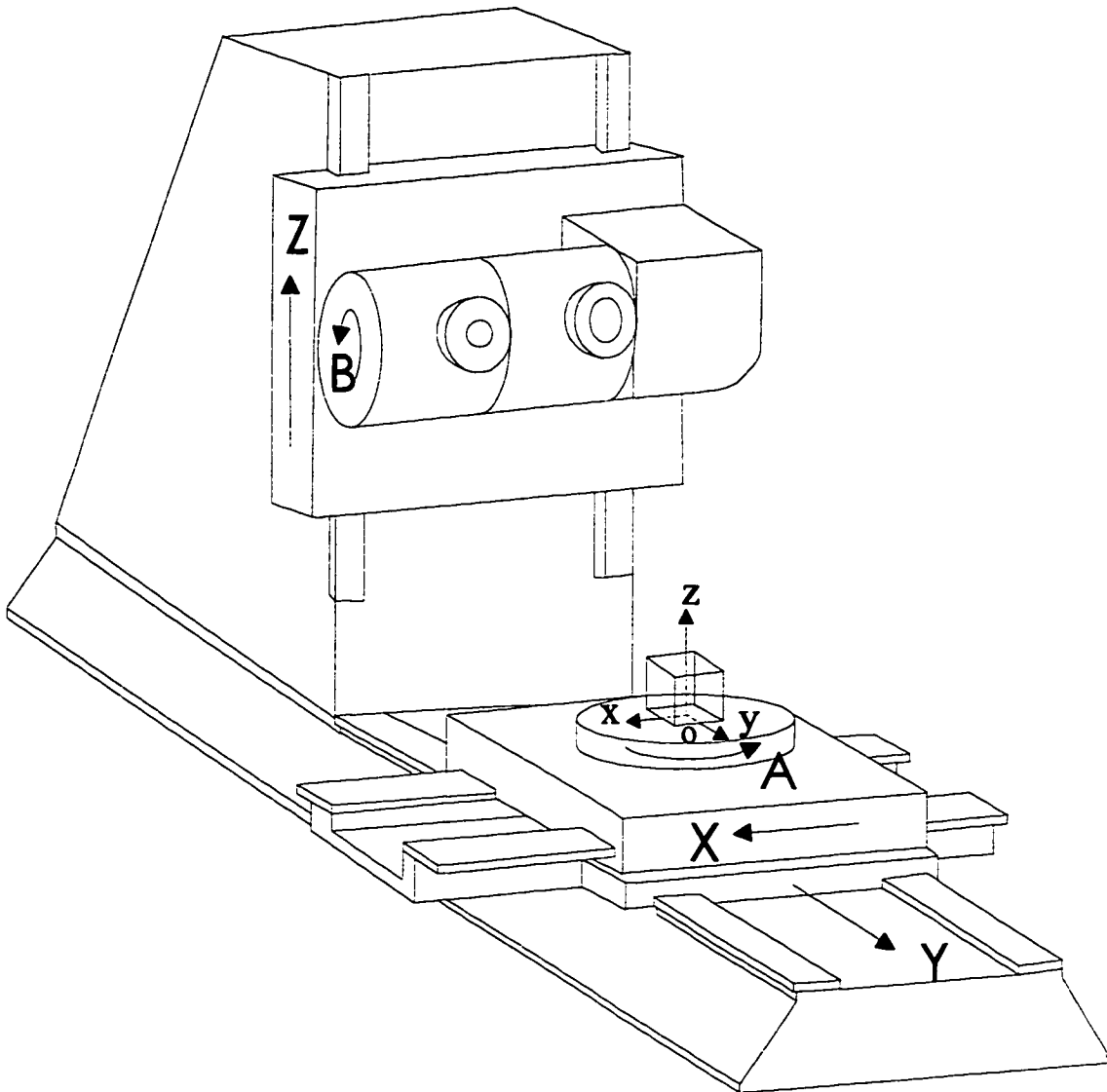
### 3.8 Final Remarks

By means of the SDE method, an algorithm for the calculation and display of boundary points of the cutter swept volume for a typical 5-axis NC milling machine has been developed. The points on the boundary of the cutter swept volume are shown to be a subset of the union of 1) the grazing points of the cutter for the entire sweep, 2) the ingress points of the cutter at the beginning of the sweep, and 3) the egress points of the cutter at the end of the sweep. The grazing points at each time instant are computed from a tangency function which describes the relationship between the sweep vector and the outward surface normals for all the points on the boundary of the cutter.

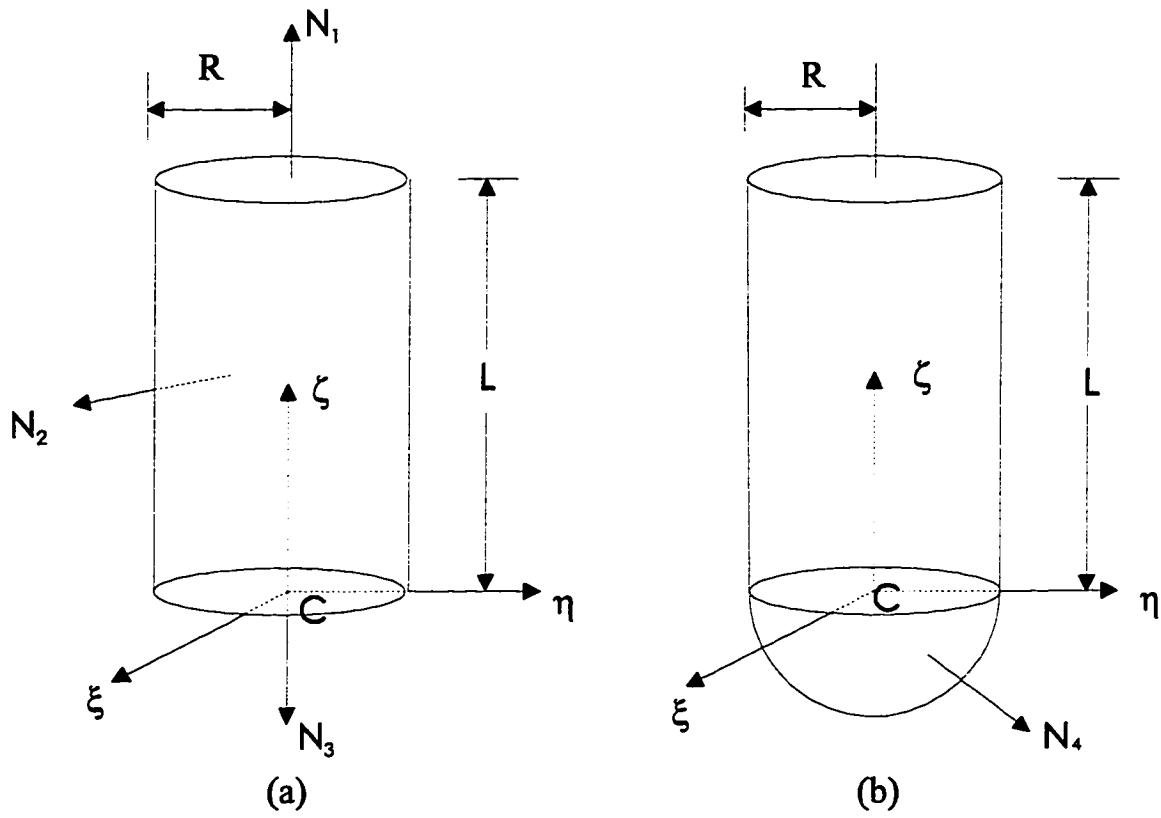
A machine control data based interpolation method is uniquely developed to describe the interpolation motion equation of a five-axis NC milling tool and applied to NC simulation and verification.

The work presented forms a basis of the research toward simulation and verification of NC programs for 5-axis milling. In Chapter 5, an interface with state-of-the-art CAD/CAM systems will be developed to generate a useful application of the swept volume research on practical computer NC verification.

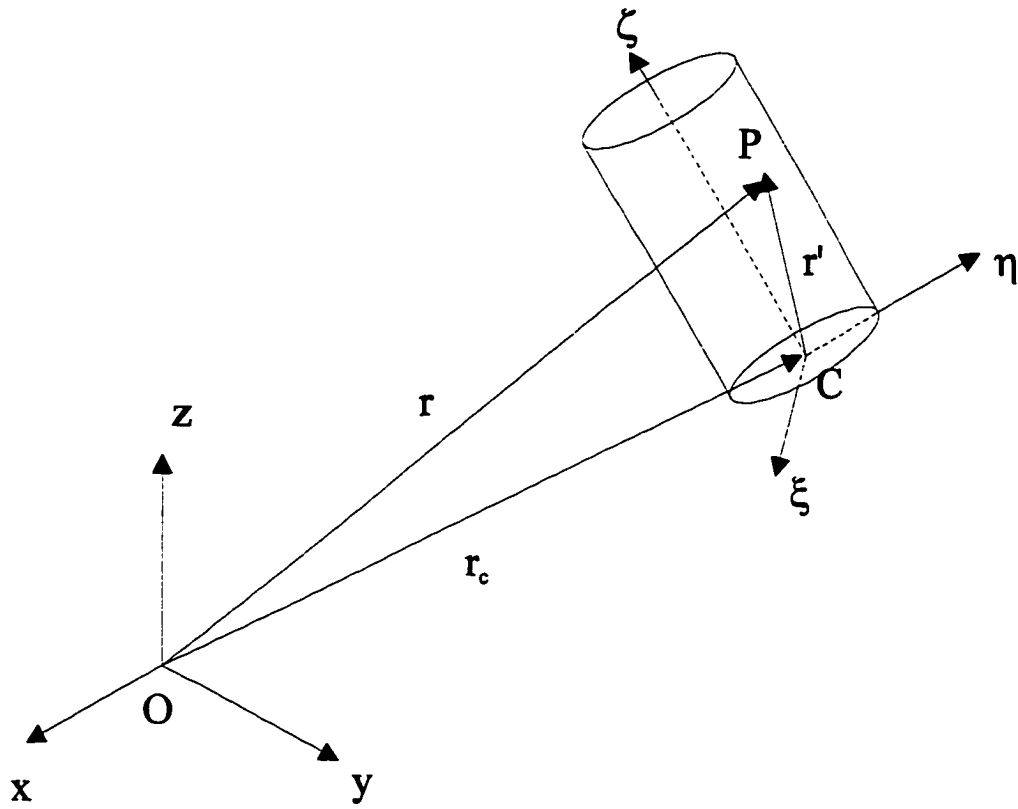
In this chapter, the SDE approach is only implemented for flat or ball-end tools and Type A machines. The same approach may be applied to general 7-parameter APT tools (Appendix B) and more general five-axis machines by performing general kinematics analysis procedures described in Appendix A.



**Figure 3.1** A five-axis NC milling machine and reference coordinate frame O-xyz

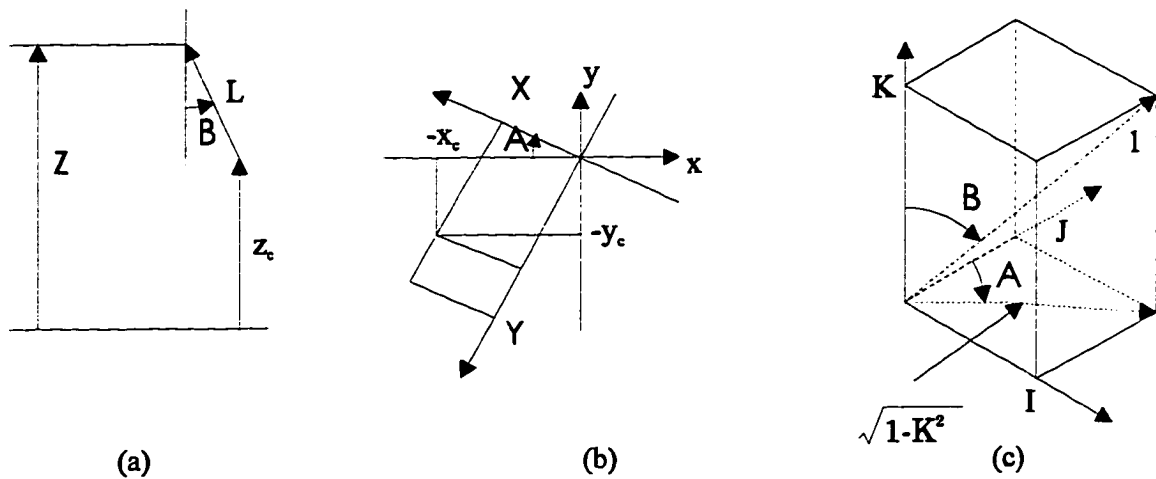


**Figure 3.2** Tool geometry and moving coordinate frame  $C$ - $\xi\eta\zeta$



**Figure 3.3** Relationship of the coordinate frames





**Figure 3.4** Geometric relationship between MCD data and CL data

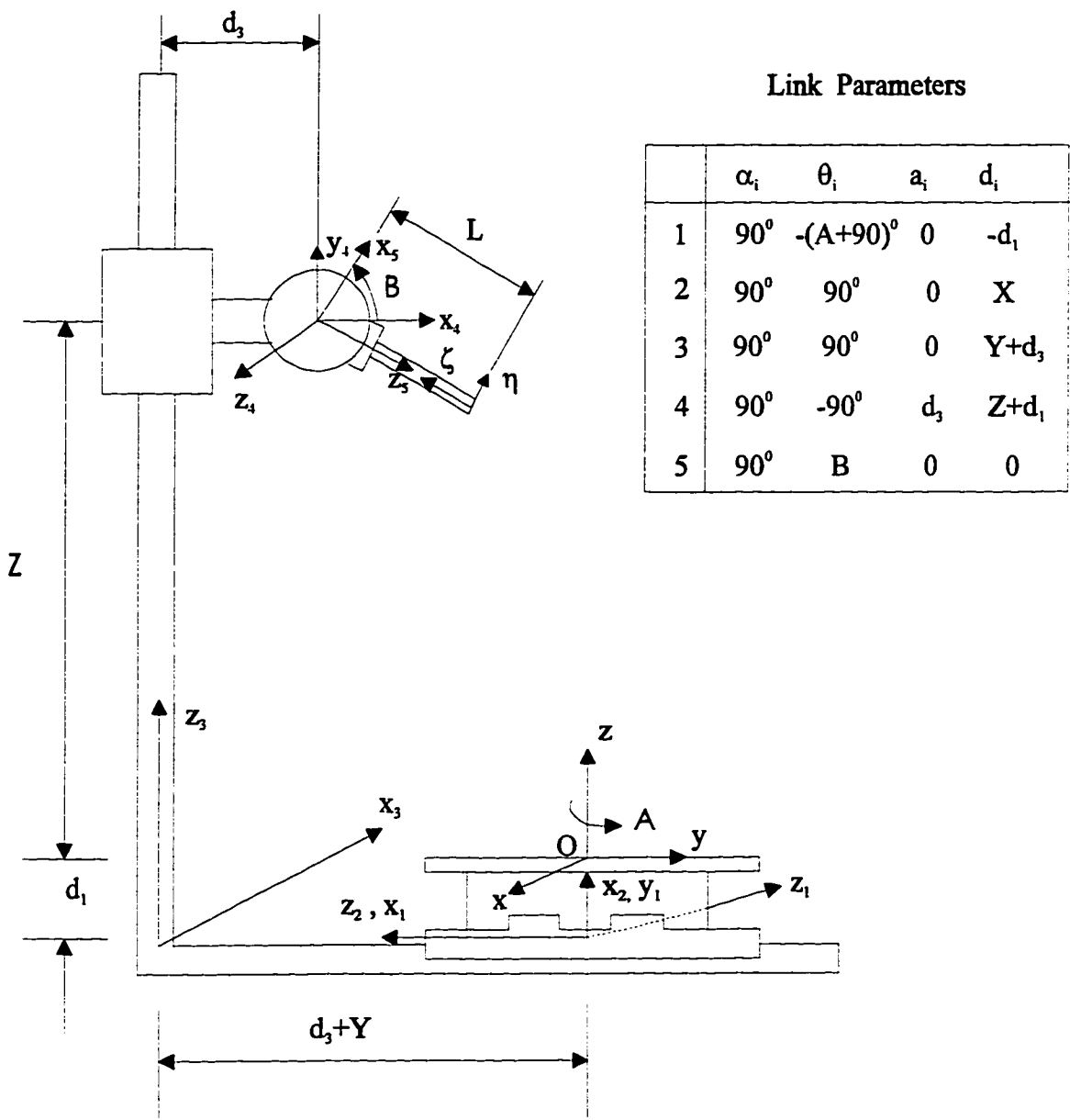
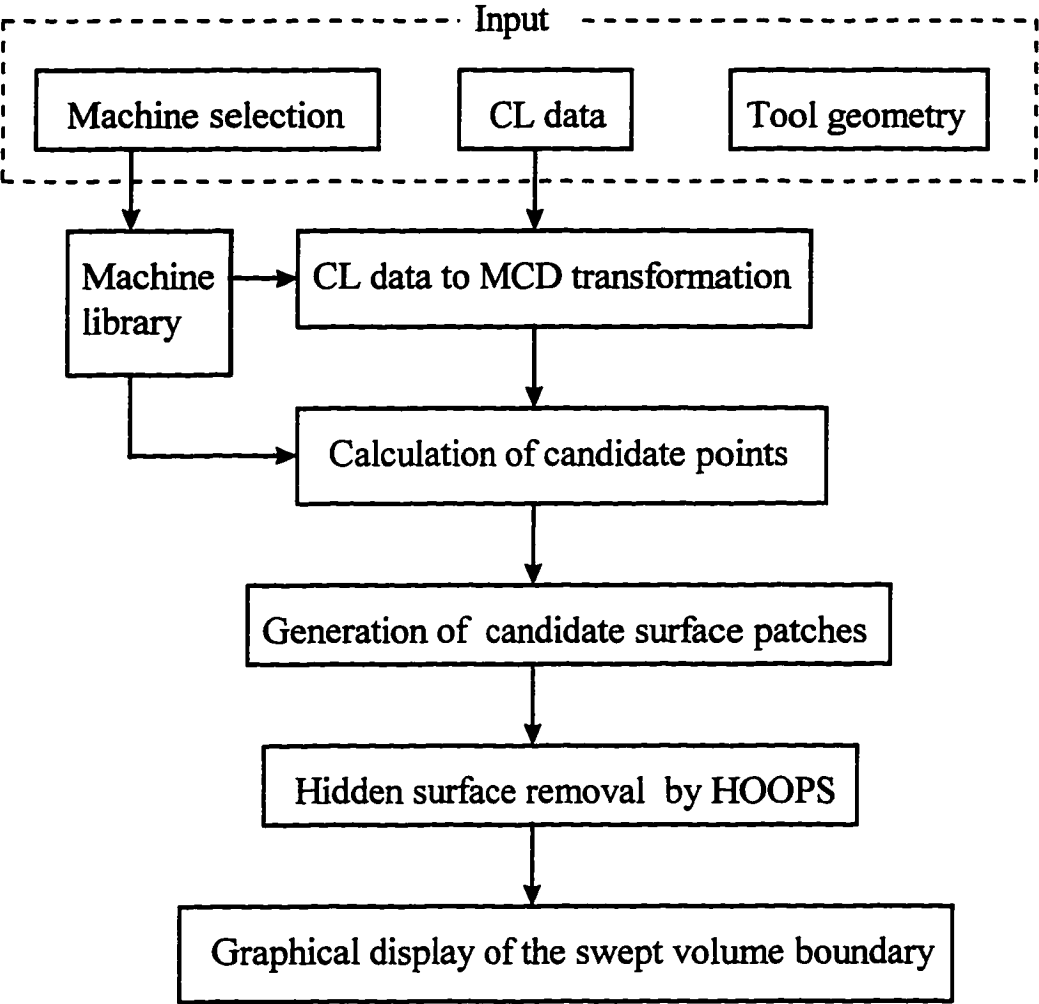


Figure 3.5 Denavit-Hartenberg representation for the machine



**Figure 3.6** Flow chart of the SDE-3D computer program



**Figure 3.7** Swept volume and some of the surface patches

## CHAPTER 4

### GENERATING SWEEP SOLIDS FOR NC VERIFICATION USING THE SEDE METHOD

In this chapter, a SEDE-based algorithm for the numerical computation of swept volume is described. Then it is combined with some novel smooth approximation formulas in order to calculate the swept volumes generated by a general 7-parameter APT tool for a large class of sweeps that include the motions encountered in five-axis NC milling processes. Using the SEDE algorithm a C program called SWEEP-SEDE is written to represent cutter swept volumes. The program was first implemented for a smooth object whose boundary can be represented by a smooth function and then extended to more general piecewise smooth objects like general 7-parameter APT tools. Theoretical aspects associated with this implementation include: object description and approximation parametrically, representing motion equation parametrically, time complexity analysis of the SEDE algorithm, upper bound error estimate.

#### 4.1 SEDE Swept Volume Algorithm

The SDE, BFF and the SEDE can be used as a theoretical framework for the construction of an efficient scheme for the numerical approximation of swept volumes. The algorithm can be described as follows:

##### **Step 1. Represent solid object implicitly:**

In the SDE/SEDE approach, the object surface ( $\partial M$ ) is defined as the set of points satisfying

$$f(x) = 0$$

for a smooth function  $f$ . The normal vector on the object surface has continuous derivative at every point of  $\partial M$ . Usually a smooth function  $f$  characterizing the

object is not a priori given. It is therefore necessary to find a smooth function  $\hat{f}$  such that the locus of the equation

$$\hat{f}(x) = 0$$

approximates the surface of the object at a level consistent with accuracy requirements.

**Step 2. Represent sweep  $\sigma$  parametrically:**

A sweep is represented by

$$\sigma_t(x) = \Gamma(t) + \tau(t)x.$$

**Step 3. Calculate initial grazing points:**

$\partial_0 M(0)$  is the solution set of the equations

$$\hat{F}(x, 0) = 0,$$

$$\langle N, X_\sigma \rangle = 0.$$

This set is approximated by triangulating the surface defined by the first equation above with a triangulation mesh size  $\delta$  ( $=$  maximum diameter of all triangles and  $\delta > 0$ ) and then applying the second equation on each vertex of the triangulation. A discrete approximation  $\{x_{(1)}^0, \dots, x_{(m)}^0\}$  of the curve  $\partial_0 M(0)$  is obtained. The grazing curve  $\partial_0 M(0)$  consists of  $m$  points separated by a distance of not more than  $\delta$ .

**Step 4. Find grazing set:**

Select a time step  $\epsilon > 0$ , where  $\epsilon = 1/n$  for some positive integer  $n$ , and compute discrete points on the approximate trajectories of the SEDE starting at the points  $\{x_{(1)}^0, \dots, x_{(m)}^0\}$  for  $0 \leq t \leq 1$  using a Runge-Kutta scheme with global truncation error  $O(\epsilon^4)$ . This yields approximate trajectories  $\{x_{(k)}^0, \dots, x_{(k)}^n\}$  for  $1 \leq k \leq m$ .

**Step 5. Find ingress and egress sets:**

Compute  $\partial_- M(0)$  which is characterized by  $\hat{F}(x, 0) = 0$  and  $\langle N, X_\sigma \rangle < 0$  and  $\partial_+ M(1)$  by finding the points satisfying  $\hat{F}(x, 1) = 0$  and  $\langle N, X_\sigma \rangle > 0$ .

**Step 6. Perform trimming:**

The trimming procedure may be done by computing  $F(x_{(k)}^j, i\epsilon)$  ( $i < j$ ) along the approximate trajectories obtained by solving SEDE equation. If at any time instance,  $F(x_{(k)}^j, i\epsilon) < 0$  for some  $i < j$  is detected, remove or trim the point  $x_{(k)}^j$  from the trajectory. This condition may have to be tested at additional points when the curvature of the sweep is large.

**Step 7. Triangulate the grazing set:**

Triangulate the grazing set by linear interpolation between successive untrimmed points on all trajectories and linear interpolation between all untrimmed points on adjacent trajectories, that is, linear interpolation between  $x_{(k)}^j$  and  $x_{(k)}^{j+1}$  and linear interpolation between  $x_{(k)}^j$  and  $x_{(l)}^m$  for all  $|l - k| = 1$  and  $|m - j| \leq 1$  as shown in Fig. 2.4.

**Step 8. Output:**

Obtain an approximation of the swept volume and use the numerical data in conjunction with the triangulation to produce a graphical representation of the swept volume.

Several features of the algorithm described in steps 1-8 were specifically designed to facilitate integration of the computer implementation of the SEDE algorithm with the particular commercial CAD software that is used. With minor modification the basic algorithm can be effectively interfaced with any of the common CAD software packages for a variety of practical applications.

The properties of the SEDE are illustrated in Fig. 2.4, including a triangulation of the grazing set associated with the foliation by trajectories.

To summarize: first, the object  $M$  and its sweep  $\sigma$  must be described parametrically. And then the SDE approach is used to compute the initial grazing points ( $\partial_0 M(0)$ ) of the object  $M$ . The grazing sets of subsequent time instances can be obtained by solving the sweep-envelope differential equation (SEDE) using the computed  $\partial_0 M(0)$  as the initial condition. After performing the trimming, an approximation of the swept volume is obtained and the numerical data are used in conjunction with a triangulation to produce a graphical representation of the approximate swept volume. Fig. 4.1 shows a flow chart of the SEDE algorithm and its implementation procedures. It will be shown in the next sections how this approach can be applied to swept volumes of cutting tools.

#### 4.2 Decomposition of Tool into Closed Smooth Objects

The object considered here is a general 7-parameter APT tool model which is described in Chapter 2 (Section 2.5.1) and Appendix B. Fig. 4.2 shows the  $\xi\zeta$  cross section of a typical 7-parameter APT tool, its local coordinate frame  $C - \xi\eta\zeta$  and some critical points. The boundary surface of a general 7-parameter APT tool is only piecewise smooth and can be decomposed into three smooth parts: lower conical part, toroidal part and upper conical part. At present the SEDE method requires the object to be closed and bounded with a boundary surface  $\partial M$  on which the normal vector has continuous derivatives at every point of  $\partial M$ . So smoothing of a general 7-parameter APT tool boundary surface becomes a crucial step in the SEDE algorithm implementation.

Only recently, describing complicated objects by implicit polynomials (Keren et al., 1994; Taubin et al., 1994) is receiving the attention in vision, graphics, robotics and computer-aided geometric design. Super-quadric polynomials have proved to be



very useful in computer vision for representing spheres, cylinders, cones and planes. A super-quadric is defined as

$$C_s = \{(x, y, z) : x^s + y^s + z^s = 1\}.$$

$C_s$  is a nonconvex star-shaped set when  $s$  is small and approaches the unit square as  $s$  grows. But super-quadrics are limited to symmetric objects and are too restricted, so usually a complicated object has to be decomposed into many disjoint parts, each of them described by a super-quadric.

Considering the characteristics of a super-quadric, the geometric feature and the axisymmetric property of a general 7-parameter APT tool, modified super-quadric polynomials are derived in the following subsections to smooth a general tool part by part. The basic idea is to round off the sharp corners in the cross section of a 7-parameter APT tool and derive implicit polynomial functions so that this approximated functions have continuous derivatives of all orders at any point  $x$ . After the smoothing procedure, a general 7-parameter APT tool is decomposed into three smooth parts: lower conical part, toroidal part and upper conical part, each of them described by a modified super-quadric.

#### 4.2.1 Lower Conical Part

Let  $r_D$  and  $\zeta_D$  represent the maximum radius and the height of the lower conical part respectively and  $\xi, \eta, \zeta$  the coordinates of any point on the tool surface in the tool coordinate frame. By taking some practice, an approximate functional representation is derived and of the form:

$$\hat{f}_1(\xi, \eta, \zeta) = \left(\frac{\sqrt{\xi^2 + \eta^2}}{r_D}\right)^{2p} + \left(\frac{\zeta}{\zeta_D}\right)^{2p} \left(\frac{\zeta - \zeta_D + 1}{\frac{\zeta_D}{2}}\right)^{2p} - \left(\frac{\zeta}{\zeta_D}\right)^{2p} = 0 \quad (4.1)$$

where  $0 < \zeta \leq \zeta_D$  and

$$\zeta_D = \sin \alpha (a \cos \alpha + b \sin \alpha) \quad (4.2)$$

The exponent  $p$  controls the accuracy. It is found that the accuracy of the approximation above can be improved by increasing the parameter  $p$  ( $p = 1, 2, 3, 4, \dots$ ). If  $p$  is too small, a crude approximation to the actual surface is obtained. If  $p$  is too large, the degree of the implicit polynomial becomes very high and the computation cost will increase. From the practice,  $p = 10$  is a critical value for achieving good approximation and low degree of polynomials. The speed of an accuracy improvement is almost a constant for  $p > 10$ . Illustration of this process is shown in Fig. 4.3, assuming that  $d = 2$ ,  $r = 0$ ,  $a = 1$ ,  $b = 2$ ,  $\alpha = \arctan 2$ ,  $\beta = 0$ ,  $h = 2$ ,  $p = 1, 2, 3, 6, 7, 8, 10, 50, 100, 200$  respectively. It can be calculated that  $r_D = 1$ ,  $\zeta_D = 5$ .

#### 4.2.2 Toroidal Part

The toroidal part of the tool can be represented exactly in the form

$$f_2(\xi, \eta, \zeta) = (\sqrt{\xi^2 + \eta^2} - a)^2 + (b - \zeta)^2 - r^2 = 0 \quad (4.3)$$

where  $\zeta_D \leq \zeta \leq \zeta_F$  and

$$\zeta_F = \cos^2 \beta [2b - \tan \beta (\frac{d}{2} - \frac{d}{2} \tan \alpha \tan \beta - a)] \quad (4.4)$$

#### 4.2.3 Upper Conical part

By taking some practice, an approximate functional representation is derived and provided by:

$$\hat{f}_3(\xi, \eta, \zeta) = [\frac{\sqrt{\xi^2 + \eta^2}}{r_F + (\zeta - \zeta_F) \tan \beta}]^{2q} + (\frac{\zeta - \zeta_F - \frac{h - \zeta_F}{2}}{\frac{h - \zeta_F}{2}})^{2q} - 1 = 0 \quad (4.5)$$

where  $\zeta_F \leq \zeta \leq h$ .  $\zeta_F$  is determined by Eqn.(4.4).

The exponent  $q$  controls the accuracy. It is found that the accuracy of the approximation above can be improved by increasing the parameter  $q$  ( $q = 1, 2, 3, 4, \dots$ ). If  $q$  is too small, it will lead to a crude approximation to the actual surface. If  $q$  is too large, the degree of the implicit polynomial becomes very high

and the computation cost will increase. From the practice,  $q = 100$  is a critical value for achieving good approximation and low degree of polynomials. The speed of an accuracy improvement is almost a constant for  $q > 100$ . Illustration of this process is shown in Fig. 4.4, assuming that  $d = 0.5, r = 0, a = 0, b = 0, \alpha = 0, \beta = 0, h = 5, q = 1, 2, 5, 10, 20, 30, 50, 100$  respectively.

Using these functional representations of the tool surfaces, the normals on the tool surfaces, derivatives of normals and the Hessian matrix can be calculated easily.

### 4.3 General Motion Equation

Generally, the motion equation may be described as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \Gamma_x(t) \\ \Gamma_y(t) \\ \Gamma_z(t) \end{pmatrix} + \begin{pmatrix} \tau_{11}(t) & \tau_{12}(t) & \tau_{13}(t) \\ \tau_{21}(t) & \tau_{22}(t) & \tau_{23}(t) \\ \tau_{31}(t) & \tau_{32}(t) & \tau_{33}(t) \end{pmatrix} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix} \quad (4.6)$$

where the vector  $(x, y, z)^T$  represents any point in the reference coordinate frame  $xyz$  and  $(\xi, \eta, \zeta)^T$  represents any point in the tool coordinate frame  $\xi\eta\zeta$ , and

$$\begin{aligned} \tau_{11}(t) &= \cos r_x(t) \cos r_y(t) \\ \tau_{21}(t) &= \sin r_x(t) \cos r_y(t) \\ \tau_{31}(t) &= -\sin r_y(t) \\ \tau_{12}(t) &= \cos r_x(t) \sin r_y(t) \sin r_z(t) - \sin r_x(t) \cos r_z(t) \\ \tau_{22}(t) &= \sin r_x(t) \sin r_y(t) \sin r_z(t) + \cos r_x(t) \cos r_z(t) \\ \tau_{32}(t) &= \cos r_y(t) \sin r_z(t) \\ \tau_{13}(t) &= \cos r_x(t) \sin r_y(t) \cos r_z(t) + \sin r_x(t) \sin r_z(t) \\ \tau_{23}(t) &= \sin r_x(t) \sin r_y(t) \cos r_z(t) - \cos r_x(t) \sin r_z(t) \\ \tau_{33}(t) &= \cos r_y(t) \cos r_z(t) \end{aligned}$$

#### 4.4 Five-Axis Motion Equation

In Chapter 3, the tool motion equation is derived in accordance with actual five-axis machine motion but may be inconvenient due to its machine dependent characteristics. In this section, a CL interpolation based five-axis motion planar is adopted.

For the motion of a multi-axis machine, it can be generally assumed that each axis has a constant speed and all the axes start and end at the same time, i.e. the normal form of operation is joint-interpolated motion. For a five-axis machine, both the position of the control point and the orientation of the tool-axis have to be interpolated.

The motion from one configuration to the next configuration by such an interpolation is referred to as *five-axis motion*. In NC simulation, the linear interpolation between two tool configurations may be approximated based on CL (Cutter Location) interpolation (Sambandan, 1988).

CL Data is defined as

$$\mathbf{CL} = (x_c, y_c, z_c, i, j, k),$$

where  $(x_c, y_c, z_c)$  (called the Tool Control Point or TCP) are the coordinates of tool end  $C$  in the reference frame and  $i, j, k$  are the normalized directional cosines of the moving tool axis ( $\zeta$  - axis) in the reference frame. The CL Data defines the tool location and orientation in the workpiece coordinate frame.

Let  $\theta = \arctan(\frac{j}{i})$  and  $\varphi = \frac{\pi}{2} - \arctan(\frac{k}{\sqrt{i^2+j^2}})$ . The transformation of coordinate systems from tool frame to workpiece frame (or motion equation ) is of the form

$$\begin{aligned} \mathbf{r} &= \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} \\ &+ \begin{pmatrix} \cos \theta \cos \varphi & -\sin \theta & \sin \varphi \cos \theta \\ \cos \varphi \sin \theta & \cos \theta & \sin \theta \sin \varphi \\ -\sin \varphi & 0 & \cos \varphi \end{pmatrix} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix} \end{aligned} \quad (4.7)$$

where

$$\begin{pmatrix} x_c(t) \\ y_c(t) \\ z_c(t) \\ i(t) \\ j(t) \\ k(t) \end{pmatrix} = \begin{pmatrix} x_c(0) \\ y_c(0) \\ z_c(0) \\ i(0) \\ j(0) \\ k(0) \end{pmatrix} + t \begin{pmatrix} x_c(1) - x_c(0) \\ y_c(1) - y_c(0) \\ z_c(1) - z_c(0) \\ i(1) - i(0) \\ j(1) - j(0) \\ k(1) - k(0) \end{pmatrix}$$

$x_c(0), y_c(0), z_c(0), i(0), j(0), k(0)$  and  $x_c(1), y_c(1), z_c(1), i(1), j(1), k(1)$  represent the initial and final configurations of the tool respectively, and  $t \in [0, 1]$  is the normalized time variable.

#### 4.5 Generating Initial Grazing Points

The SDE approach is based upon to compute  $\partial_0 M(0)$  which is characterized by  $F(x, 0) = 0$  (initial tool configuration) and  $\langle N, X_\sigma \rangle = 0$ , where  $N$  denotes the outer unit normal vector on  $\partial M$  and  $X_\sigma$  is given by Eqn.(2.3). The computed initial grazing points are reorganized in clockwise or counter-clockwise order and represented by  $\{x_{(1)}^0, \dots, x_{(m)}^0\}$  (i.e.,  $m$  initial grazing points altogether).

In the SEDE method, the calculation of initial grazing points is critical, and failure to satisfy the tolerance requirement will lead to crude approximation of the swept volume. So an algorithm is developed here to find initial grazing points recursively. This algorithm may also be used in the calculation of grazing points by the SDE method.

##### Algorithm for finding initial grazing points recursively:

- {Input:** (1) tool geometry  $d, r, a, b, \alpha, \beta, h$ ;  
 (2) tool discretization size in  $\zeta$  axis direction  $N_z$ , mesh size for a circle  $N_{cir}$ , time discretization step  $N_t$ , number of steps used to test possible grazing points between any two adjacent points  $N_{inter}$ ;  
 (3) tolerance TOL;

(4) motion parameter.

**Output:** ordered initial grazing points. }

**type**

pointtype = record

x: double; { x coordinate }

y: double; { y coordinate }

z: double { z coordinate }

**end;**

**type**

curvetype = record

P: pointtype; {  $x, y, z$  coordinates of a point }

$n_1$ : integer; {  $N_{inter}$  }

number: integer; { total number of points in this curve }

next : ↑ curvetype;

prev : ↑ curvetype

**end;**

**procedure** FindInitialGrazing(var Curve: curvetype); {main program }

**var**

$i, j$  : integer;

$z_i, r_i$ : double;

$\theta_j, \theta_{j+1}$ : double;

$N_g$ : constant; {number of initial grazing points}

**begin**

{ tool surface is sliced into  $N_z$  sections with index  $i$ ; }

**for**  $i := 1$  to  $N_z$  **do** { on each slice of tool surface }

```

begin
   $z_i = \frac{ih}{N_z};$ 
  find  $r_i$  using the tool surface equation;
  { on each slice of the tool, discretize the circle into
     $N_{cir}$  points with index  $j$ ; }
  for  $j:=0$  to  $N_{cir} - 1$  do
    begin
       $\theta_j := \frac{2j\pi}{N_{cir}}; \{ \text{at point } P_j \}$ 
       $\theta_{j+1} := \theta_j + \frac{2\pi}{N_{cir}}; \{ \text{at point } P_{j+1} \}$ 
      Recursive(Curve,  $\theta_j$ ,  $\theta_{j+1}$ ,  $z_i$ ,  $r_i$ );
    end;
  end;

```

{ The next while loop is optional, and is used to interpolate virtual grazing points on slices where no grazing points are found by the regular routine. }

```

while not the last point in Curve do
  begin
    for any two adjacent grazing points  $G0(z_0, \theta_0)$  and  $G1(z_1, \theta_1)$  do
      begin
        for each slice between  $G0$  and  $G1$  do
          begin
            add two virtual grazing points using the
            interpolation of  $G0$  and  $G1$ ;
            insert them into data structure Curve;
          end;
        end;
      end;
     $G0=G1; G1=next;$ 
  end;

```

```

    end;    { end of while loop }

    adjust points in Curve into ordered points;

end;    { END FindInitialGrazing }

procedure Recursive(var tlp: curvetype;  $\theta_j, \theta_{j+1}, z_i, r_i$ : double);
    { find grazing points recursively. }

var
    dot0, dot1: double;
     $\theta$ : double;

begin
    if point  $P_{j+1}(z_i, \theta_{j+1})$  the last point in this tlp then stop;
    find the outer normals at points  $P_j$  and  $P_{j+1}$ ;
    find the SVFs for points  $P_j$  and  $P_{j+1}$ ;
    find dot product dot0 at  $P_j$  and dot1 at  $P_{j+1}$ ;
    if ( dot0  $\leq$  TOL ) then
        insert point  $P_j$  into data structure tlp
    else if dot0 and dot1 have different sign then
        begin
            { there exists a grazing point in segment  $P_j P_{j+1}$  }
             $\theta := \theta_j + (\theta_{j+1} - \theta_j)/2$ ;
            { find grazing point recursively; }
            Recursive(tlp,  $\theta_j, \theta, z_i, r_i$ );
            Recursive(tlp,  $\theta, \theta_{j+1}, z_i, r_i$ );
        end
    else
        return;
    end;    { END Recursive }

```



#### 4.6 Determining Grazing Set

Select a time step  $\epsilon > 0$  and compute discrete points approximately on the trajectories of the SEDE starting at the points  $\{x_{(1)}^0, \dots, x_{(m)}^0\}$  for  $0 \leq t \leq 1$  by using the SEDE equation and a fourth-order Runge-Kutta scheme with global truncation error  $O(\epsilon^4)$ . This yields approximate trajectories  $\{x_{(k)}^0, \dots, x_{(k)}^n\}$  for  $1 \leq k \leq m$ , where  $\epsilon = 1/n$  (i.e.,  $n+1$  time instances altogether).

**Algorithm for finding grazing points by the SEDE and RK methods:**

**{Input:** (1) tool geometry  $d, r, a, b, \alpha, \beta, h$ ;  
 (2) tool mesh size in  $\zeta$  axis direction  $N_z$ , mesh size for a circle  $N_{cir}$ ,  
 time discretization steps  $N_t$ , number of steps used to test  
 intermediate grazing points between any two adjacent points  $N_{inter}$ ;  
 (3) tolerance TOL;  
 (4) motion parameters.

**Output:** Oriented polyhedral approximation of the swept solid }

**type**

hulltype = record

number: integer; {total number of points in each grazing curve}

front : ↑ curvetype;

rear : ↑ curvetype

**end;**

**procedure** SEDE-RK() ;

{ finding grazing points by SEDE and RK }

**var**

$k, q$ : integer;

```

     $t_k$ : double;
    hull: ↑hulltype;
    curve: ↑curvetype;
     $N_g$ : integer;    { number of initial grazing points }

begin
    input CL file and postprocessing;
    FindInitialGrazing(curve);
    hull↑.rear := curve ;
    hull↑.number := curve↑.number ;
     $N_g$ := curve↑.number;
    { the normalized time is discretized into  $N_t$  intervals with index  $k$ ; }
    allocate memory;
    calculate initial SEDE orbit  $\dot{x}^0$ ;
    while  $k \langle \rangle N_t$  do
    begin
         $t_k := \frac{k}{N_t}$ ;
        for  $q := 0$  to  $N_g - 1$  do { at each point in curve }
            perform RK formulas to get hull↑.front at time  $t_k$ ;
            hull↑.rear := hull↑.front;
        end { end while }
        free memory;
    end.    { end SEDE-RK }

```

#### 4.7 Determining Ingress and Egress Sets

Compute  $\partial_- M(0)$  which is characterized by  $F(x, 0) = 0$  and  $\langle N, X_\sigma \rangle < 0$  and  $\partial_+ M(1)$  by finding the points satisfying  $F(x, 1) = 0$  and  $\langle N, X_\sigma \rangle > 0$ .

For integration with CAD/CAM software in which the tool configurations are modeled and displayed for animation of tool movements, the calculation of ingress and egress points is replaced by the tool configurations at the appropriate time instances at the preprocessing stage. The tool swept volume for each NC block is the union of the tool initial position, final position and the grazing set.

#### 4.8 Boundary Trimming

As described in the SDE and SEDE theories, some of the grazing points calculated are not actual boundary points of the swept volume and should be trimmed. H. Jiang (Jiang, 1993) described local and global trimming procedures which are efficient for the sweeping of polygonal objects in a 2-D space (i.e. a plane). The local trimming procedure is applied during the process of generating grazing points. The global trimming procedure is applied after all the grazing points have been generated.

The trimming process in the 3D case is the same in principle; however, it is computationally much more demanding. The trimming procedure is performed by computing  $F(x_{(k)}^j, i\epsilon)$  ( $i < j$ ) along the approximate trajectories obtained by solving SEDE equation. If at any time instance,  $F(x_{(k)}^j, i\epsilon) < 0$  is detected, remove or trim the point  $x_{(k)}^j$  from the trajectory. This condition may have to be tested at additional points when the curvature of the sweep is large.

A five-axis motion is composed of simple, linearly interpolated pieces or blocks. By considering only the relative positions of any two tool configurations, five-axis motions can be classified into seven types (illustrated in Fig. 4.5). Of the seven types of motions, Type 1 is a singular case in which all the points on the tool surface are grazing points. It is not necessary to perform the trimming procedures for Type 2, 3 and 7 because no self-intersections occur. Local trimming procedure (Section 2.4) is needed for Type 4, 5 or 6 motions, i.g., planar rotational motions,

and the details have been discussed in Jiang's (Jiang, 1993) and Deng's work (Deng et al., 1994a).

#### 4.9 Swept Volume Representation and Graphic Display

The boundary surface of a swept volume may be represented in three ways: (i)function; (ii)B-spline; (iii)polygons. Since the SEDE sweep generator is targeted to be integrated with Deneb's Virtual NC in which polygons are used to represent solid primitives, the explicit polygonal set method is utilized to represent the boundary of the swept volumes, which is easier for interface with Virtual NC.

Using the SEDE approach described above, the grazing set of the swept volume is obtained. An important step is using the calculated grazing points in conjunction with the triangulation to produce a representation of the approximate swept volume for either graphical display or standard Boolean subtraction.

Let  $j$  represent any time instance ( $j=0,1,2,...,n$ ) and  $k$  be an index of any grazing point ( $k=1,2,..., m$ ). Four adjacent grazing points  $x_{(k)}^j, x_{(k+1)}^j, x_{(k)}^{j+1}$  and  $x_{(k+1)}^{j+1}$  are used as a vertex set.

$$V_{kj} = (x_{(k)}^j, x_{(k+1)}^j, x_{(k)}^{j+1}, x_{(k+1)}^{j+1})$$

Then the vertex set is organized into  $2m \times n$  triangles.

$$\begin{aligned} P_{k,j}^1 &= (x_{(k)}^j, x_{(k+1)}^j, x_{(k)}^{j+1}) \\ \text{and } P_{k,j}^2 &= (x_{(k+1)}^j, x_{(k+1)}^{j+1}, x_{(k)}^{j+1}). \end{aligned} \tag{4.8}$$

The initial and final grazing curves (the front and rear holes) have to be covered to form a closed, oriented grazing solid.

By union of initial and final tool configurations (also represented using triangulation) and the grazing set, the approximation of the swept volume is obtained. The result may be displayed using a GSL (Graphical Simulation Language) (Deneb,

1995) macro on the platform of Virtual  $NC^{TM}$  (VNC). It reads the vertex information of all polygons and displays the solid in Virtual NC's 3D workcell. The swept volume may be saved in standard CAD data transformation format such as IGES, STL, Pro/E part file or CATIA, using the build-in data exchange function in VNC. The approach can be readily interfaced with CAD/CAM systems; this will be demonstrated in the sequel using VNC to simulate the material removal process in five-axis NC milling in the following chapters.

For more accurate surface representation, a parametric method, e.g., B-spline, may be adopted.

#### 4.10 Complexity Analysis of the SEDE Algorithm

To measure the efficiency of an algorithm, a subjective method, called the worst-case time complexity analysis, may be adopted (Aho & Ullman, 1987). The running time of an algorithm is measured by a growth rate function  $f(n)$  (here  $n$  means the input size).  $f(n)$ , also called the time complexity of an algorithm, is the sum of the principal operations for solving a problem such as additions and comparisons. One of the important tasks of an algorithm designer is to develop an algorithm with as lower growth rate (time complexity) as possible. From the algorithms described in Section 4.6, 4.7, and 4.10, the time complexity can be calculated by following the time complexity analysis procedures described by Aho & Ullman (Aho & Ullman, 1987).

Let  $T$  denote the global time complexity for the SEDE algorithm,  $c_1$  to  $c_5$  be constants and  $n_1$  the discretization steps chosen to test grazing points between any two points corresponding to  $(\theta_j, z_j)$  and  $(\theta_{j+1}, z_{j+1})$  respectively.

The time complexity for subroutine FindInitialGrazing (Section 4.6) is:

$$T_1 = \sum_{i=1}^{N_z} [c_1 + \sum_{j=0}^{N_{cir}-1} (c_2 + T(n_1))] \quad (4.9)$$

where  $T(n_1)$  is the time complexity for the recursive subroutine Recursive (Section 4.6) and

$$\begin{cases} T(n_1) &= 2T(\frac{n_1}{2}) \\ T(1) &= c_3 \end{cases} \quad (4.10)$$

Solving the above recursive equation,

$$T(n_1) = c_3 n_1 \quad (4.11)$$

Substituting (4.11) into (4.9), the time complexity for finding the initial grazing points is obtained

$$\begin{aligned} T_1 &= \sum_{i=1}^{N_z} [c_1 + \sum_{j=0}^{N_{cir}-1} (c_2 + c_3 n_1)] \\ &= \sum_{i=1}^{N_z} [c_1 + (c_2 + c_3 n_1) N_{cir}] \\ &= [c_1 + (c_2 + c_3 n_1) N_{cir}] N_z \\ &= O(N_z N_{cir} n_1) \end{aligned} \quad (4.12)$$

The time complexity for SEDE-RK in Section 4.7 is:

$$\begin{aligned} T_2 &= c_4 + \sum_{k=1}^{N_t} (c_5 + \sum_{q=0}^{N_g-1} c_6) \\ &= c_4 + \sum_{k=1}^{N_t} (c_5 + c_6 N_g) \\ &= c_4 + (c_5 + c_6 N_g) N_t \\ &= O(N_t N_g) \end{aligned} \quad (4.13)$$

It can be seen that the time complexity for graphic display in Section 4.10 is also :

$$T_3 = O(N_t N_g) \quad (4.14)$$

If  $N = \max(N_z, N_{cir}, n_1, N_t, N_g)$ , then the global time complexity for the SEDE algorithm is:

$$T = T_1 + T_2 + T_3 = O(N_z N_{cir} n_1) + O(N_t N_g) + O(N_t N_g)$$

$$= O(N^3) \quad (4.15)$$

This result implies that the worst-case running time of this SEDE based algorithm grows at the rate of  $N^3$ . It has not been shown that if  $O(N^3)$  is the lower bound of all SEDE based algorithms. By modifying some of the steps in the SEDE algorithms, the time complexity for an SEDE-based algorithm may be reduced to  $N^2$ .

#### 4.11 SWEEP-SEDE Program: Implementation

The SEDE swept volume theory and algorithm described above are used to develop a computer program, called SWEEP-SEDE, to compute and plot pictorially the swept volumes generated by a moving tool under either general or five-axis motion. The general procedures for the SEDE approach are illustrated in Fig. 4.1. The software is implemented using C language and Virtual NC's graphics display function under the IRIX operating system on an INDIGO 2 Silicon Graphics Workstation. This sweep generator serves as a sweep function for computer aided design and can also be combined with other CAD/CAM technology for applications such as computer-based NC verification or robot motion planning.

**Input of SWEEP-SEDE:** Tool geometry in seven parameters; tool discretization step for lower conical ( $Dis_{low}$ ), toroidal ( $Dis_{tor}$ ) and upper conical ( $Dis_{upp}$ ) parts; motion equation and parameters; time discretization step ( $Dis_{time}$ ) and tolerance requirement ( $TOL$ ) for numerical calculation on computer.

**Output of SWEEP-SEDE:** Oriented and triangulated polyhedral approximation of the cutter swept volume.

### 4.12 Illustrative Examples

In this section, some examples using the SWEEP-SEDE program are given, which illustrate the steps in the SEDE algorithm. In all the examples, the tool center path and tool configurations at certain time steps are displayed first, followed by the wireframe picture of the moving grazing curves at all time instances and finally the swept volume after triangulation and shading procedures.

#### Example 4.1

This example shows the calculation and display of the grazing set of a flat-end cutter swept volume using the SWEEP-SEDE program. The tool undergoes a one-block five-axis motion. The origin of the tool coordinate frame is selected at the tip of the tool and  $\xi, \eta, \zeta$  are the coordinates of any point on the tool surface in the tool coordinate frame. The surface of the tool is represented by the smooth function

$$(\xi^2 + \eta^2)^{\frac{q}{2}} + \left(\frac{2\zeta - h/2}{h/2}\right)^{2q} = r. \quad (4.16)$$

Here  $h$  is the tool length,  $r$  the tool radius and the exponent  $q$  controls the accuracy with which the corners of the flat-end tool are rounded off. When the parameter  $q$  is increased to about 100, a good approximation of the surface of a flat-end tool with radius  $r$  and length  $h$  is obtained.

The input specifications for this example are as follows:  $M$  is a flat-end tool represented by Eqn.(4.16) with  $q = 100, r = 0.25$  inches and  $h = 3.0$  inches. The sweep  $\sigma$  is a typical five-axis motion and is approximated by CL interpolation. The tool initial and final positions are

$$(0.78342, -0.25614, 0.57666, 0.3535, -0.61226, 0.70700),$$

$$(-0.43208, 1.32494, 0.67159, 0.00000, -0.11471, 0.99334).$$

The sweep interval is  $[0,1]$  with 20 time steps and 10 tool discretization steps.



Fig. 4.6(a) shows the tool initial and final positions. Fig. 4.6(b) shows the grazing points of the tool swept volume. Fig. 4.6(c) shows the grazing curves at certain time instances. Fig. 4.6(d) shows the grazing solid after triangulation and shading procedures. The cutter swept volume is the union of the tool configurations (Fig. 4.6(a)) and grazing solid (Fig. 4.6(d)). The display is produced by utilizing a commercial software package.

### Example 4.2

This example shows a *one-block five-axis motion*, a sweep in which the tool center point moves in a straight line and the tool orientation performs a linear interpolation. The input specifications for this example are given as follows:

**Object:**  $M$  is a flat-end tool (its boundary is approximated by a smooth function as shown in Example 4.1) with  $r = 2.5$  inches and  $h = 6$  inches.

**Motion equation:** CL interpolation. The tool initial and final positions are:

(0.00000 0.00000 1.00000 -0.82882 3.45578 -0.71972),

(0.00000 0.08716 0.75556 -4.06208 1.32494 -0.67160).

**Sweep interval:**  $[0,1]$

**Other input:**  $N_t = 20$ ,  $N_z = 20$ ,  $TOL = 0.0001$ .

Fig. 4.7 shows the grazing points. Fig. 4.8 shows the wireframe picture of the moving grazing curves at all time instances. Fig. 4.9 shows the swept volume after triangulation and shading procedures.

### Example 4.3

This example shows a *translational sweep*, a sweep in which the tool center point moves in a circle and the tool keeps a fixed orientation. The input specifications for this example are given as follows:

**Object:**  $M$  is a tool represented by 7-parameter APT tool model:  $d = 4, r = 1, a = 1, b = 2.414, \alpha = \pi/4.0, \beta = 0, h = 7$ .

**Motion equation:**  $\Gamma(t)$  and  $\tau(t)$  are specified by the following equations:

$$\Gamma_x = 10\cos(2\pi t);$$

$$\Gamma_y = 10\sin(2\pi t);$$

$$\Gamma_z = -7.07\sin(2\pi t) + 7.07\cos(2\pi t);$$

$$r_x = 0.0;$$

$$r_y = 0.0;$$

$$r_z = 0.0;$$

**Sweep interval:**  $[0,1]$

**Other input:**  $Dis_{time} = 40; Dis_{upp} = 4; Dis_{tor} = 4; Dis_{low} = 4; Dis_{cir} = 16;$   
 $TOL = 0.0001$ .

Fig. 4.10 shows the tool center path and tool configurations at certain time steps. Fig. 4.11 shows the wireframe picture of the moving grazing curves at all time instances, calculated by SWEEP-SEDE. Fig. 4.12 shows the swept volume after triangulation and shading procedures.

#### Example 4.4

This example shows a *translational rotational sweep*, a sweep in which the tool center point moves in a 3D curve and the tool orientation changes with time  $t$ . The input specifications for this example are given as follows:

**Object:**  $M$  is a tool represented by 7-parameter APT tool model:  $d = 4, r = 1, a = 1, b = 2.414, \alpha = \pi/4.0, \beta = 0, h = 7$ .

**Motion equation:**  $\Gamma(t)$  and  $\tau(t)$  are specified by the following equations:

$$\Gamma_x = 10\cos(2\pi t);$$

$$\Gamma_y = 10\sin(2\pi t);$$

$$\Gamma_z = 0.0;$$

$$r_x = 2\pi t;$$

$$r_y = 2\pi t;$$

$$r_z = 0.0.$$

**Sweep interval:**  $[0,1]$

**Other input:**  $Dis_{time} = 80$ ;  $Dis_{upp} = 4$ ;  $Dis_{tor} = 4$ ;  $Dis_{low} = 4$ ;  $Dis_{cir} = 16$ ;

$TOL = 0.0001$ .

Fig. 4.13 shows the tool center path and tool configurations at certain time steps. Fig. 4.14 shows the wireframe picture of the moving grazing curves at all time instances calculated by SWEEP-SEDE. Fig. 4.15 shows the swept volume after triangulation and shading procedures.

### 4.13 Error Analysis and Control

The main sources of modeling errors caused by the SEDE algorithm and their quantified error bounds are analyzed and derived in this section. According to the SEDE theory, algorithm and implementation described above, there are at least six types of possible errors in the SWEEP-SEDE program:

1. The error caused by approximating the tool boundary surface. The tool surface  $f$  is approximated by a smooth function  $\hat{f}$ .
2. The error caused by discretizing the tool object  $M$  into a set of points for the use in calculating initial grazing points. The maximum mesh size is  $\delta_1$ .
3. The error caused by the tolerance in calculating the initial grazing points. i.e., instead of finding  $\hat{F} = 0$  and  $\langle N, X_\sigma \rangle = 0$ , we calculate  $\hat{F} = 0$  and  $\langle N, X_\sigma \rangle < TOL$ . Here  $TOL$  represents a small positive number. In our SWEEP-SEDE program, the default  $TOL$  value is  $10^{-4}$ .

4. The error caused by Runge-Kutta method used in solving SEDE numerically. If the time discretization step is represented in  $\epsilon$ , the global truncation error is  $O(\epsilon^4)$ .
5. The error caused by triangulating the grazing set by linear interpolation between all successive untrimmed points on all approximate trajectories and connecting them into a closed polyhedral swept volume solid.
6. Tool movements are described as line segments (linear interpolation).

Among the six types of errors, Type 3 (TOL) is associated with computer approximation; Type 6 (CL interpolation) error is related to the specific five-axis NC machine model; Type 4 is standard. So attention is focused on Type 1, 2 and 5 errors.

#### 4.13.1 Type 1: Error Caused by Tool Smoothing Approximation

Let  $f_1 = 0$  ( $\hat{f}_1 = 0$ ) and  $f_3 = 0$  ( $\hat{f}_3 = 0$ ) represent the exact (approximated) boundary surface equation for the lower conical part and upper conical part of a general 7-parameter APT tool, respectively. They are derived in appendix B (Section 4.2). Then the absolute error for lower conical section (denoted by  $e_{a1}$ ) and upper conical section (denoted by  $e_{a3}$ ) caused by tool smoothing approximation can be represented as

$$\begin{aligned}
 e_{a1} &= |\hat{f}_1 - f_1| \\
 &= \left| \left( \frac{\sqrt{\xi^2 + \eta^2}}{r_D} \right)^{2p} + \left( \frac{\zeta}{\zeta_D} \right)^{2p} \left( \frac{\zeta - \zeta_D + 1}{\frac{\zeta_D}{2}} \right)^{2p} - \left( \frac{\zeta}{\zeta_D} \right)^{2p} \right. \\
 &\quad \left. - \zeta + \tan \alpha \cdot \sqrt{\xi^2 + \eta^2} \right| \tag{4.17}
 \end{aligned}$$

$$\begin{aligned}
 e_{a3} &= |\hat{f}_3 - f_3| \\
 &= \left| \left[ \frac{\sqrt{\xi^2 + \eta^2}}{r_F + (\zeta - \zeta_F) \tan \beta} \right]^{2q} + \left( \frac{\zeta - \zeta_F - \frac{h - \zeta_F}{2}}{\frac{h - \zeta_F}{2}} \right)^{2q} - 1 \right. \\
 &\quad \left. - \tan \beta \cdot \zeta - \frac{d \cos(\alpha + \beta)}{2 \cos \alpha \sin \beta} + \sqrt{\xi^2 + \eta^2} \right| \tag{4.18}
 \end{aligned}$$

The absolute error caused by tool smoothing approximation, denoted by  $e_a$ , is  $e_a = \max(e_{a1}, e_{a3})$  for the entire tool boundary surface.

#### 4.13.2 Type 2: Error Caused by Tool Discretization

For the purpose of calculating initial grazing points, a tool surface is discretized into a discrete point set. The points between any two adjacent discrete points are approximated by linear approximations. The approximation of a function  $f$  by a straight line gives to a truncation error described by a theorem (Elden & Wittmeyer-Koch, 1990) as follows:

**Theorem** If  $f$  is twice continuously differentiable, and is approximated for  $x_1 \leq x \leq x_2 = x_1 + \Delta x$  by linear interpolation between  $(x_1, f_1)$  and  $(x_2, f_2)$ , the truncation error can be estimated as

$$|R_T| \leq \frac{\Delta x^2}{8} \max |f''(x)|. \quad (4.19)$$

The type 2 error is caused by discretizing the tool object  $M$  into a set of points. It can be easily seen that this type of error is related to the local geometric features of the tool boundary surfaces such as curvature. If  $f_t$  represents the tool boundary surface and  $h_2$  the maximum mesh size in discretizing the tool object, a truncation error caused by discretizing the tool object  $M$  into a set of points (Elden & Wittmeyer-Koch, 1990), denoted by  $e_d$ , is estimated as

$$e_d = \frac{h_2^2}{8} |\max(f_t'')| \leq C_1 \frac{h_2^2}{8} = O(h_2^2) \quad (4.20)$$

#### 4.13.3 Type 5: Error Caused by Polyhedral Approximation

The boundary points calculated by the SEDE algorithm are connected linearly to form the calculated grazing points into a closed solid for either visual display or Boolean operation as described in Section 4.10. The approximation of a function by a straight line gives to a truncation error. The boundary surface (say  $f_s$ ) of the cutter swept volume is represented by a set of triangles to form a closed polyhedron. Linear interpolation between any two vertices of a triangle is adopted and a truncation error (Elden & Wittmeyer-Koch, 1990), denoted by  $e_p$ , is estimated as

$$e_p = \frac{\bar{h}^2}{8} |max(f_s'')| \leq C_2 \frac{\bar{h}^2}{8} = O(\bar{h}^2) \quad (4.21)$$

where  $\bar{h}$  is the maximum length among all the edges of the triangles which form the swept solid and  $C_2$  is a positive constant.

#### 4.13.4 Conclusion

By conducting the above error analysis and referring to the global error analysis procedures given by Jiang (Jiang et al., 1993), it can be seen that if  $K$  represents the maximum mesh size of the tool surface and triangulation of boundary points, then the global error (in combining Type 2 and 5 errors) in the swept volume computation using the SEDE method is  $O(K^2)$ . To increase the accuracy, possible methods may include: use more accurate surface fitting method such as cubic splines (Elden & Wittmeyer-Koch, 1990) or reduce the mesh size of the tool surface.

#### 4.14 Final Remarks

A new algorithm for characterizing swept volume boundaries has been introduced using the trajectories of the SEDE which start at the initial grazing points of the moving object. It provides a useful tool for computing swept volumes for three-dimensional smooth objects undergoing arbitrary smooth motions. Since the grazing

point set needs essentially to be computed only at the initial position of the object – the remaining grazing points are generated by the flow of the sweep-envelope equation – so the computation complexity is drastically reduced. This has been proved by the complexity analysis, actual implementation and examples.

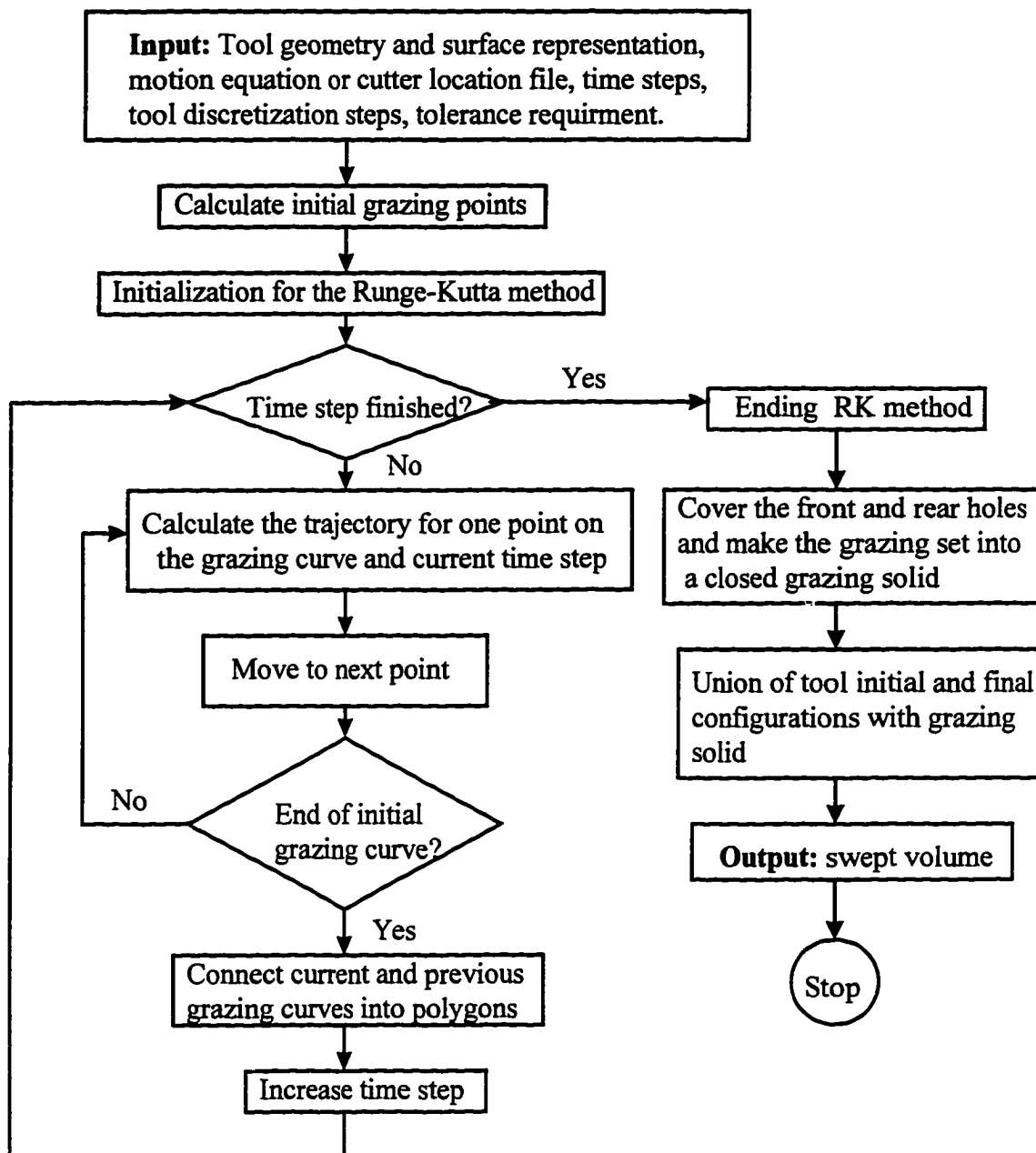
The SEDE algorithm and the SWEEP-SEDE program implemented in this chapter have shown that the orbits of the initial grazing points of the object obtained by solving the SEDE generate the main portion of the swept volume boundary and the SEDE algorithm provides an automatic connectivity for the computed boundary points. This connectivity would facilitate integration with standard algorithm for visualization and Boolean operations in NC verification. In a few singular cases, the regular procedures for finding the grazing points may fail to work. These cases include simple vertical translational motion.

A method for smooth functional approximation of a general 7-parameter APT tool capable of providing any desired accuracy has been developed. This may be useful for other applications where functional modeling of a tool surface is useful. A method for representing five-axis motion parametrically using a CL interpolation is described.

The method for surface fitting described in this paper is developed for our integration with Deneb's VNC which employs polygonal representation for any geometry. The SEDE algorithm can be easily modified to be integrated with other CAD/CAM software.

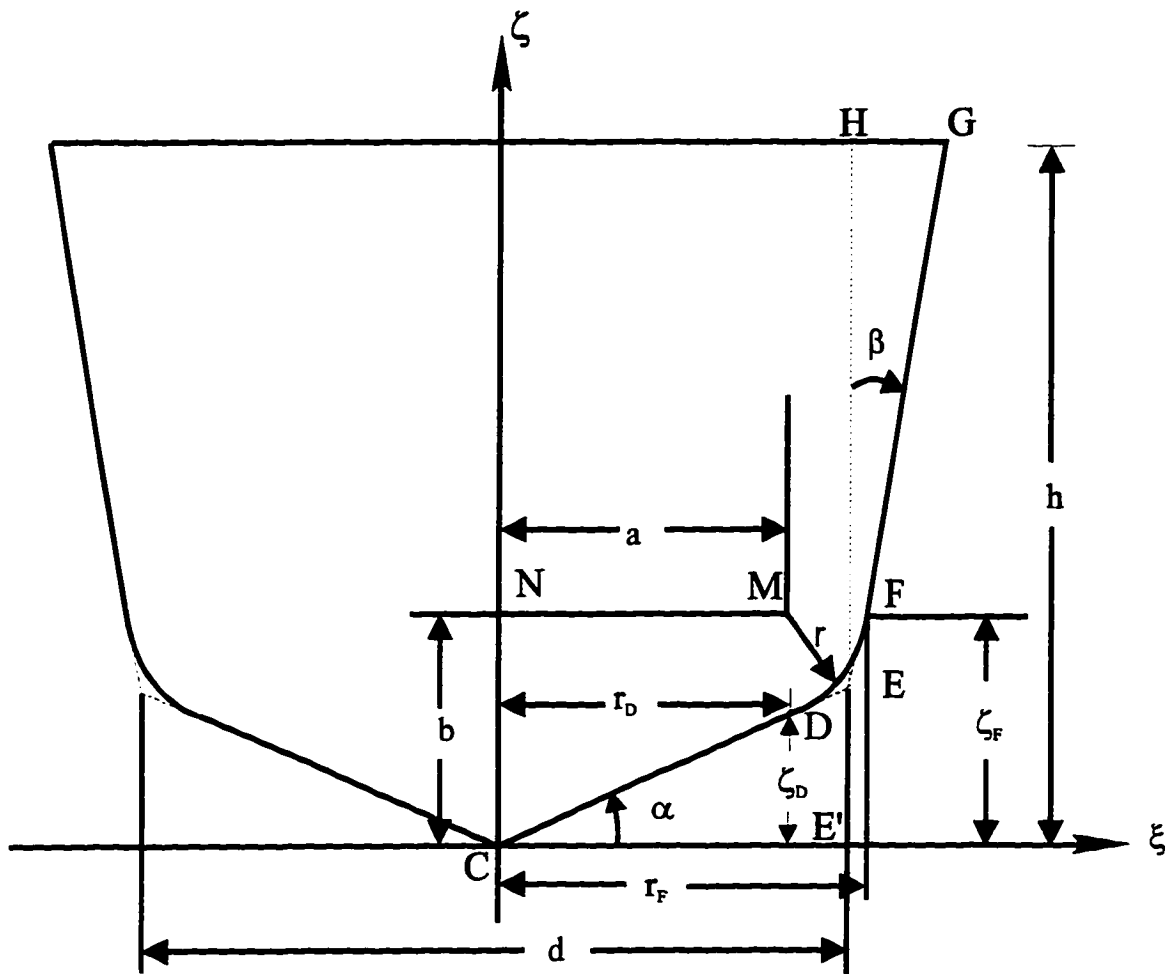
The sources of modeling errors and their upper error bounds have been analyzed and derived. The time complexity for the SEDE algorithm has also been analyzed.

The SEDE feature has been used to construct an efficient algorithm that can be incorporated into a commercial NC machining program and will be presented in the next chapter.



**Figure 4.1** Flow chart of calculation of grazing points by SEDE

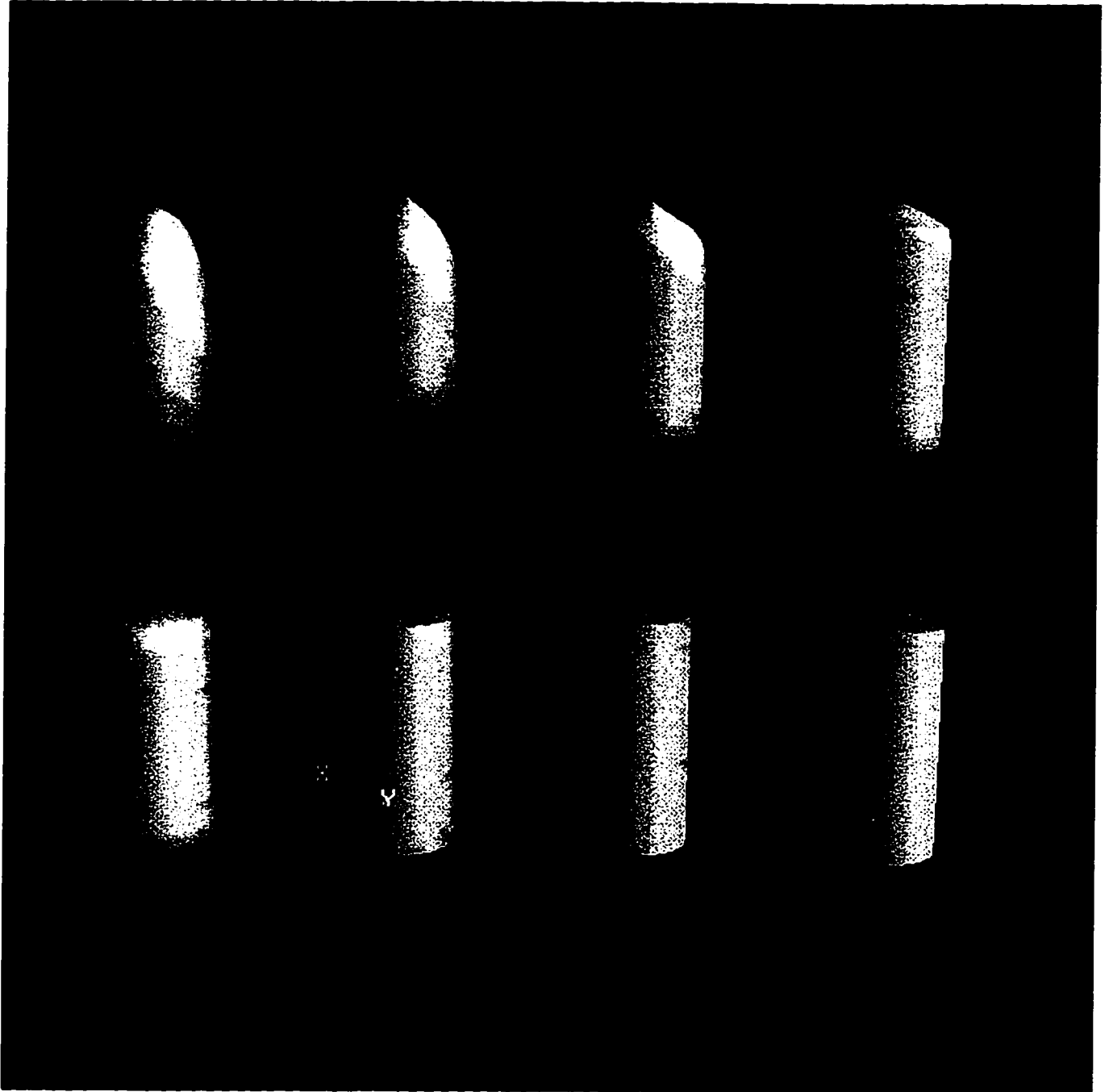




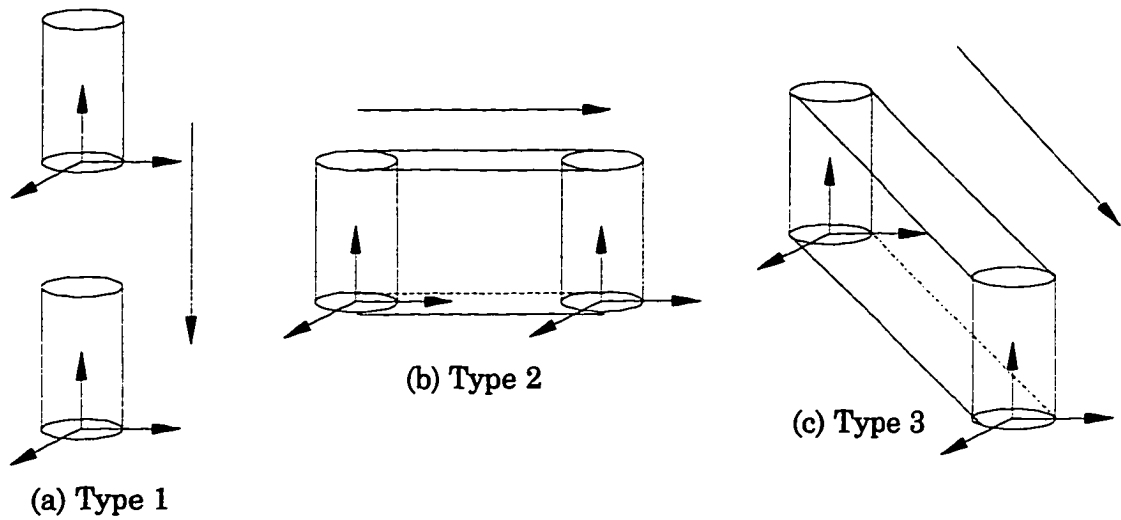
**Figure 4.2** General 7-parameter APT tool



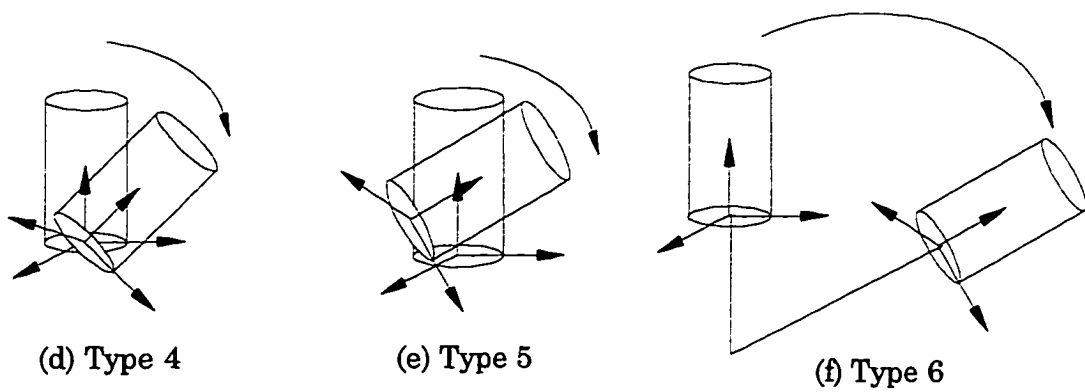
**Figure 4.3** Approximation of a cone surface in a single smooth function



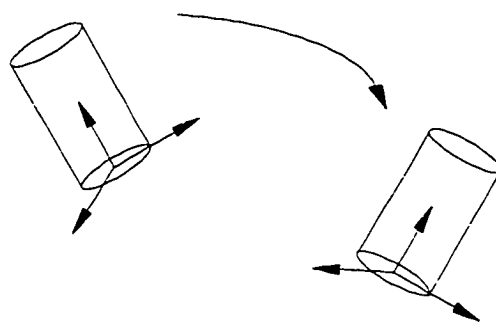
**Figure 4.4** Approximation of flat-end tool surface in a single smooth function



(a)-(c): Cutters under five-axis planar motions: translational

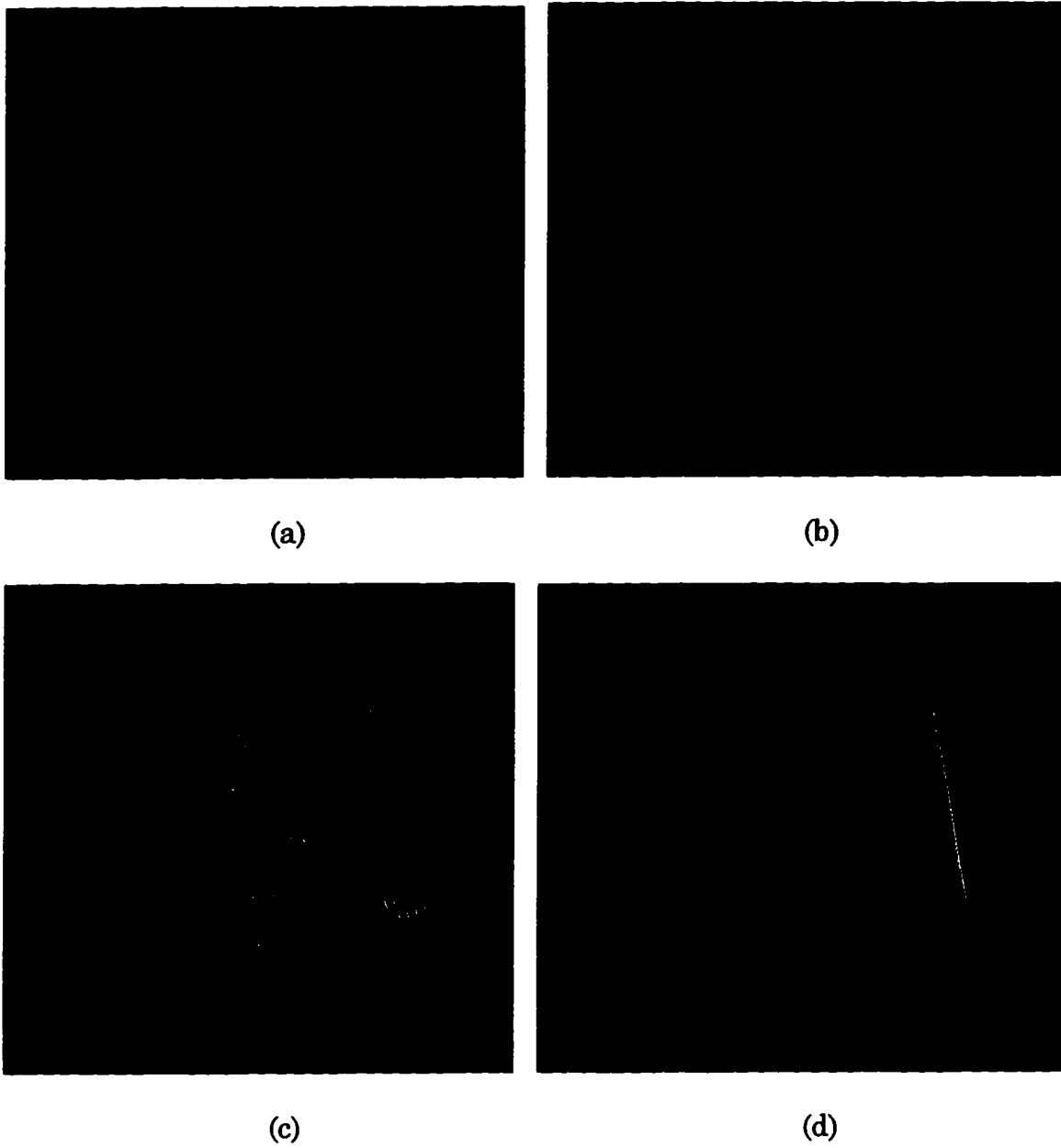


(d)-(f): Cutters under five-axis planar motions: rotational

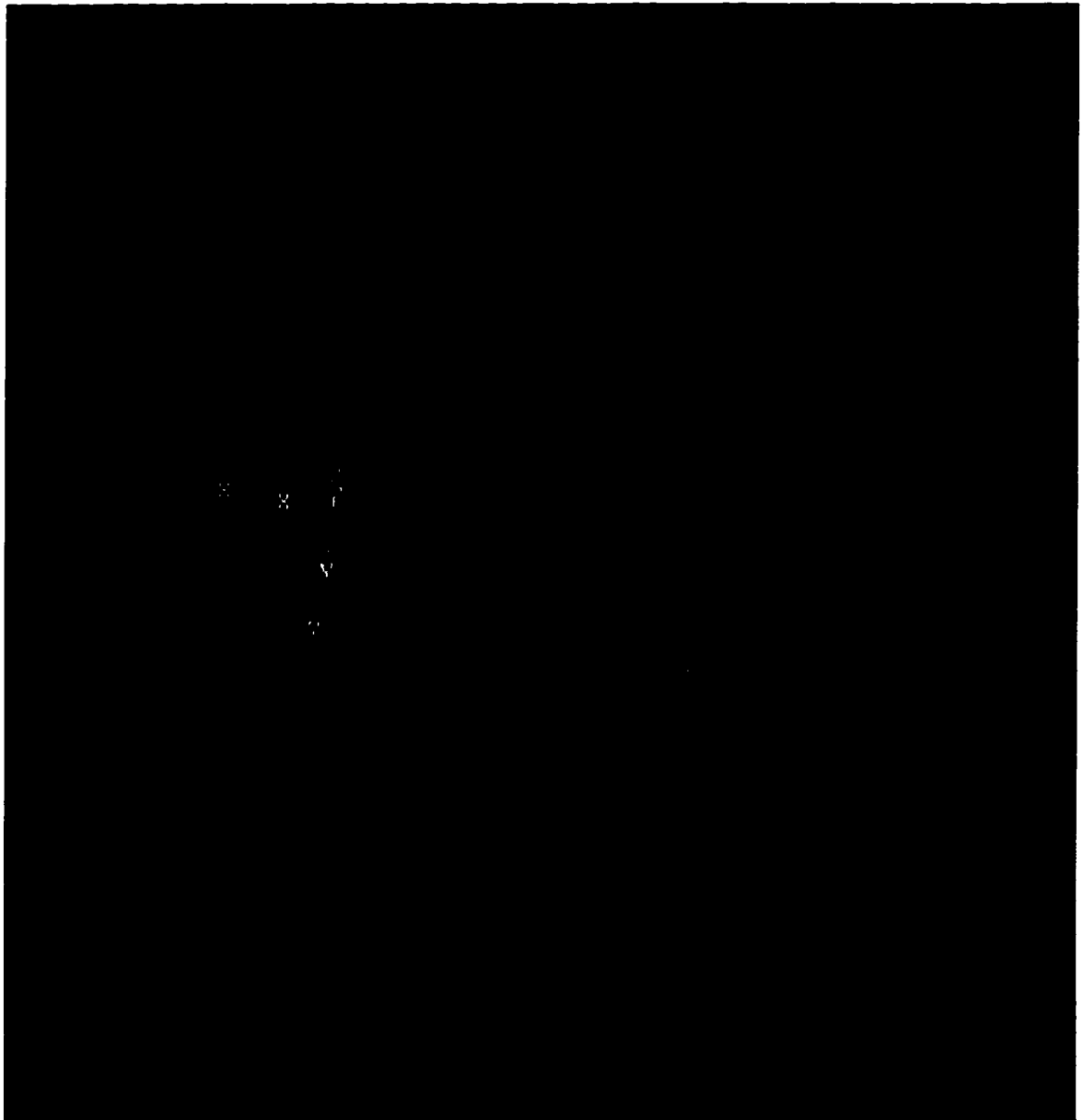


(g) Type 7: cutters under five-axis non-planar motion

**Figure 4.5:** Moving cutters under one-block five-axis motions



**Figure 4.6** Graphical output for Example 4.1



**Figure 4.7** Grazing points generated for Example 4.2



**Figure 4.8** Grazing curves generated for Example 4.2

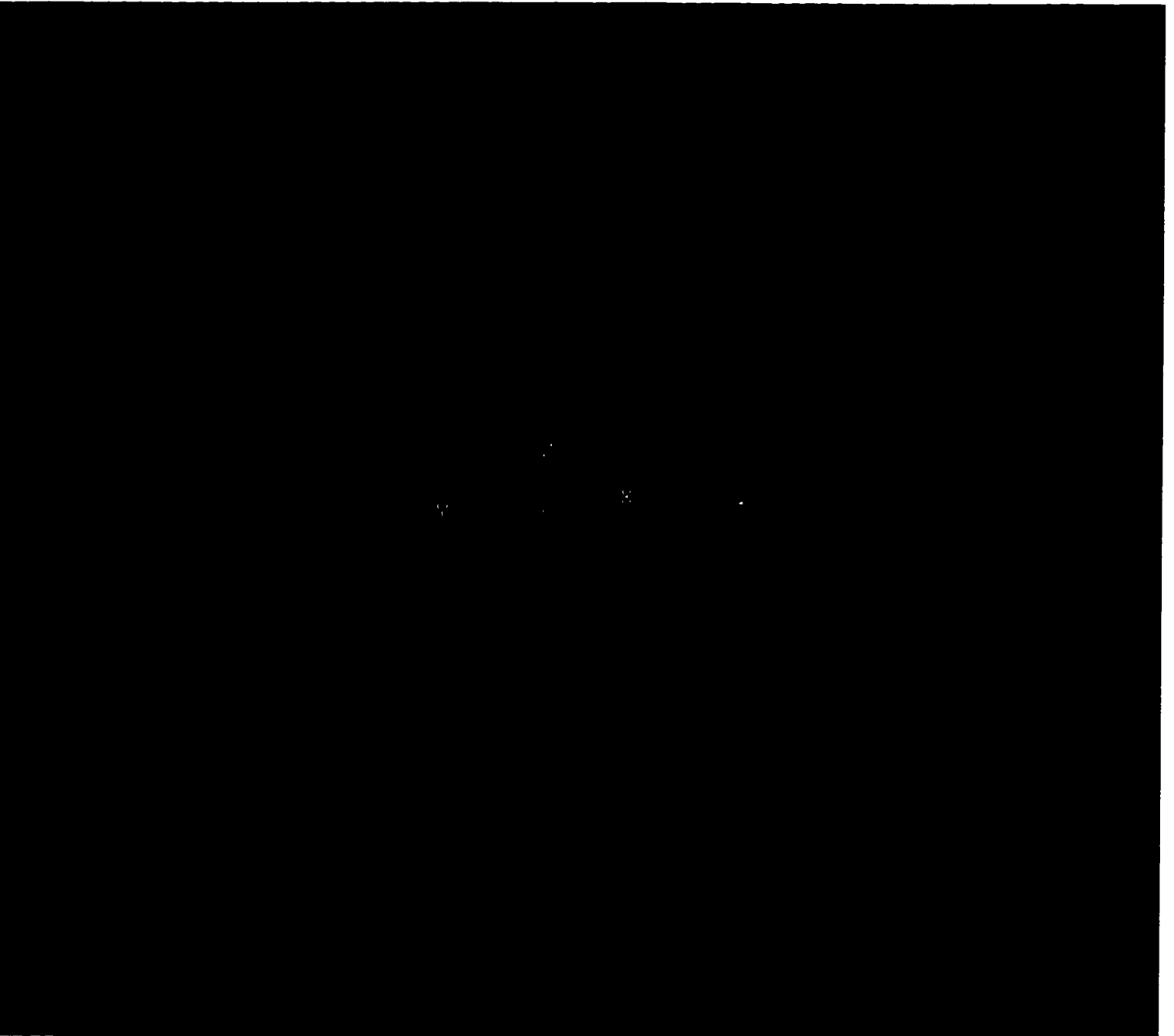


**Figure 4.9** Swept volume generated for Example 4.2

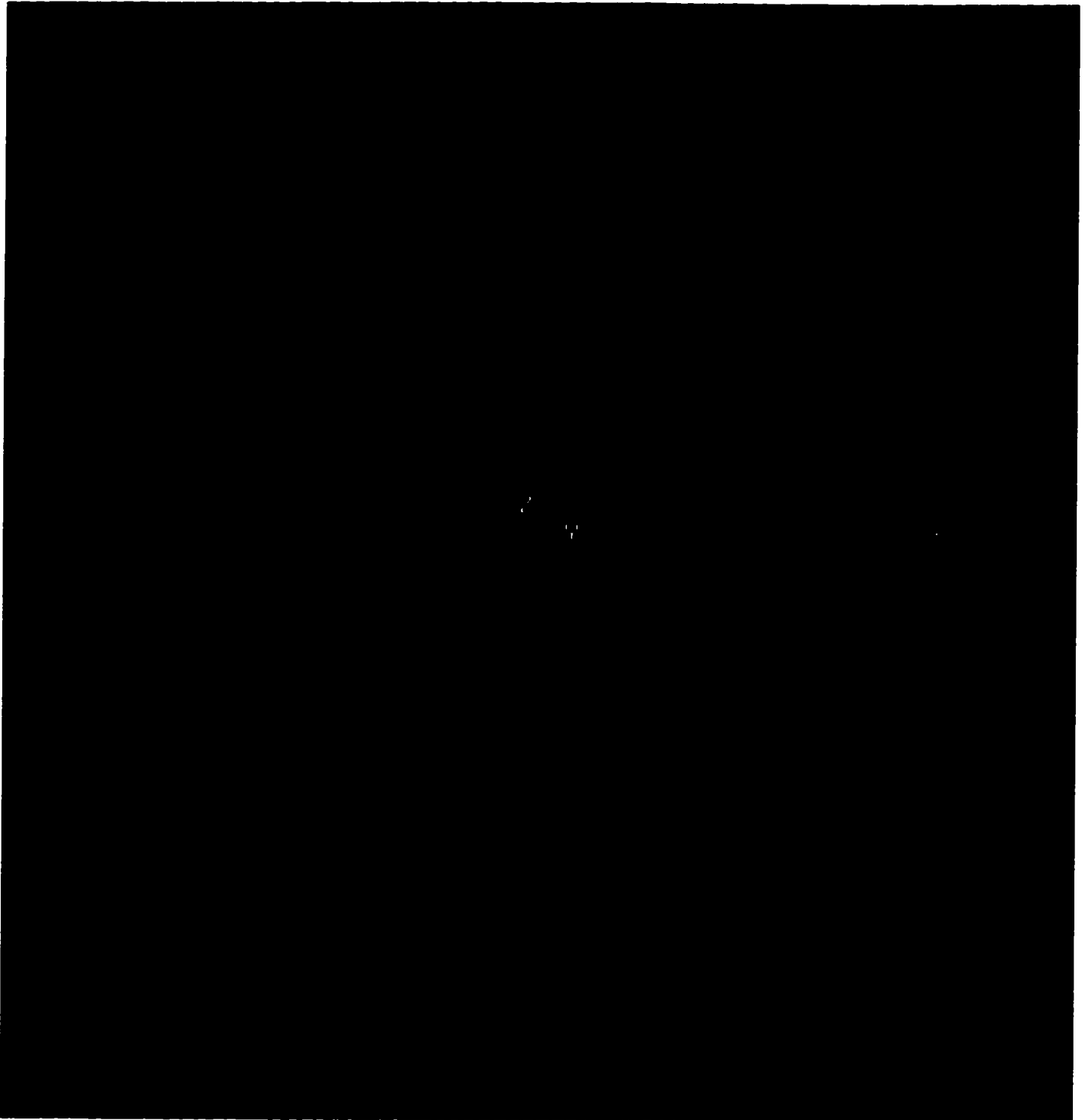




**Figure 4.10** TCP path and tool configurations for Example 4.3



**Figure 4.11** Grazing curves generated for Example 4.3



**Figure 4.12** Swept volume generated for Example 4.3

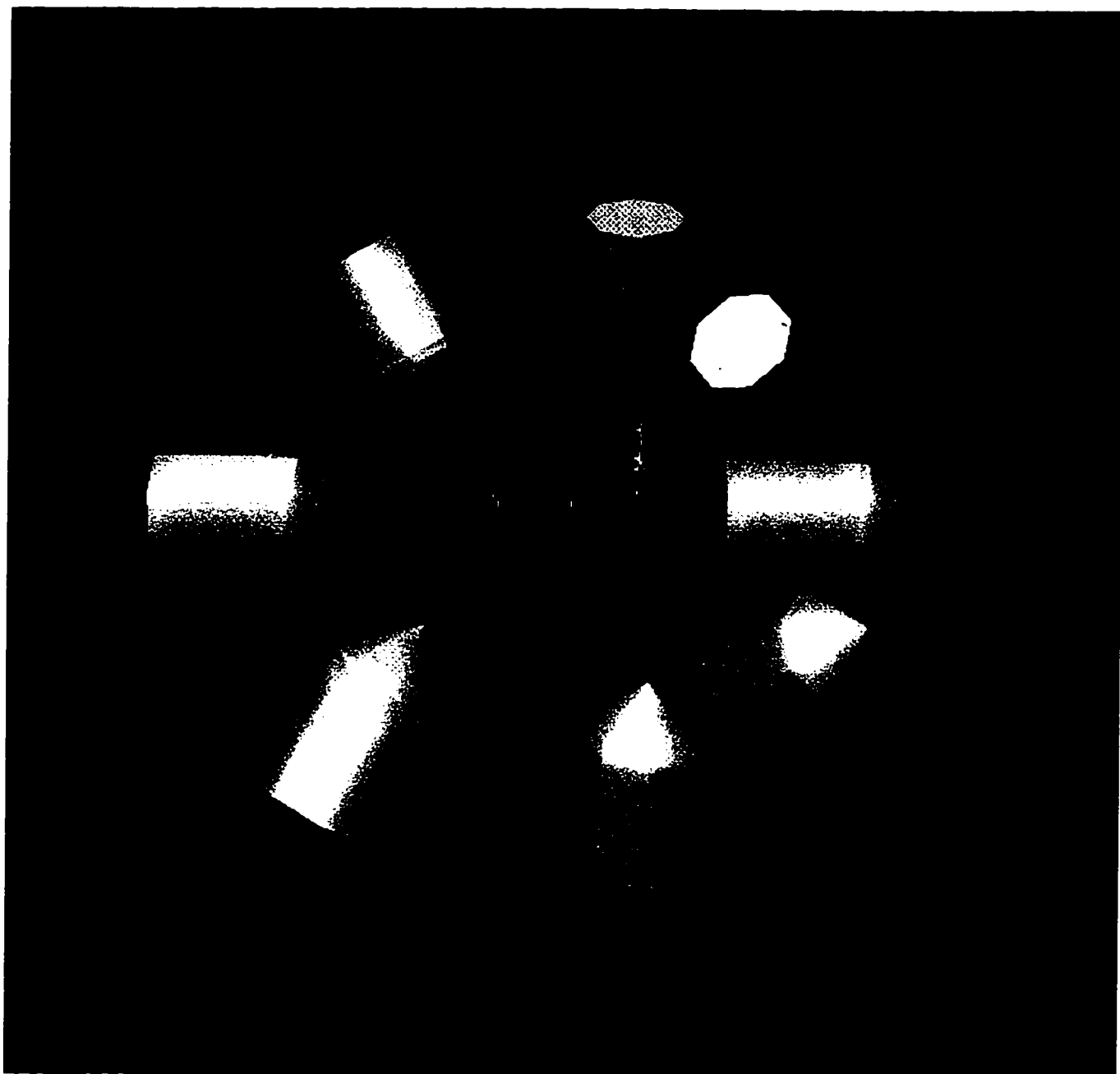
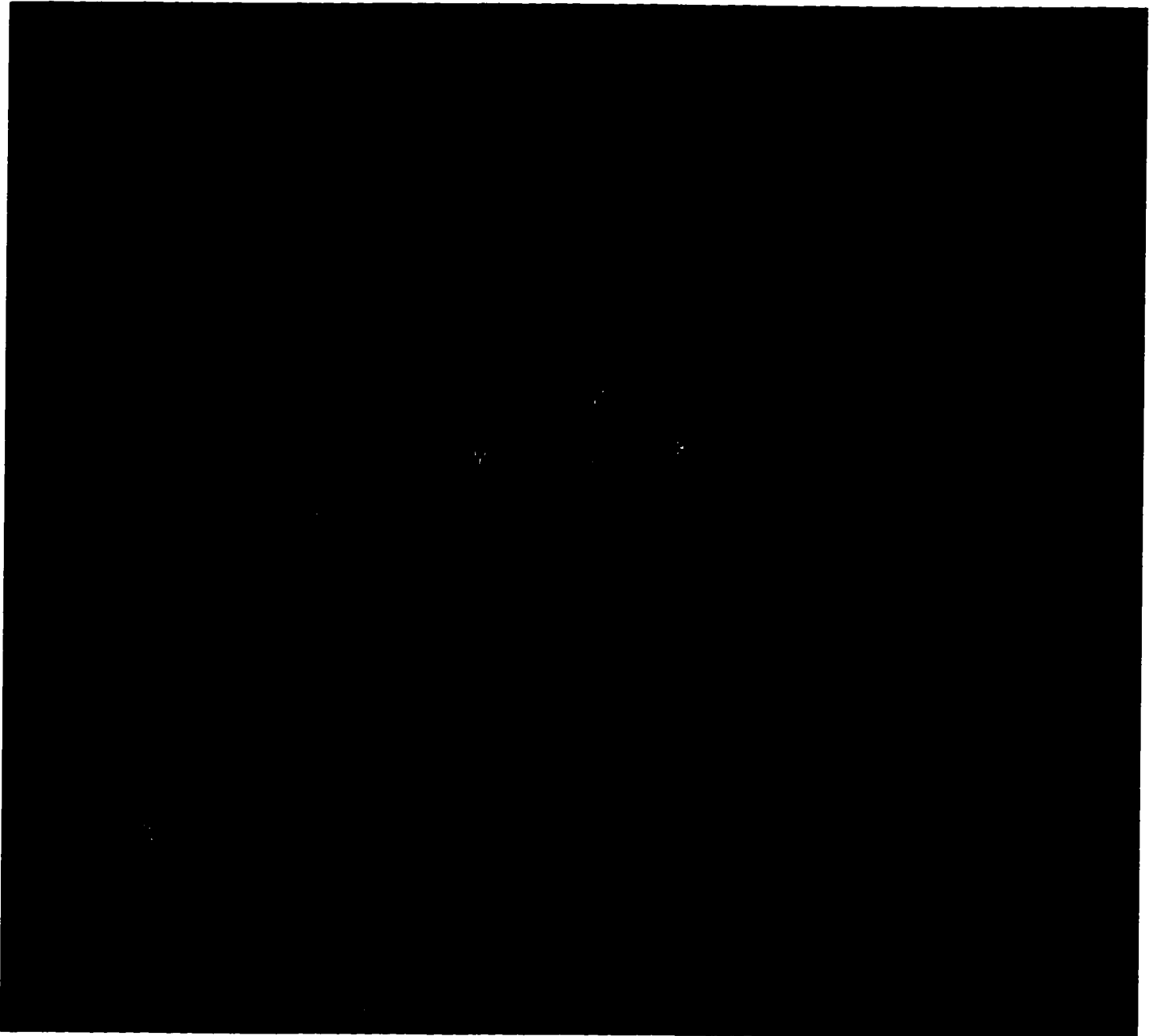


Figure 4.13 TCP path and tool configurations for Example 4.4



**Figure 4.14** Grazing curves generated for Example 4.4



**Figure 4.15** Swept volume generated for Example 4.4

## **CHAPTER 5**

### **APPLICATION OF SDE/SEDE APPROACH TO NC MACHINING AND COMPARISON WITH ENVELOPE METHOD**

An integrated SDE/SEDE swept volume study with part design, multi-axis NC tool path planning, NC program verification and actual NC machining is described in this chapter. First the sweep generator SDE-3D, described in chapter 3, is used to assist the study on determination of tool orientation in 5-axis tool path design of sculptured surfaces; then the sweep generator SWEEP-SEDE described in Chapter 4 is used to replace the sweep module in a commercial NC simulation system for material removal simulation of five-axis NC milling process. Finally, the SDE/SEDE swept volume research is incorporated with a five-axis NC milling CAD/CAM system at NJIT. From the implementation and application of the SDE/SEDE, a comparison of SDE/SEDE methods with the envelope method is presented.

#### **5.1 Integration of SDE Approach with Research on Five-Axis NC Tool Path Generation**

The SDE-3D sweep generator is integrated with a research project on the five-axis NC tool path generation of sculptured surfaces. The details of this project are included in a Deng, Leu and Wang's paper (Deng et al., 1996). In this section this project is briefly introduced, followed by a software integration and an illustrative example.

Generation of five-axis NC tool path is a two-step process: step one, calculation of inclination and yaw angles based on curvature machining theory; step two, verification of the calculated result to avoid over-cut or under-cut problems. In the Deng, Leu and Wang's work, step two is done by integrating the SDE sweep module and the Boolean subtractor in a commercial NC simulation software called Virtual NC<sup>TM</sup> (Deneb, 1995). With the help of CAD/CAM technologies, a researcher

can visually verify the correctness of the proposed cutter location data and make necessary adjustments.

#### **5.1.1 Determination of Flat-End Cutter Orientation in 5-Axis Machining by Curvature Matching Method**

The concept of curvature matching was first implemented by Jenson (Jenson et al., 1992,1995) to determine the flat-end cutter orientation for 5-axis NC machining of sculptured surfaces. But only inclination angle was considered in his study. Deng, Leu and Wang's method (called the extended curvature matching method) is one in which the full advantages of 5-axis machining are exploited by considering both the cutter's inclination and yaw capability and the local differential geometric features of a designed surface. The machining error, or cusp height, is derived for two machining scenarios.

An algorithm has been developed to compute the cutter orientation by matching the slope and curvature of the surface at each discretized point on the sculptured surface while minimizing the machining error and satisfying constraints on both inclination and yaw angles.

#### **5.1.2 Verification of CL Data Using SDE-3D Sweep Generator**

The inclination and yaw angles calculated using the extended curvature matching method has to be transformed to CL data using the 5-axis NC machining model described in Deng et al.'s paper. The SDE-3D sweep generator then takes the CL data as an input and calculates the swept volumes generated by the moving tool. To avoid the internal patches of a swept volume, the swept volumes generated by the bottom edge of a flat-end tool are computed and displayed. The Boolean subtraction of the swept volumes from the stock is performed using VNC.

It is found in surface machining simulation that although the extended curvature matching method leads to a more efficient and accurate tool path



generation, collision is a commonly encountered problem when using this method. Therefore, NC simulation and verification is necessary at the design stage. If a collision is found during the simulation process, the cutter location information associated with this collision is sent back to the machining process planner. The tool size has to be changed or some other methods have to be adopted to modify this incorrect cutter location.

### 5.1.3 Example

Fig. 5.1 provides an example illustrating how the extended curvature matching method in machining a sculptured surface can be integrated with the cutter swept-volume research to generate the flat-end cutter orientation and perform material removal simulation of a 5-axis NC milling process. The curve in the plane perpendicular to the cutter motion is described by  $z = x^2$ . The top left shows the bottom surface of the cutter in contact with various points of a sculptured surface. The top right shows the swept volume of the cutter traveling through these points. By taking the Boolean subtraction of the cutter swept volume from the stock (illustrated in the bottom left), the workpiece after this machining path is obtained as shown in the bottom right.

## 5.2 Integration of SEDE Approach with Commercial NC Software

In this section, integration of the SEDE research on swept volume with an existing NC simulation commercial software, called Virtual NC<sup>TM</sup>, is discussed. The SEDE sweep module is used to replace Virtual NC's convex hull sweep algorithm. After the integration, the machine tool simulation occurs in exactly the same fashion with the exception of the method used to create the geometric representation of the tool swept volume. The main advantages of this integration are: 1) to allow a more accurate

swept volume computation method to be utilized within VNC; 2) to provide an environment for visualizing and analyzing the results of the SEDE research.

### 5.2.1 About Virtual NC

Deneb's Virtual NC<sup>TM</sup> (VNC) (Deneb, 1995) is an interactive 3D simulation environment specifically for visualizing and analyzing the functionality of a machine tool, its CNC controller, and the material removal process. Machines, tooling and fixtures are rapidly modeled using the built-in component libraries with their associated degrees of freedom. "Mimic", Virtual NC's configurable controller emulator, is capable of reproducing the functionality of a CNC controller. An NC programmer can evaluate the actual machine controller and different processing techniques without risking damages to the actual machine tool. Cycle information, such as depth of cut, axis position, speed, feedrate, cycle time, can be obtained by the material removal process simulation. Virtual NC automatically detects near misses and collisions and specifies collision tolerances. An NC program can be monitored and edited when the simulation is being done. Process evaluation and refinement can be done by VNC functions.

### 5.2.2 Integration Procedures

In NC verification, the geometry of the machined part can be expressed as the difference between the stock and the swept volume of the cutter. So a sweep generator, incorporated with a Boolean subtractor and a collision detector, can be used to simulate the material removal process of 5-axis NC milling. In this work, Virtual NC<sup>TM</sup> (VNC) is chosen as the Boolean subtractor and display platform. The SWEEP-SEDE sweep generator, described in chapter 4, is used to replace the convex

hull sweep module in VNC for a more accurate swept volume representation. The process is illustrated in Fig. 5.2.

The SEDE module computes the swept volume using current and previous locations of the tool. But VNC takes NC codes as its input. So a transformation of MCD data to CL data by a kinematics analysis of the machine or an inquiry of tool position using VNC's CLI (Command Line Interpreter, a language) calls is necessary for the integration. The geometric representation of the swept volume is then passed to VNC. SEDE then commands VNC to perform the Boolean operation to subtract the swept volume of the tool from the workpiece. Since this process is needed at each simulation update, rapid communication between VNC and SEDE is essential. The modification to VNC at the core source level has been made by Deneb Robotics. Dynamic Shared Object (DSO) communication has been added to VNC by the company to facilitate communication between VNC and SEDE. A custom application interface (API) and a layer of functions have also been created for use by the SEDE module.

In order to facilitate this integration, a step has to be added to the SWEEP-SEDE program that translates CL data into the general motion characterization of sweeps (e.g., general motion equation described in Chapter 4), and oriented output data has to be provided for the polygonal (triangular) decomposition of the approximate swept volume in the form that is used in VNC. VNC uses the swept volume data together with a Boolean subtraction algorithm to simulate the material removal on a workpiece subjected to the cutter motion. A GSL (Graphical Simulation Language) program has been written to display the swept volume generated by the SEDE sweep generator in the workcell of VNC. Since VNC simulates and verifies an NC program in a machining environment, an actual milling machine (FADAL VMC-20) has been modeled and analyzed by kinematics analysis. By using the Boolean subtractor and verifier in Virtual NC, the material removal process is simulated and

analyzed in an interactive 3D simulation environment. Spatial errors in NC programs (such as gouging and misses) are analyzed and computed.

### 5.2.3 Modeling and Kinematics Analysis of FADAL VMC-20

Since VNC simulates the NC processes in machining environments, an actual machine, FADAL VMC-20, is modeled and analyzed in this section. The general kinematics analysis procedures (Appendix A) for five-axis NC milling machines have to be followed.

FADAL CMC-20 is a Type B machine (Fig. 5.3) with a fixed configuration. It is modeled based on its geometry and kinematics attributes using VNC's machine modeling libraries (Fig. 5.4). Denavit-Hartenberg notation is introduced to derive the motion equation and the transformation between CL and MCD data (Fig. 5.5). The parameters for the Denavit-Hartenberg notation are as follow:

Link	frame	$\alpha_i$	$\theta_i$	$a_i$	$d_i$
0	$xyz$				
1	$x_1y_1z_1$	$-90^0$	$180^0 + A$	0	$-d_1$
2	$x_2y_2z_2$	$90^0$	$90^0 - B$	$-d_2$	0
3	$x_3y_3z_3$	$90^0$	0	$-d_3$	$X$
4	$x_4y_4z_4$	$90^0$	$90^0$	0	$Y + d_4$
5	$x_5y_5z_5$	0	0	0	$Z + d_5$

**Table 5.1** Denavit-Hartenberg parameters for FADAL VMC-20 machine

#### Sweep equation

By multiplying the transformation matrices of all links, the transformation matrix between the tool coordinate frame and the reference coordinate frame is obtained as

$$T_i = \begin{pmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = A_1 A_2 \dots A_5 A_t \quad (5.1)$$

where

$$A_1 = \begin{pmatrix} -\cos A & 0 & \sin A & 0 \\ -\sin A & 0 & -\cos A & 0 \\ 0 & -1 & 0 & -d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.2)$$

$$A_2 = \begin{pmatrix} \sin B & 0 & \cos B & -d_2 \sin B \\ \cos B & 0 & -\sin B & -d_2 \cos B \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.3)$$

$$A_3 = \begin{pmatrix} 1 & 0 & 0 & -d_3 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & X \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.4)$$

$$A_4 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & Y + d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.5)$$

$$A_5 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & Z + d_5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.6)$$

$$A_t = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -d_6 \\ 0 & 0 & 1 & -h \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.7)$$

Separating the transformation matrix into a translational component and a rotational component and using the regular (instead of homogeneous) representation, the equation representing the trajectory of any point in the tool coordinate frame is obtained:

$$\mathbf{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} + \begin{pmatrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{pmatrix} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix} \quad (5.8)$$

where the vector  $\mathbf{r} = (x, y, z)^T$  represents any point in the reference coordinate frame  $O-xyz$  and  $\mathbf{r}' = (\xi, \eta, \zeta)^T$  represents any point in the tool coordinate frame  $C-\xi\eta\zeta$ .

For this particular machine, the sweep equation is:

$$\begin{aligned} \mathbf{r} &= \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} \\ &+ \begin{pmatrix} -\cos B \cos A & -\sin A & -\sin B \cos A \\ -\cos B \sin A & \cos A & -\sin B \sin A \\ \sin B & 0 & -\cos B \end{pmatrix} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix} \end{aligned} \quad (5.9)$$

where

$$\begin{aligned} p_x &= -X \cos B \cos A + d_2 \cos A \sin B + d_3 \cos A \sin B + h \cos A \sin B \\ &\quad - (Z + d_5) \cos A \sin B + d_6 \sin A - (Y + d_4) \sin A \end{aligned} \quad (5.10)$$

$$\begin{aligned} p_y &= -d_6 \cos A + (d_4 + Y) \cos A - X \cos B \sin A + d_2 \sin B \sin A \\ &\quad + d_3 \sin B \sin A + h \sin B \sin A - (d_5 + Z) \sin B \sin A \end{aligned} \quad (5.11)$$

$$p_z = -d_1 + d_2 \cos B + d_3 \cos B + h \cos B - (Z + d_5) \cos B + X \sin B \quad (5.12)$$

### Transformation between CL data and MCD data

From the above sweep equation (5.9), the tool axis has direction  $(0, 0, 1)$  in the tool coordinate frame and  $(I, J, K)$  in the reference coordinate frame; therefore

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \cos B & 0 & -\sin B \\ 0 & 1 & 0 \\ \sin B & 0 & \cos B \end{pmatrix} \begin{pmatrix} \cos A & -\sin A & 0 \\ \sin A & \cos A & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} I \\ J \\ K \end{pmatrix} \quad (5.13)$$

where  $(I, J, K)$  represents the normalized directional cosines.

Solving (5.13) for  $I, J, K$ ,

$$I = \frac{\cos A}{\sin^2 A + \cos^2 B + \sin^2 B \cos A} \quad (5.14)$$

$$J = -\frac{\sin A}{\sin^2 A + \cos^2 B + \sin^2 B \cos A} \quad (5.15)$$

$$K = \frac{1}{\cos B} + \frac{\tan B (\sin^2 A - \cos^2 A)}{\sin^2 A + \cos^2 B + \sin^2 B \cos A} \quad (5.16)$$

Solving (5.13) for  $A$  and  $B$ ,

$$A = \tan^{-1}\left(-\frac{J}{I}\right) \quad (0 \leq A \leq 2\pi) \quad (5.17)$$

$$B = \tan^{-1}\left(-\frac{\sqrt{1-K^2}}{K}\right) \quad \left(-\frac{3\pi}{2} \leq B \leq 0\right) \quad (5.18)$$

From the above sweep equation (5.9), noticing that the control point  $C$  has coordinates  $(0,0,0)$  in the tool coordinate frame and  $(x_c, y_c, z_c)$  in the reference coordinate frame, it is found that

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} + \begin{pmatrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (5.19)$$

$$\begin{aligned} x_c = & -X \cos B \cos A + d_2 \cos A \sin B + d_3 \cos A \sin B + h \cos A \sin B \\ & -(Z + d_5) \cos A \sin B + d_6 \sin A - (Y + d_4) \sin A \end{aligned} \quad (5.20)$$

$$\begin{aligned} y_c = & -d_6 \cos A + (d_4 + Y) \cos A - X \cos B \sin A + d_2 \sin B \sin A \\ & + d_3 \sin B \sin A + h \sin B \sin A - (d_5 + Z) \sin B \sin A \end{aligned} \quad (5.21)$$

$$z_c = -d_1 + d_2 \cos B + d_3 \cos B + h \cos B - (Z + d_5) \cos B + X \sin B \quad (5.22)$$

Solving (5.19) for  $X, Y, Z$ ,

$$\begin{aligned} X = & -\csc B (d_1 - d_2 \cos B - d_3 \cos B + d_5 \cos B - h \cos B - z_c) \\ & -\cot B \{ \cos B (d_1 - d_2 \cos B - d_3 \cos B + d_5 \cos B - h \cos B - z_c) \\ & + \sin B \cos A (-x_c + d_2 \cos A \sin B + d_3 \cos A \sin B - d_5 \cos A \sin B \\ & + h \cos A \sin B - d_4 \sin A + d_6 \sin A) - \sin B \sin A (d_4 \cos A \\ & - d_6 \cos A - y_c + d_2 \sin B \sin A + d_3 \sin B \sin A - d_5 \sin B \sin A \\ & + h \sin B \sin A) \} \end{aligned} \quad (5.23)$$

$$\begin{aligned} Y = & \sec A (d_4 \cos A - d_6 \cos A - y_c + d_2 \sin B \sin A + d_3 \sin B \sin A \\ & - d_5 \sin B \sin A + h \sin B \sin A) \\ & + \sin B \tan A [ \cos B (-d_1 + d_2 \cos B + d_3 \cos B - d_5 \cos B + h \cos B - z_c) \\ & - \sin B \cos A (-x_c + d_2 \cos A \sin B + d_3 \cos A \sin B - \\ & - d_5 \cos A \sin B + h \cos A \sin B - d_4 \sin A + d_6 \sin A) \\ & - \sin B \sin A (d_4 \cos A - d_6 \cos A - y_c + d_2 \sin B \sin A + \\ & d_3 \sin B \sin A - d_5 \sin B \sin A + h \sin B \sin A) ] \end{aligned}$$

$$\begin{aligned}
& -\tan A[\cos B \csc B(-d_1 + d_2 \cos B + d_3 \cos B - d_5 \cos B + h \cos B - z_c) \\
& + \cot B \cos B(-d_1 + d_2 \cos B + d_3 \cos B - d_5 \cos B + h \cos B - z_c) \\
& + \sin B \cos A(-x_c + d_2 \cos A \sin B + d_3 \cos A \sin B \\
& + d_5 \cos A \sin B + h \cos A \sin B - d_4 \sin A + d_6 \sin A) \\
& + \sin B \sin A(d_4 \cos A - d_6 \cos A + d_2 \sin B \sin A \\
& + d_3 \sin B \sin A - d_5 \sin B \sin A + h \sin B \sin A)] \quad (5.24)
\end{aligned}$$

$$\begin{aligned}
Z = & \cos B(-d_1 + d_2 \cos B + d_3 \cos B - d_5 \cos B + h \cos B - z_c) \\
& + \sin B \cos A(-x_c + d_2 \cos A \sin B + d_3 \cos A \sin B \\
& - d_5 \cos A \sin B + h \cos A \sin B - d_4 \sin A + d_6 \sin A) \\
& + \sin B \sin A(d_4 \cos A - d_6 \cos A - y_c + d_2 \sin B \sin A \\
& + d_3 \sin B \sin A - d_5 \sin B \sin A + h \sin B \sin A) \quad (5.25)
\end{aligned}$$

#### 5.2.4 Illustrative Examples

After the integration of SEDE program with VNC, the integrated system can simulate and verify any given NC codes for five-axis NC milling of sculptured surfaces. Other inputs for this integrated system include the simulated five-axis machine workcells, stocks, fixtures and tools. Two examples are given to illustrate the capabilities and the effectiveness of this integrated five-axis NC simulation and verification system.

##### Example 5.1

This example shows a simulation of a four-block motion of a flat-end tool in NC machining. It illustrates the procedure employed to integrate the SWEEP-SEDE program with VNC. The motion of each block is a typical five-axis motion.

The tool surface representation and the input are the same as in Example 4.1 except that the tool radius is  $r = 0.25$  inches and its length is  $h = 5$  inches.



The workcell is modeled based on an actual five-axis NC milling machine (FADAL VMC-20). Illustrations of the swept volumes generated by SWEEP-SEDE and the material removal after Boolean subtraction as produced by the graphical output from VNC are shown in Fig.5.6.

Fig.5.6(a) shows the machining setup and workpiece before machining. Fig.5.6(b) shows five given tool configurations of the flat-end cutter during the machining process. Each block undergoes a general five-axis motion with both translation and rotation. The grazing solids of the cutter swept volumes calculated by SWEEP-SEDE after triangulation and shading are shown in Fig.5.6(c). The cutter swept volume is the union of the five tool configurations (Fig.5.6(b)) and the four grazing solids (Fig.5.6(c)). Fig.5.6(d) shows the machined workpiece, which is obtained by performing Boolean subtraction of the cutter swept volumes from the stock using VNC software.

### Example 5.2

This example shows a simulation of machining one blade of a turbine impeller in rough cut using a flat-end tool with radius 0.125 inches and length 3.0 inches. It is used to illustrate further integration of the SWEEP-SEDE with VNC. It is similar to Example 5.1 and the cutter swept volumes and material removal produced by the graphical output from the commercial software are shown in Fig.5.7.

The NC codes to be verified are generated by commercial software and are assumed to be known here and this will be discussed in the next section.

The simulated workcell of the FADAL VMC-20 milling machine with a cylindrical stock is shown in Fig.5.7(a). Fig.5.7(b) shows the tool configurations at certain positions along the machining path. By using the SWEEP-SEDE module, the grazing solids of the cutter are computed and unioned with the tool configurations (illustrated in Fig.5.7(b)) as displayed in Fig.5.7(c). The machined workpiece, as illustrated in

Fig.5.7(d), is obtained by subtracting the tool swept volume (Fig.5.7(c)) from the stock.

In this example, only one critical part of the process of machining one blade of a turbine impeller is illustrated. The process for finishing the blade and machining the other blades can be performed similarly.

### 5.3 Integration of SEDE Sweep Module with 5-Axis NC Milling CAD/CAM System

In this section, the SEDE swept volume study described above has been interfaced with an integrated 5-axis NC milling CAD/CAM system at NJIT. Actual machining of a turbine impeller is presented.

#### 5.3.1 System Description

The integrated CAD/CAM system of 5-axis NC machining developed at NJIT includes a CAD software for part design, a multi-axis tool path generator, a post processor for transformation of cutter location data to machine control data, NC simulator/verifier, a FADAL VMC-20 five axis NC milling machine (Fig. 5.3) and a coordinate measurement machine. A schematic diagram of this system is given in Fig. 5.8.

**CAD software for design:** Pro/Engineer<sup>TM</sup> from Parametric is chosen to model the 3D part. Key features of Pro-E include powerful parametric design, unified CSG format.

**Rapid prototyping:** The designed part by Pro-E can be verified by transforming the geometry of the part to an SLA file and loading it to the 3D System's SLA-250 at NJIT to build a physical SLA model using a laser stereolithography technique.

**Tool path generation and simulation:** Parametric's Pro/Manufacture<sup>TM</sup> is used to generate the cutter location data for a given designed part. The boundary

of the to-be-machined part, machining parameters and machining strategies have to be determined by the programmer. More advanced CL generation methods have been developed by Deng et al. (Deng et al., 1996) by combining the curvature of the machined part surface and tool geometry. The same software can be used to simulate the NC milling process using the cutter location data in a stock-tool environment. This can only be taken as a visual verification of CL data.

**NC code generation by post processor:** After the generation of the cutter location data, a post processor is needed to transform the CL data into machine axis data. The milling machine used here is a FADAL VMC-20. ICAM<sup>TM</sup> is chosen as the postprocessor to transform CL data to NC code which is accepted by the FADAL VMC-20 for actual machining.

**NC program simulator/verifier:** Virtual NC<sup>TM</sup> from Deneb Robotics, Inc. and the SEDE sweep generator developed by the swept volume study are combined to simulate and verify the NC code generated by ICAM in a machining environment. To facilitate this process, a FADAL VMC-20 workcell was modeled and its kinematics attributes were established using the original machine model. By using the Boolean subtractor and verifier in Virtual NC, material removal process is simulated and analyzed in an interactive 3D simulation environment. Spatial errors in NC programs (such as gouging and misses) and machining cycle information (such as cycle times, depth of cut, axis speed, etc.) are detected and measured automatically. Process evaluation and refinement can be performed by functions in Virtual NC.

**Five-axis machining by FADAL VMC-20:** The NC program after the verification by the CAD/CAM system is downloaded to the FADAL VMC-20's controller for actual machining.

**Machining error estimate:** In this CAD/CAM system, the geometry of the machined part may be checked by a coordinate measurement machine at the Center for Manufacturing Systems of NJIT and be compared with the designed part.

### 5.3.2 Case Study : Turbine Impeller

The design, tool path generation, NC code verification, and manufacturing of a turbine impeller using NJIT's five-axis NC milling CAD/CAM system are presented as a case study in this section.

Turbine impellers have wide applications in shipbuilding, automotive, and aircraft industries. Intensive attention has been paid to designing a high-efficiency profile (Korakianitis, 1993; Wilson, 1984) and actual multi-axis NC machining of a turbine impeller. Since a turbine blade is composed of sculptured surfaces, currently available techniques for multi-axis NC machining of sculptured surface (Takeuchi, 1992; Li & Jerard, 1994; Liu, 1995) may be adopted. It has been proven that five-axis machining of a sculptured surface is one of the most difficult problems for NC programmers due to the complex surface curvature and tool movement. So simulation and verification of the NC program generated manually or automatically becomes essential in machining a turbine impeller.

Parametric design of the turbine blade profile is done using Pro/Engineer based on the theory described in (Wilson, 1984). Fig. 5.9 shows the designed part. The designed part may be saved in IGES format for part transformation among CAD/CAM software, using Pro/E's data exchange function. This part is also saved into a SLA file and is sent to the 3D System SLA-250 at NJIT for part generation. The cutter location data is verified by Pro-Manufacture in a stock-tool environment. The NC codes are checked and analyzed by Virtual NC in a machining environment which includes the tool, workpiece and machine. Fig. 5.10 shows the simulation in the process generated by VNC.

The designed stock is a cylinder with diameter 3.0 inches and length 3.0 inches. The stock is machined using a flat-end tool with length 3.0 inches and diameter 0.25 inches. Illustrations of the swept volumes and the material removal process produced by the graphical output from the commercial software are shown in Fig.5.11. Fig.5.11(a) shows the simulated machining workcell before machining. Fig.5.11(b) shows some tool configurations of the flat-end cutter. By using the SEDE module, the grazing points of the cutter swept volume after triangulation and connection are shown in Fig.5.11(c). Fig.5.11(d) shows the machined workpiece, which is obtained by performing Boolean subtraction of the cutter swept volume from the stock.

The verified NC codes are then loaded to FADAL VMC-20 for actual machining. The machined parts are shown in Fig. 5.12.

#### **5.4 Comparison of SDE/SEDE Methods with Envelope Method**

Since very few exact analyses have been carried out for other swept volume methods, a quantitative comparison of computational cost is not possible. Thus one can only compare the qualitative efficiency of these methods. The SEDE method is qualitatively more efficient since it requires computation of points on the swept volume boundary only at the initial and final position of the object. The SDE/SEDE methods and the envelope method are two of the important swept volume modeling methods for boundary representation of a swept volume. So attention is focused on comparison of these two methods in the following sections.

##### **5.4.1 General Comparison**

The envelope method (Wang & Wang, 1988; Sambandan, 1988, 1990; Narvekar, 1991) models the swept volume by the solution of the following two normally non-

linear envelope equations:

$$F(x, t) = 0$$

$$F_t(x, t) = 0$$

Solving the above non-linear envelope equations either analytically or numerically is a mathematically difficult task. To solve the envelope equations, one typically uses an algorithm for solving nonlinear equations like Newton's method. In some cases, when the equations are complicated and have a widely varying Jacobian, convergence is either very slow or cannot be efficiently achieved (Elden & Wittmeyer-Koch, 1990). For the SEDE method, convergence is automatically achieved via a Runge-Kutta method.

To facilitate the envelope method, first an object in space is decomposed into subsets of curves or surfaces (called generators) which can be represented by a family of curves or surfaces parametrically. Then the envelope equations are used to determine the critical surface patches corresponding to relevant generators. The final boundary of a swept volume is contained in the critical surface patches which most likely intersect each other. For a complex object which consists of a large number of generators, the computational cost will be dramatically high. The envelope method proves that the boundary of a swept volume is formed by some self-intersecting envelope surface patches but does not address how to do the trimming. It relies on the hidden surface removal technique in image space for swept volume display.

The SDE method requires an object surface to be piecewise smooth. An object can be represented by parametric equations or a discrete point set and shows advantages in dealing with complex or non-smooth objects. So far the SEDE method has only been applied to smooth objects. Commonly used cutter geometries like general 7-parameter APT tools are approximated by smooth functions with accuracy control. In the SDE/SEDE study, the trimming problem is formulated, classified,

summarized into algorithms (see Chapter 2 and 4) and has been implemented in the two-dimensional case. It has been pointed out in the SEDE study that five-axis motion can be decomposed into small piecewise smooth motions where no trimming is needed.

#### **5.4.2 Numerical Calculation: Computer Program**

The three-dimensional SDE/SEDE sweep generators (SDE-3D and SWEEP-SEDE) developed in this dissertation are analogous to the envelope sweep generators (SWEEP-2D and SWEEP-3D) developed at Cornell University (Sambandam, 1988, 1990) in terms of the input, output, object representation and swept volume display. But our 3D SDE/SEDE sweep generators are different from the envelope sweep generators as follows:

(1) Five-axis motion description. The program for envelope method adopts CL interpolation only and the 3D SDE/SEDE programs incorporate both CL and MCD interpolations to describe the actual motion of a five-axis NC milling controller between any two machine configurations;

(2) Swept volume representation. The envelope programs represent a swept volume in image space for visual display and therefore may not be accurate enough for collision detection and other applications. The SEDE sweep program represents a swept volume as a solid and supports any standard CAD data transformation format (i.g., IGES, SLA) for the data transformation among CAD/CAM software.

(3) Complexity and error analysis. The time complexity and upper error bound of the SEDE algorithm has been analyzed. But no such analysis is given in the envelope programs.

(4) Application in actual NC verification. The SEDE program has been integrated with commercial CAD/CAM software and systems to simulate NC milling processes in a machining environment with high-quality graphical display

and animation. Using this system, both CL data and NC codes may be verified visually with some automatic verification functions. In contrast, the envelope program has not been proven useful in real NC applications; it only supports CL data verification in stock-tool environments and is basically for visual verification of gross errors.

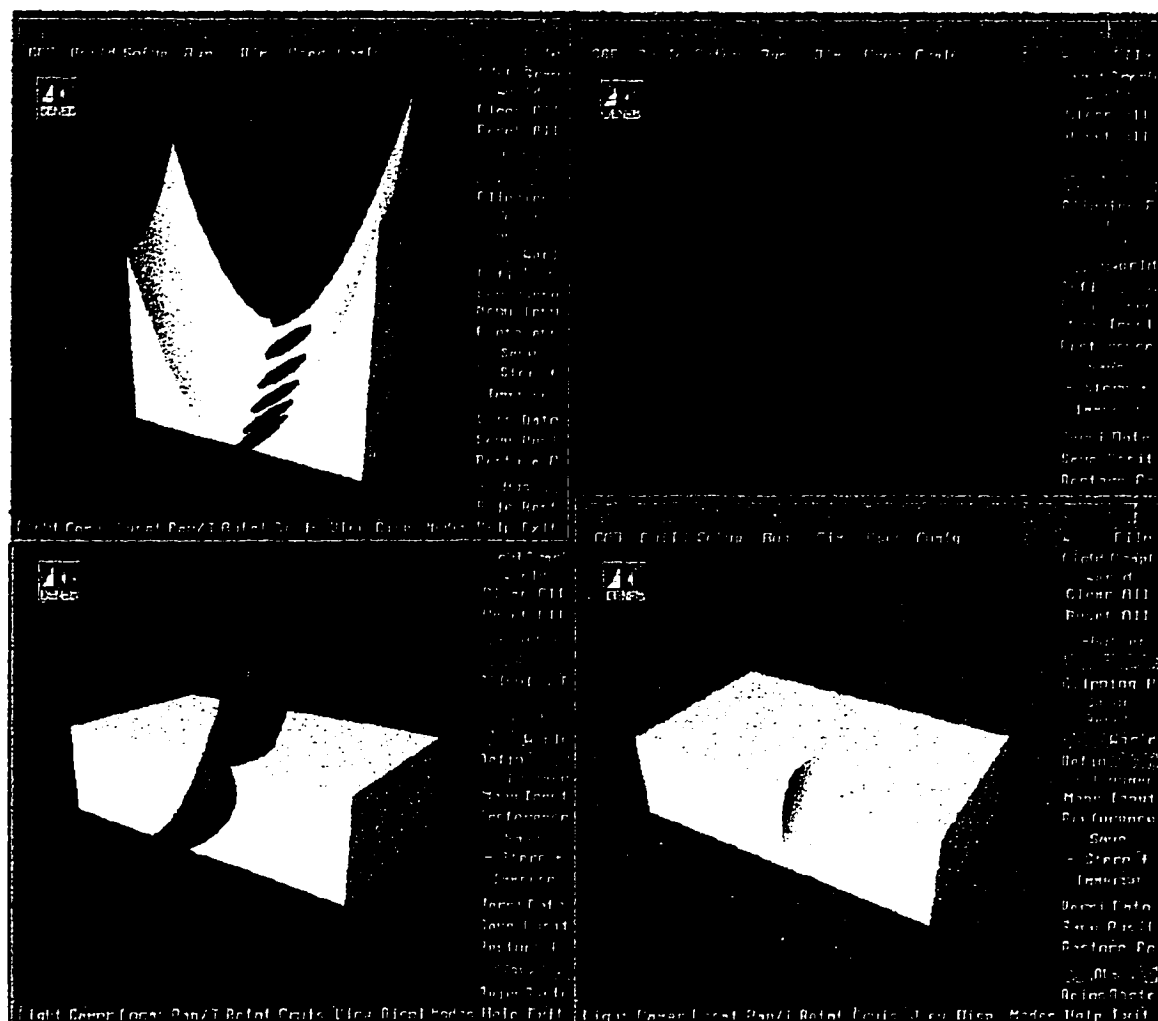
### 5.5 Final Remarks

The SEDE method appears to provide a useful tool for computing swept volumes in space that incorporates both the geometry of the object and the sweep. It shows advantages in actual implementation and application compared with the envelope method. The integration of the SEDE approach with commercial NC software improves the performance of the software by providing a more accurate swept volume representation and error estimates that are especially important in multi-axis NC machining verification.

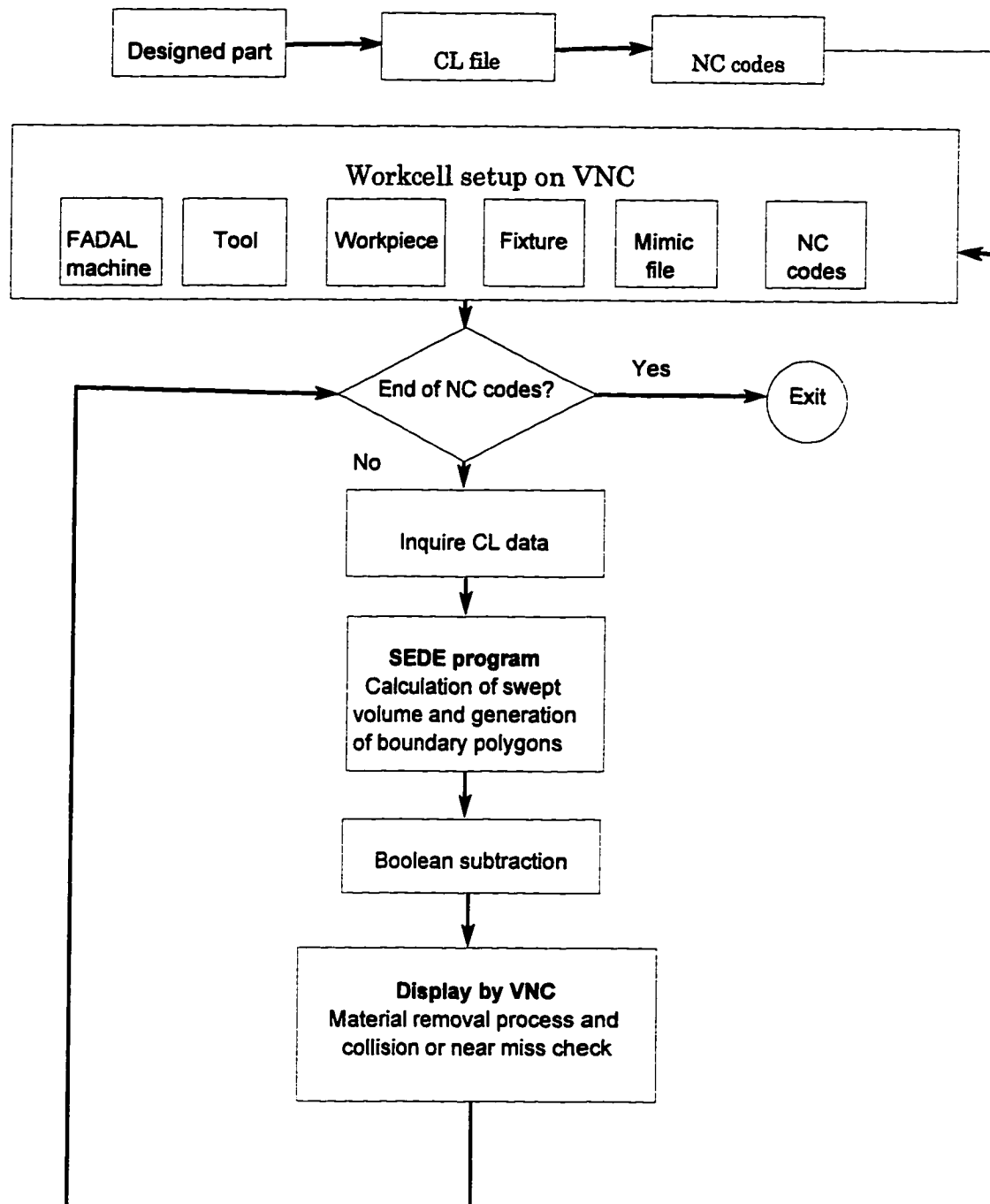
The integrated CAD/CAM system for five-axis NC milling, which features means for parametric design, several ways of NC program verification in machining environments and five-axis NC milling capacity, has enhanced the design and manufacturing capability of the Center for Manufacturing Systems at NJIT.

The SEDE method requires computation of points on the swept volume boundary only at the initial and final position of the object. Solving a sweep envelope differential equation is easy via a Runge-Kutta method the convergence is automatically achieved. To solve the envelope equations, in some cases, convergence is either very slow or cannot be efficiently achieved. So the SEDE method is qualitatively more efficient than the envelope method. Both the SEDE method and the envelope method have been implemented for numerical calculation of swept volumes. The SEDE program has shown to be more advanced in five-axis motion description, swept volume representation and application in actual verification.

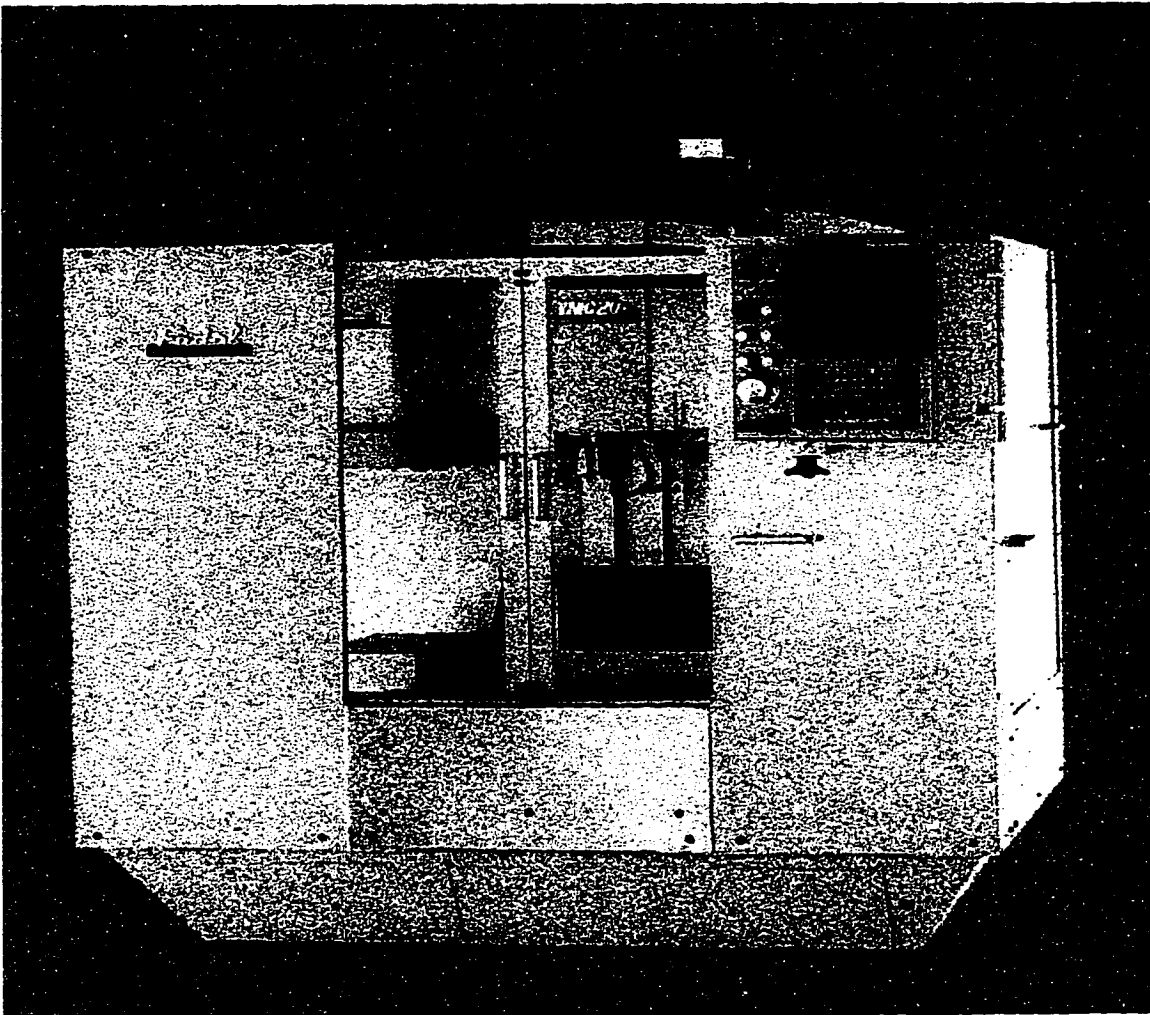




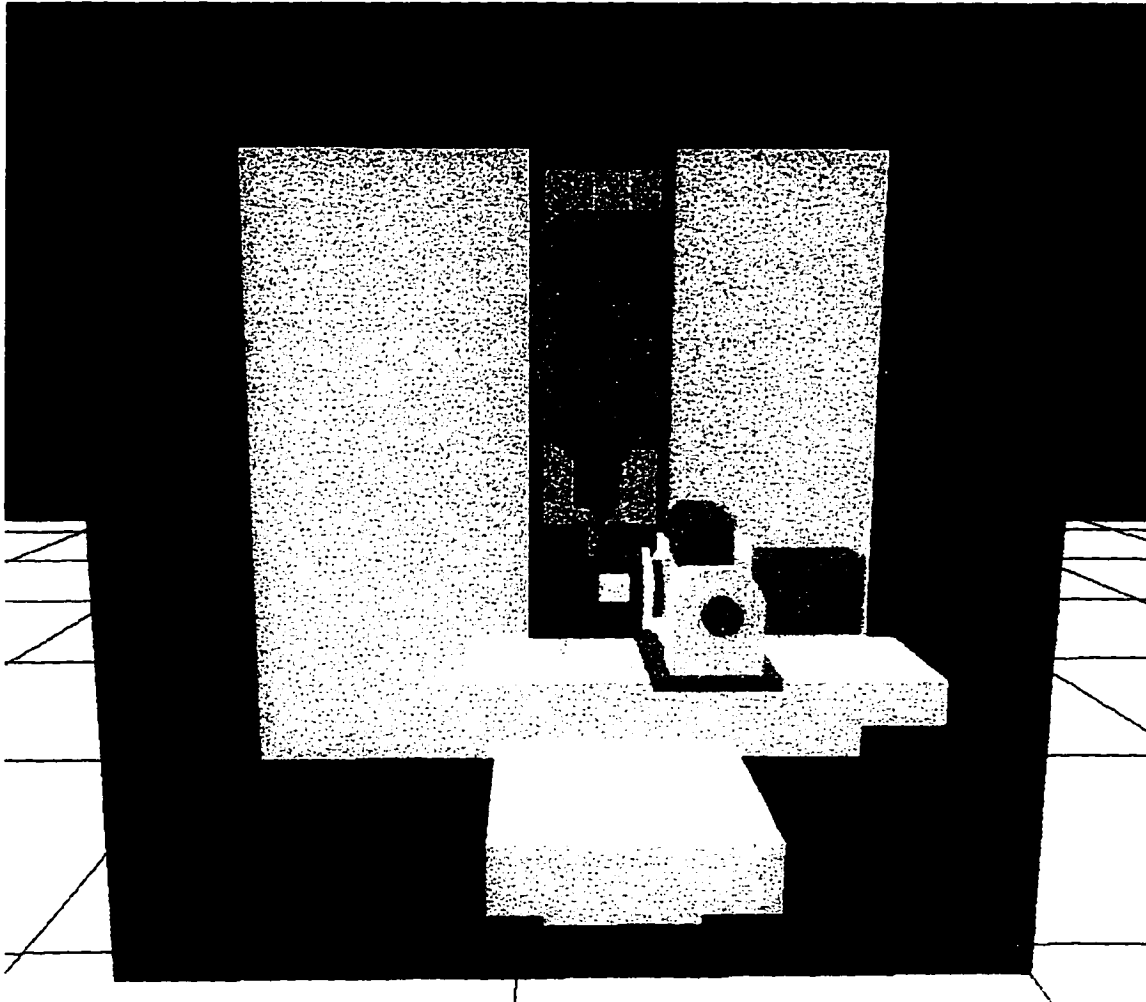
**Figure 5.1** NC simulation for calculated inclination and yaw angles



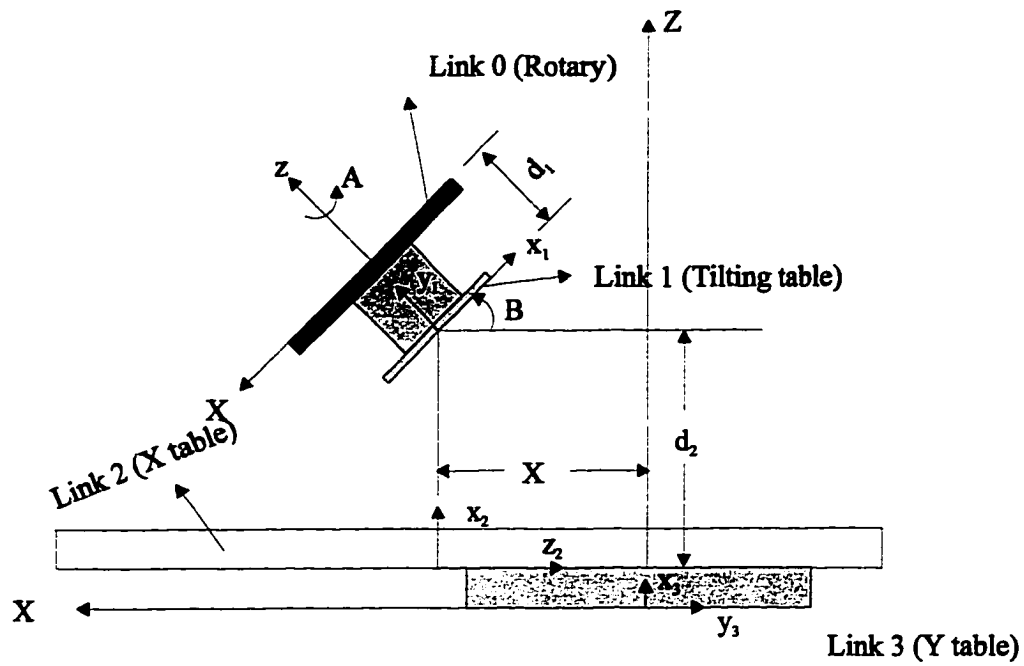
**Figure 5.2** Integration of SWEEP-SEDE sweep module with commercial software



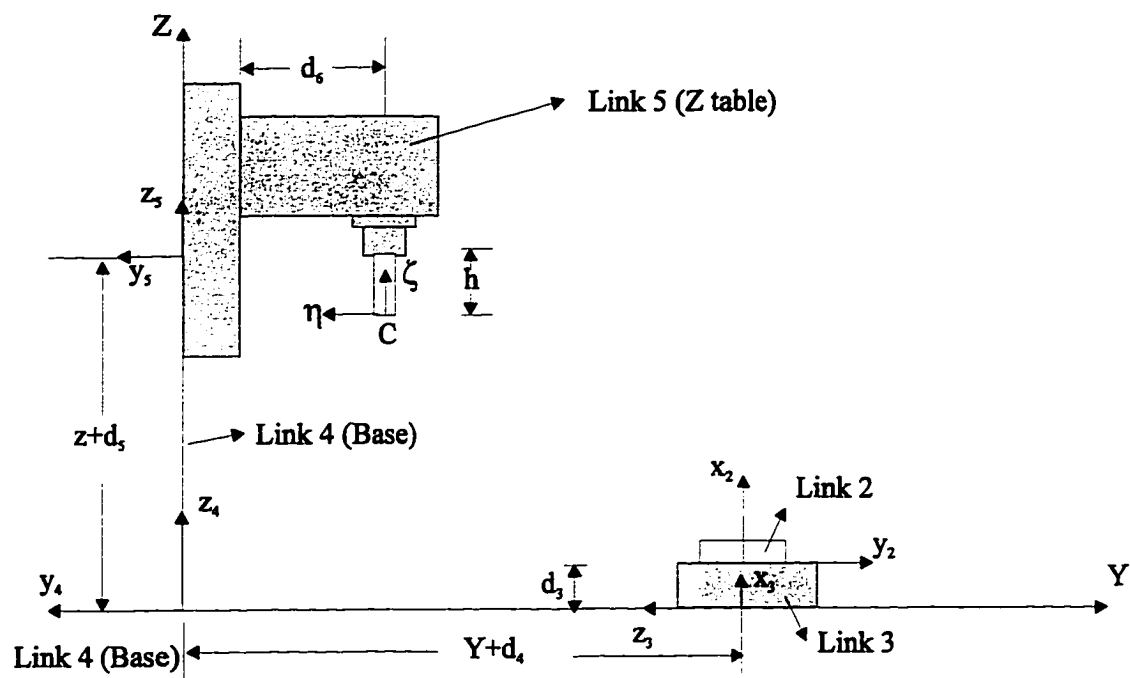
**Figure 5.3** FADAL VMC-20 five-axis milling machine



**Figure 5.4** Simulated workcell of FADAL VMC-20  
five-axis milling machine

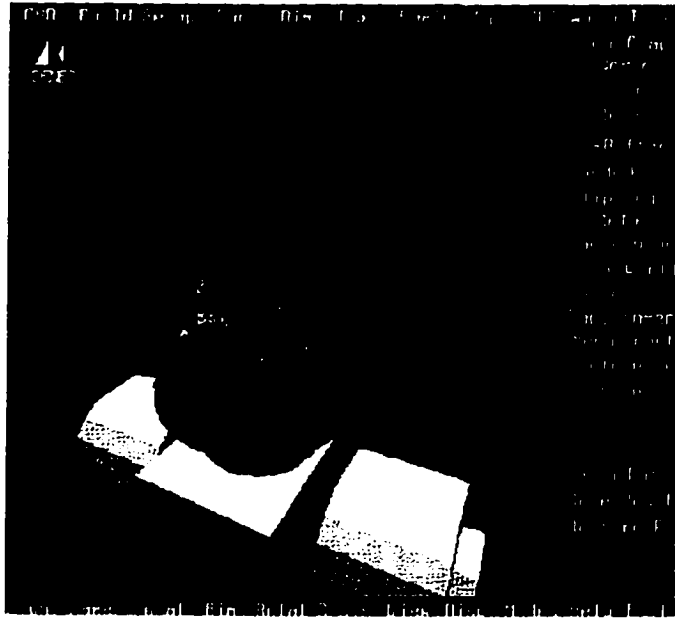


(a) Link 0 -- Link 2

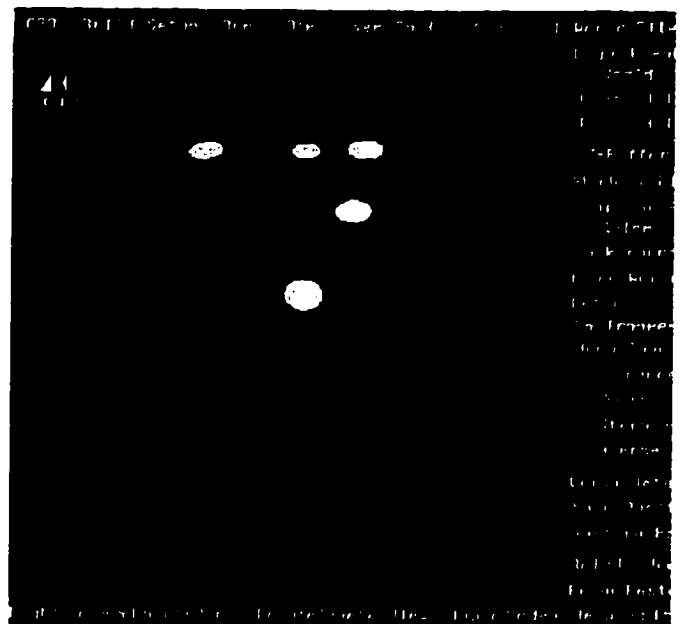


(b) Link 3 -- Link 5

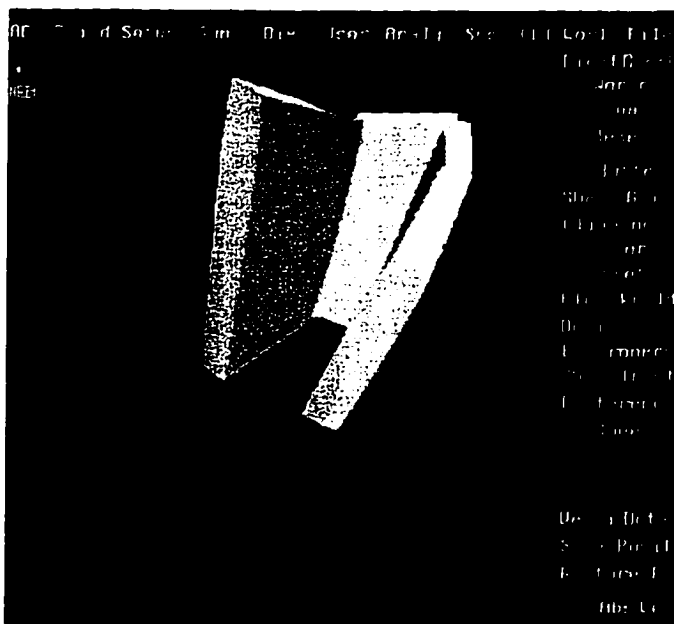
**Figure 5.5** Denavit-Hartenberg representation of FADAL VMC-20



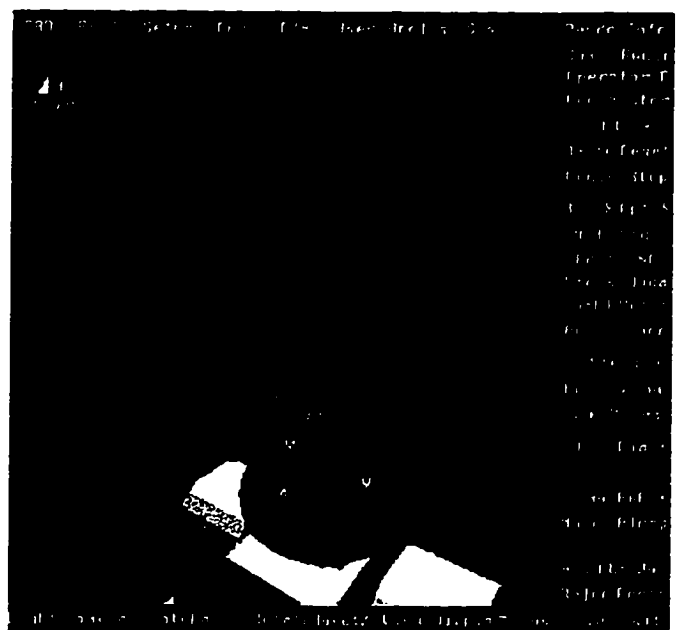
(a)



(b)



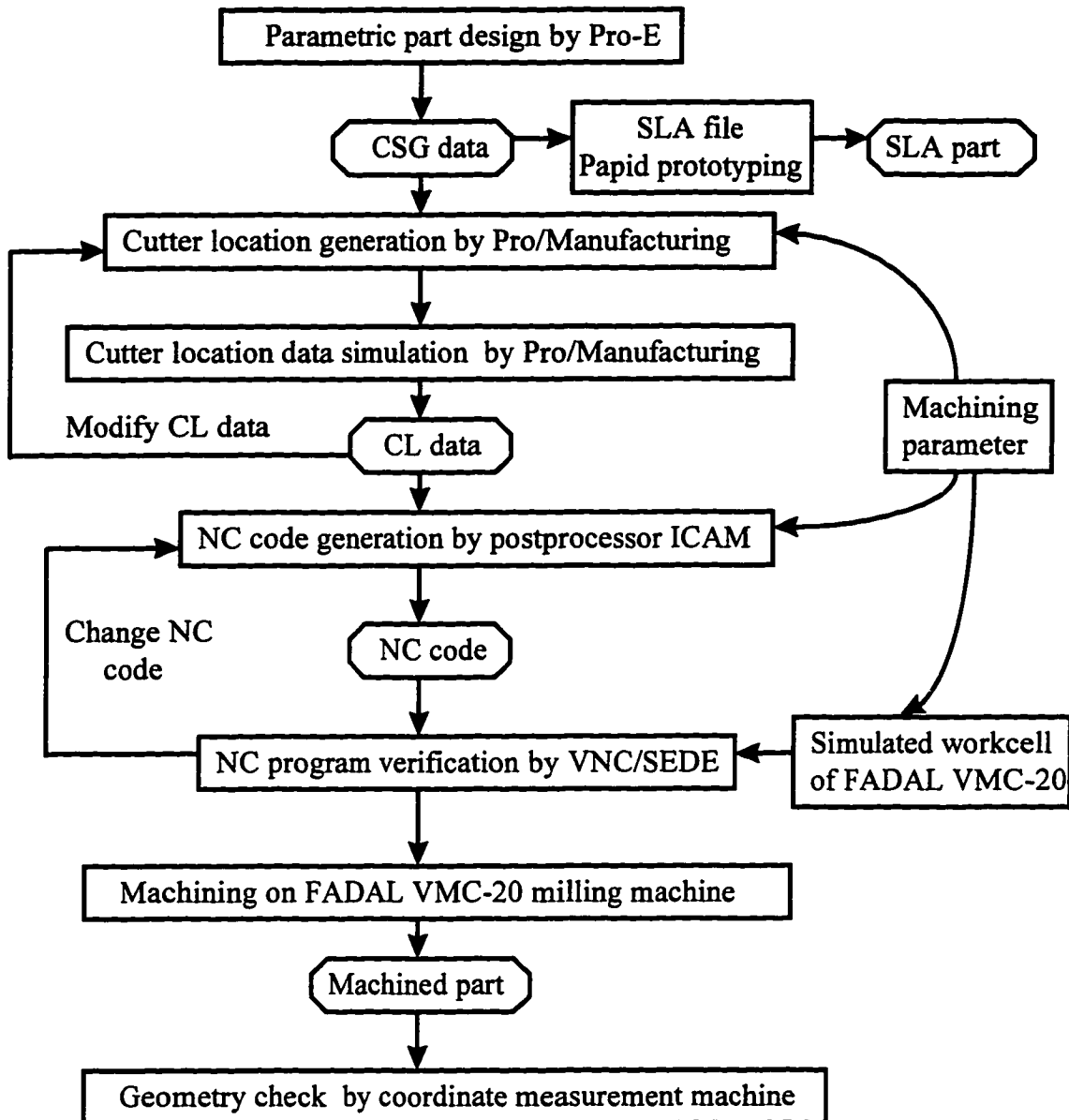
(c)



(d)

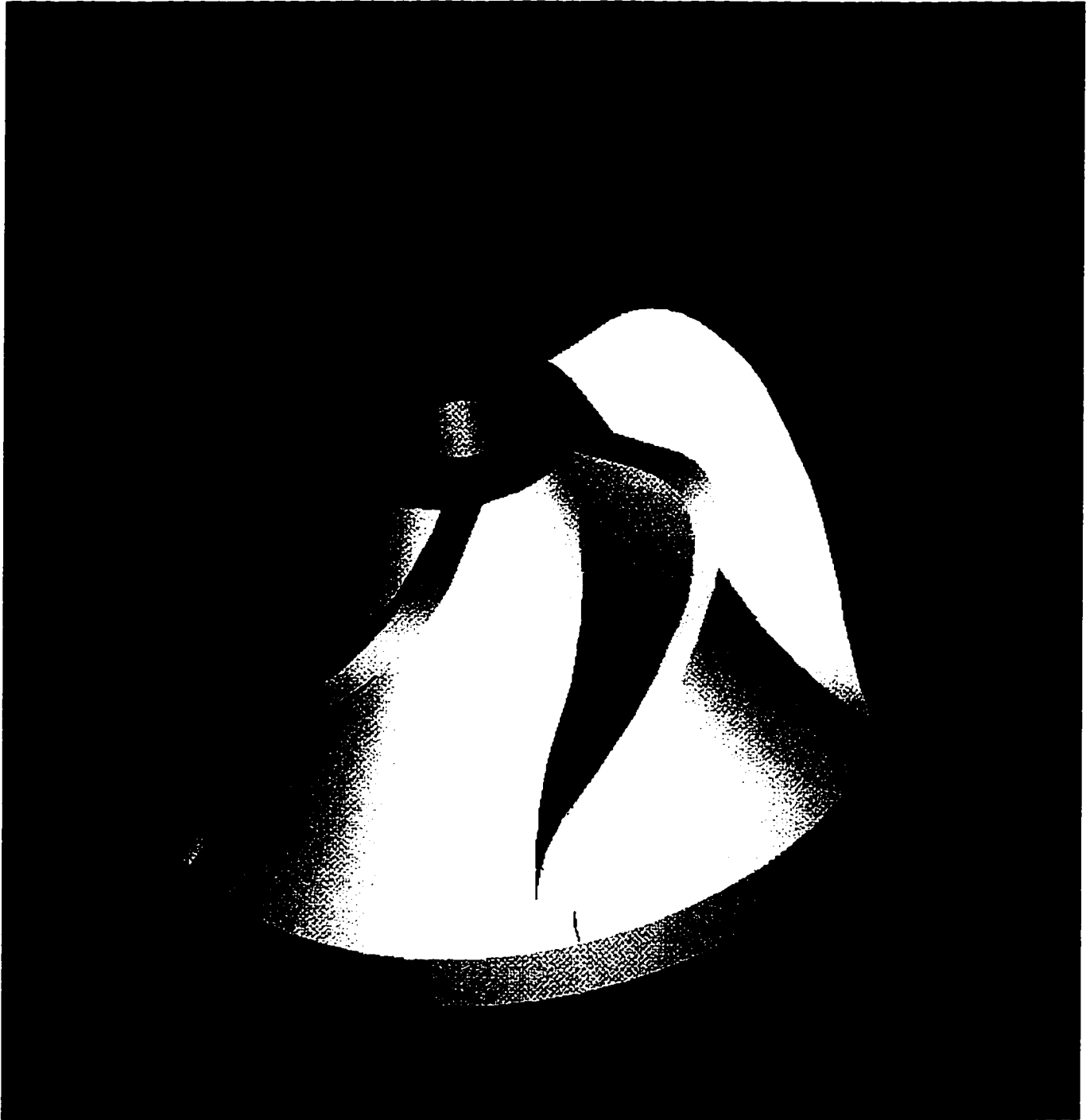
Figure 5.6 Graphical output for Example 5.1





**Figure 5.8** Integrated CAD/CAM system for five-axis NC milling process

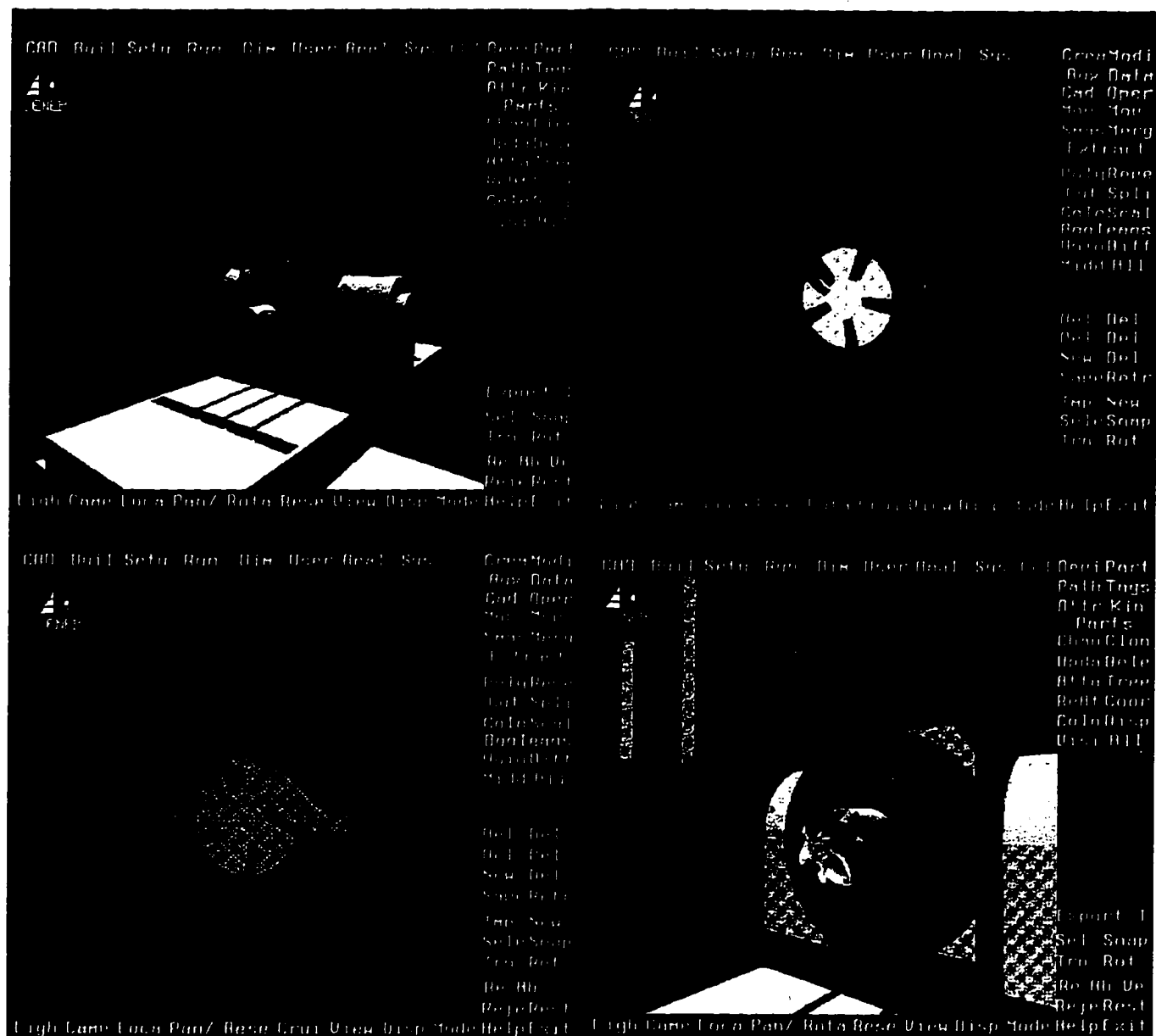




**Figure 5.9** Designed part of turbine impeller



**Figure 5.10** NC program simulation in process on Virtual NC



**Figure 5.11** Turbine machining simulation by SEDE and VNC



**Figure 5.12** Completed parts: SLA, wax, metal (from left to right)

## CHAPTER 6

### CONCLUSIONS AND FUTURE WORK

Implementation of the SDE theory has been extended to three dimensional swept volumes generated by a moving tool in five-axis NC milling. The points on the boundary of the cutter swept volume are shown to be a subset of the union of 1) the grazing points of the cutter for the entire sweep, 2) the ingress points of the cutter at the beginning of the sweep, and 3) the egress points of the cutter at the end of the sweep. A machine control data based interpolation method is developed to uniquely describe the interpolation motion equation of a five-axis NC milling tool and applied to NC simulation and verification.

A novel SDE/SEDE based algorithm has been developed to characterize the boundary points of a swept volume generated by a smooth object undergoing smooth motion. It is qualitatively more efficient since it requires computation of points on the swept volume boundary only at the initial and final position of the object. The computational time complexity of the SEDE algorithm has been evaluated as  $N^3$ , where  $N$  is the maximum number of discretization of the tool surface and time interval. The upper bound error for the SEDE swept volume method has been estimated to be  $K^2$ , where  $K$  is the maximum mesh size of the tool surface and triangulation of boundary points. Using the SEDE approach, the problem of approximating a swept volume is reduced to solving a differential equation. The solutions of differential equations have been the focus of many researchers for many years and hence such solutions allow one to robustly and accurately construct the swept volume by exploiting that knowledge.

The SDE/SEDE methods have been successfully used to calculate the boundary points of a swept volume in space that incorporates both the geometry of the object and the motion. It may be used in representing the swept volume generated by a moving five-axis NC milling tool and forming the sweep generator in

a computer based NC simulator/verifier. This has been proved by the integration of the SEDE sweep module with a commercial CAD/CAM system.

Kinematics analysis is incorporated into the integrated SDE/SEDE approach to simulate the actual motion of a five-axis milling controller and to support NC simulation/verification in machining environments.

A method for smooth functional approximation of a general 7-parameter APT tool capable of providing any desired accuracy is described. This may be useful for other applications such as vision, graphics and computer-aided geometric design where functional modeling of a tool surface is useful.

Several features of the SEDE algorithm described in this dissertation were specifically designed to facilitate integration of the computer implementation of the SEDE algorithm with the particular commercial CAD software that is used. With minor modification the basic algorithm can be effectively interfaced with any of the common CAD software packages for a variety of practical applications.

The SEDE theory developed in this dissertation work applies primarily to smooth objects. Many of the objects encountered in manufacturing applications, however, have piecewise smooth rather than smooth boundaries. Although this has been dealt with by representing such an object as a union of smooth objects, it is desirable to deal with these objects in their exact geometry instead of approximating them by smooth objects.

The deforming object theory (Blackmore et al., 1994) developed in the context of SDEs can be used to extend the SEDE based theory and algorithm to encompass swept objects experiencing deformation. Implementation of the deforming theoretical framework to actual NC machining is an important future task.

## APPENDIX A

### KINEMATICS ANALYSIS OF GENERAL 5-AXIS NC MILLING MACHINES

The sweep equation of the MCD interpolation based five-axis motion and the transformation between CL data and MCD data may be derived by the following three-step procedure.

#### Step 1: Denavit-Hartenberg representation of the machine model

Here we find the kinematics parameters of the 5-axis machine in Denavit-Hartenberg notation (Asada & Slotine, 1986). The Denavit-Hartenberg transformation matrix for adjacent coordinate frames,  $x_i, y_i, z_i$  and  $x_{i-1}, y_{i-1}, z_{i-1}$  is

$$A_i = \begin{pmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (A.1)$$

The general machine can be specified as a sequence of transformations of coordinates:

$$T_i = \begin{pmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = A_1 A_2 \dots A_5 A_i \quad (A.2)$$

#### Step 2: kinematics equation (sweep equation)

Here we find the transformation matrices and derive the sweep equation. In other words, we calculate the transformation of coordinate systems between tool coordinate frame and workpiece coordinate frame. This has the form

$$\mathbf{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} + \begin{pmatrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{pmatrix} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix} \quad (\text{A.3})$$

where the vector  $\mathbf{r} = (x, y, z)^T$  represents any point in the reference coordinate frame  $O-xyz$  and  $\mathbf{r}' = (\xi, \eta, \zeta)^T$  represents any point in the tool coordinate frame  $C-\xi\eta\zeta$ .

This equation may be obtained either by 3D transformation theory or by introducing Denavit-Hartenberg representation for the links of the machine and multiplying the transformation matrix for each coordinate frame.

### Step 3: transformation of CL data to MCD data

Transformation of CL data to MCD data is expressed as

$$X = X(x_c, y_c, z_c, I, J, K, d, r, a, b, \alpha, \beta, h) \quad (\text{A.4})$$

$$Y = Y(x_c, y_c, z_c, I, J, K, d, r, a, b, \alpha, \beta, h) \quad (\text{A.5})$$

$$Z = Z(x_c, y_c, z_c, I, J, K, d, r, a, b, \alpha, \beta, h) \quad (\text{A.6})$$

$$A = A(x_c, y_c, z_c, I, J, K, d, r, a, b, \alpha, \beta, h) \quad (\text{A.7})$$

$$B = B(x_c, y_c, z_c, I, J, K, d, r, a, b, \alpha, \beta, h) \quad (\text{A.8})$$

where

$$I = \frac{i}{\sqrt{i^2 + j^2 + k^2}} \quad J = \frac{j}{\sqrt{i^2 + j^2 + k^2}} \quad K = \frac{k}{\sqrt{i^2 + j^2 + k^2}}$$

and  $(I, J, K)$  represents the normalized directional cosines.

This equation may be obtained either by translational and rotational relationships or by transformation equations.



## APPENDIX B

### GENERAL 7-PARAMETER APT TOOLS

A general 7-parameter APT tool is illustrated in Fig.B.1. In this section, piecewise smooth functions are adopted to represent the face equations and normals (in the tool coordinate frame) for the three faces (with the top face excluded) of the tool surface of a general 7-parameter APT tool: lower conical, toroidal and upper conical (shown in Fig.B.2). They are the basis for the further calculation of the swept volumes using the SDE method and the error analysis of tool surface approximation.

#### B.1 Lower Conical Surface(Fig.B.2(a))

On the lower conical part, from  $\triangle CDD'$ ,

$$\tan\left(\frac{\pi}{2} - \alpha\right) = \frac{DD'}{CD'} = \frac{\sqrt{\xi^2 + \eta^2}}{\zeta}$$

$$\zeta = \tan \alpha \cdot \sqrt{\xi^2 + \eta^2}$$

The exact surface equation of the lower conical section is represented by:

$$f_1(\xi, \eta, \zeta) = \zeta - \tan \alpha \cdot \sqrt{\xi^2 + \eta^2} = 0 \quad (0 < \zeta \leq \zeta_D) \quad (\text{B.1})$$

$$\zeta_D = \sin \alpha (a \cos \alpha + b \sin \alpha) \quad (\text{B.2})$$

The partial derivatives needed to calculate the normal are:

$$\begin{aligned} \frac{\partial f_1}{\partial \xi} &= -\tan \alpha \cdot \frac{\xi}{\sqrt{\xi^2 + \eta^2}} \\ \frac{\partial f_1}{\partial \eta} &= -\tan \alpha \cdot \frac{\eta}{\sqrt{\xi^2 + \eta^2}} \\ \frac{\partial f_1}{\partial \zeta} &= 1 \end{aligned}$$

$$\sqrt{\left(\frac{\partial f_1}{\partial \xi}\right)^2 + \left(\frac{\partial f_1}{\partial \eta}\right)^2 + \left(\frac{\partial f_1}{\partial \zeta}\right)^2} = \sqrt{1 + \tan^2 \alpha} = \frac{1}{\cos \alpha}$$

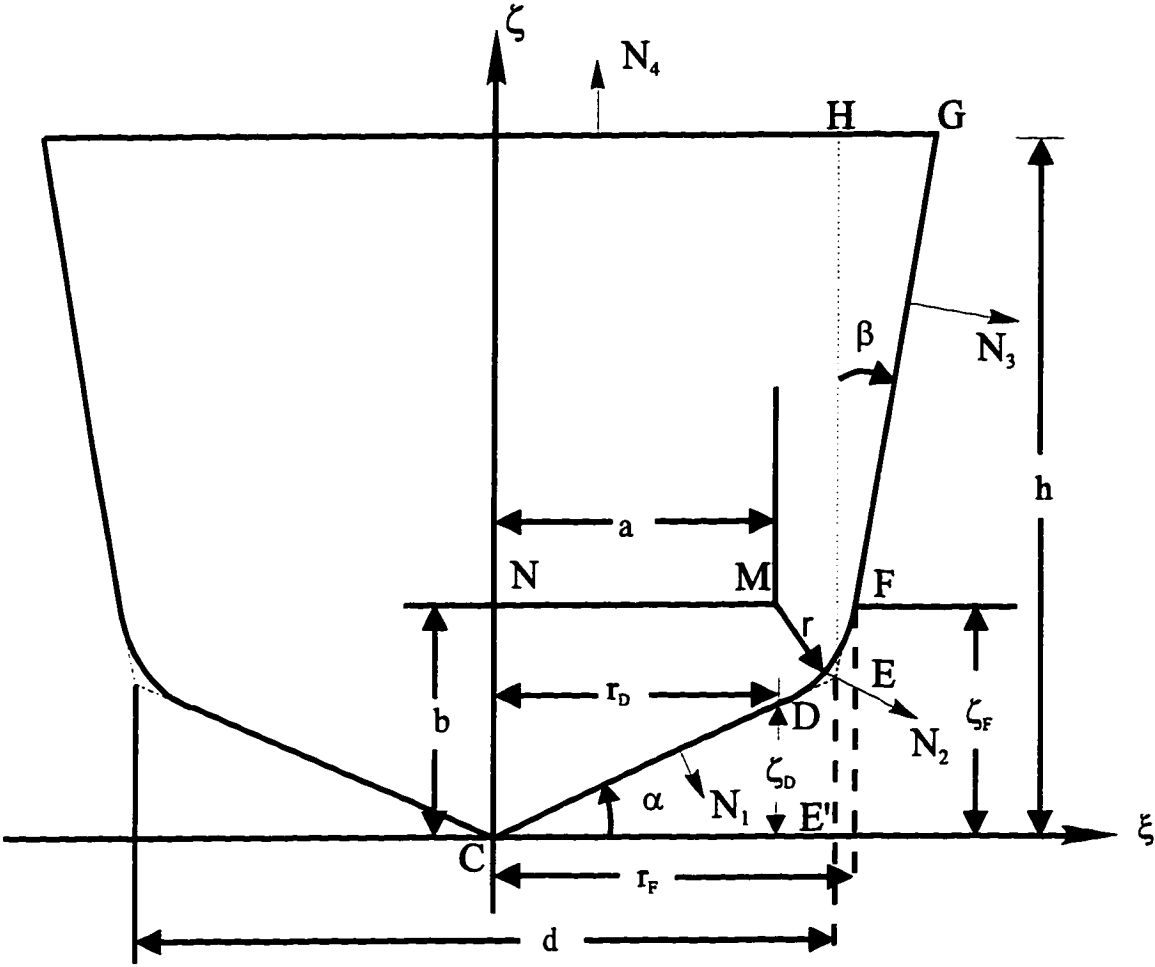
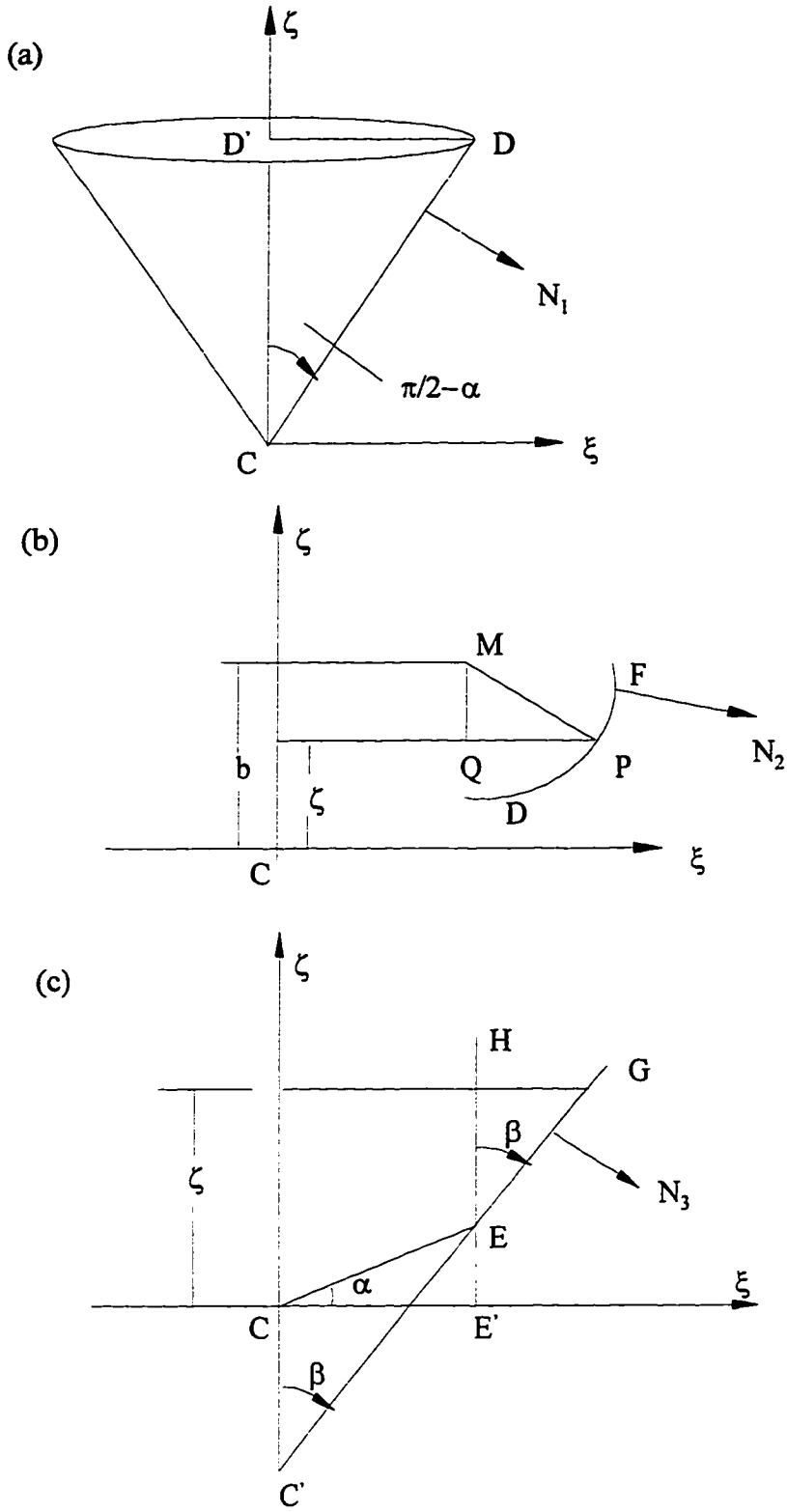


Figure B.1 General 7-parameter APT tool



**Figure B.2** Calculation of normals in tool coordinate frame

So the normal for the lower conical surface is:

$$\begin{aligned}
 N'_1 &= \frac{(\frac{\partial f_1}{\partial \xi}, \frac{\partial f_1}{\partial \eta}, \frac{\partial f_1}{\partial \zeta})}{\sqrt{(\frac{\partial f_1}{\partial \xi})^2 + (\frac{\partial f_1}{\partial \eta})^2 + (\frac{\partial f_1}{\partial \zeta})^2}} \\
 &= \cos \alpha \cdot (-\tan \alpha \cdot \frac{\xi}{\sqrt{\xi^2 + \eta^2}}, -\tan \alpha \cdot \frac{\eta}{\sqrt{\xi^2 + \eta^2}}, 1) \\
 &= (-\sin \alpha \cdot \frac{\xi}{\sqrt{\xi^2 + \eta^2}}, -\sin \alpha \cdot \frac{\eta}{\sqrt{\xi^2 + \eta^2}}, 1)
 \end{aligned} \tag{B.3}$$

## B.2 Toroidal Surface (Fig. B.2(b))

For a point  $P$  on the toroidal surface, from  $\triangle PQM$ ,

$$(\sqrt{\xi^2 + \eta^2} - a)^2 + (b - \zeta)^2 = r^2$$

The surface equation of the toroidal surface is:

$$f_2(\xi, \eta, \zeta) = (\sqrt{\xi^2 + \eta^2} - a)^2 + (b - \zeta)^2 - r^2 = 0 \quad (\zeta_D \leq \zeta \leq \zeta_F) \tag{B.4}$$

$$\zeta_F = \cos^2 \beta [2b - \tan \beta (\frac{d}{2} - \frac{d}{2} \tan \alpha \tan \beta - a)] \tag{B.5}$$

The partial derivatives needed to calculate the normal are:

$$\begin{aligned}
 \frac{\partial f_2}{\partial \xi} &= 2(\sqrt{\xi^2 + \eta^2} - a) \cdot \frac{\xi}{\sqrt{\xi^2 + \eta^2}} \\
 \frac{\partial f_2}{\partial \eta} &= 2(\sqrt{\xi^2 + \eta^2} - a) \cdot \frac{\eta}{\sqrt{\xi^2 + \eta^2}} \\
 \frac{\partial f_2}{\partial \zeta} &= 2(\zeta - b)
 \end{aligned}$$

$$\sqrt{(\frac{\partial f_2}{\partial \xi})^2 + (\frac{\partial f_2}{\partial \eta})^2 + (\frac{\partial f_2}{\partial \zeta})^2} = 2\sqrt{\xi^2 + \eta^2 + (\zeta - b)^2 - 2a\sqrt{\xi^2 + \eta^2}}$$

The normal on the toroidal surface is:

$$\begin{aligned}
 N'_2 &= \frac{(\frac{\partial f_2}{\partial \xi}, \frac{\partial f_2}{\partial \eta}, \frac{\partial f_2}{\partial \zeta})}{\sqrt{(\frac{\partial f_2}{\partial \xi})^2 + (\frac{\partial f_2}{\partial \eta})^2 + (\frac{\partial f_2}{\partial \zeta})^2}} \\
 &= \frac{(\frac{\xi(\sqrt{\xi^2 + \eta^2} - a)}{\sqrt{\xi^2 + \eta^2}}, \frac{\eta(\sqrt{\xi^2 + \eta^2} - a)}{\sqrt{\xi^2 + \eta^2}}, \zeta - b)}{\sqrt{\xi^2 + \eta^2 + (\zeta - b)^2 - 2a\sqrt{\xi^2 + \eta^2}}}
 \end{aligned} \tag{B.6}$$

### B.3 Upper Conical Surface (Fig.B.2(c))

From  $\triangle CC'E$ , find  $\overline{CC'}$ :

$$\overline{CE'} = d/2, \quad \cos \alpha = \frac{d/2}{\overline{CE}}, \quad \overline{CE} = \frac{d}{2 \cos \alpha}$$

$$\frac{\overline{CC'}}{\sin(180^\circ - 90^\circ - \alpha - \beta)} = \frac{\overline{CE}}{\sin \beta}$$

$$\frac{\overline{CC'}}{\cos(\alpha + \beta)} = \frac{d}{2 \cos \alpha \sin \beta}$$

$$\overline{CC'} = \frac{d \cos(\alpha + \beta)}{2 \cos \alpha \sin \beta}$$

$$\tan \beta = \frac{\sqrt{\xi^2 + \eta^2}}{\zeta + \overline{CC'}} = \frac{\sqrt{\xi^2 + \eta^2}}{\zeta + \frac{d \cos(\alpha + \beta)}{2 \cos \alpha \sin \beta}}$$

$$\tan \beta \cdot \zeta + \frac{d \cos(\alpha + \beta)}{2 \cos \alpha \sin \beta} = \sqrt{\xi^2 + \eta^2}$$

The surface equation for the upper conical surface is:

$$f_3(\xi, \eta, \zeta) = \tan \beta \cdot \zeta + \frac{d \cos(\alpha + \beta)}{2 \cos \alpha \sin \beta} - \sqrt{\xi^2 + \eta^2} = 0 \quad (\zeta_F \leq \zeta \leq L) \quad (B.7)$$

The partial derivatives needed to calculate the normal are:

$$\frac{\partial f_3}{\partial \xi} = \frac{\xi}{\sqrt{\xi^2 + \eta^2}}$$

$$\frac{\partial f_3}{\partial \eta} = \frac{\eta}{\sqrt{\xi^2 + \eta^2}}$$

$$\frac{\partial f_3}{\partial \zeta} = -\tan \beta$$

$$\sqrt{\left(\frac{\partial f_3}{\partial \xi}\right)^2 + \left(\frac{\partial f_3}{\partial \eta}\right)^2 + \left(\frac{\partial f_3}{\partial \zeta}\right)^2} = \sqrt{1 + \tan^2 \beta} = \frac{1}{\cos \beta}$$

The normal for the upper conical surface is:

$$\begin{aligned} N'_3 &= \frac{(\frac{\partial f_3}{\partial \xi}, \frac{\partial f_3}{\partial \eta}, \frac{\partial f_3}{\partial \zeta})}{\sqrt{\left(\frac{\partial f_3}{\partial \xi}\right)^2 + \left(\frac{\partial f_3}{\partial \eta}\right)^2 + \left(\frac{\partial f_3}{\partial \zeta}\right)^2}} \\ &= \left( \cos \beta \cdot \frac{\xi}{\sqrt{\xi^2 + \eta^2}}, \cos \beta \cdot \frac{\eta}{\sqrt{\xi^2 + \eta^2}}, -\sin \beta \right) \end{aligned} \quad (B.8)$$

## REFERENCES

1. Aho, Hopcroft, and Ullman, *Data Structures and Algorithms*, Reading, Massachusetts: Addison-Wesley, 1987.
2. Anderson, R.O., "Detecting and eliminating collisions in NC machining", *Comput. Aided Des.*, vol. 10, no. 4, pp. 231-237, July, 1978.
3. Asada, H., and Slotine, J.J.E., *Robot analysis and control*, New York: John Wiley & Sons, pp. 31-40, 1986.
4. Blackmore, D., Leu, M.C., and Wang, K. K., "Applications of flows and envelopes to NC machining", *Annals of the CIRP*, 41/1, pp. 493-496, 1992a.
5. Blackmore, D., and Leu, M.C., "Analysis of swept volume via Lie groups and differential equations", *Int. J. Robotics Res.*, vol. 11, no. 6, pp. 516-537, 1992b.
6. Blackmore, D., Leu, M.C., and Wang, W., "Classification and analysis of robot swept volumes", *Proc. of Japan/USA Symposium on Flexible Automation*, Volume 1, pp. 69-75, San Francisco, California, 1992c.
7. Blackmore, D., and Leu, M.C. "A differential equations approach to swept volume", *Proc. of Rennselaer's Second International Conference on Computer Integrated Manufacturing*, pp. 143-148, Troy, New York, 1990.
8. Blackmore, D., Leu, M.C., and Shih, F., "Analysis and modeling of deformed swept volumes", *Comput. Aided Des.*, vol. 26, pp. 315-326, 1994.
9. Blackmore, D., Leu, M.C., Qin, D., and Wang, L., "An algorithm for computing swept volumes", *Proc. of Fourth SIAM Conference on Geometric Design*, Nashville, Tennessee, Nov. 6-9, 1995.
10. Blackmore, D., Leu, M.C., and Wang, L.P., "The sweep-envelope differential equation algorithm and its application to NC machining verification", *Comput. Aided Des.*, 1996a. (accepted).
11. Blackmore, D., Leu, M.C., Wang, L., and Jiang H., "Swept volume: a retrospective and prospective view", *Journal of Neural, Parallel & Scientific Computation*, 1996b. (accepted).
12. Chappel, I.T., "The use of vectors to simulate material removal by numerically controlled milling", *Comput. Aided Des.*, vol. 15, no. 3, pp. 156-158, 1983.
13. Deneb Robotics Inc., *VNC User Manual*, Version 2.4, 1995.
14. Deng, Z., Leu, M.C., and Blackmore, D., "Application of sweep differential equation approach to nonholonomic motion planning", *Proc. of Japan-U.S.A. Symposium on Flexible Automation*, Kobe, Japan, 1994a.

15. Deng, Z., Leu, M.C., and Blackmore, D., "Application of sweep differential equation approach to 5-axis sculptured surface machining", *Proc. of the Third International Conference on Automation Technology*, Taipei, Taiwan, 1994b.
16. Deng, Z., Leu, M.C., Wang, L. and Blackmore, D., "Determination of flat-end cutter orientation in 5-axis machining", *Proc. of Manufacturing Science and Engineering-1996*, MED-Vol. 4, pp. 73-80, 1996 ASME International Mechanical Engineering Congress and Exposition, Atlanta, Georgia, Nov.17-22, 1996.
17. Drysdale R.L., Jerard, R.B., Schaudt, B., and Hauck, K., "Discrete simulation of NC machining", *Algorithmica, special issue on computational geometry*, vol. 4, no. 1, pp. 33-60, 1989.
18. Elden, L., and Wittmeyer-Koch, L., *Numerical Analysis*, San Diego, California: Academic Press, Inc., pp. 112, 1990.
19. Farlin, G., *Curves and Surfaces for Computer Aided Geometric Design*, Second Edition, San Diego, California: Academic Press, Inc., 1990.
20. Fridshal, R., et al., "Numerical control part program verification system", *Proc. MIT conference on CAD/CAM Technology for mechanical engineering*, Cambridge, Massachusetts: MIT Press, March, 1982.
21. Ithaca Software, *Hoops Graphics System: User Guides and Reference Manual*, Version 2.2, 1990.
22. Isaacson, E., and Keller, H.B., *Analysis of Numerical Methods*, New York: John Wiley & Sons, 1966.
23. Jensen, C. G., and Anderson, D. C., "Accurate tool placement and orientation for finish surface machining", *Concurrent Engineering*, ASME PED-Vol. 59, pp. 127-145, 1992.
24. Jensen, C. G., Mulkay, E. L., and Simpson, T. S., "A practical implemetation of curvature matched machining for complex geometric objects", *Concurrent Product and Process Engineering*, ASME MED-Vol.1/DE-Vol.85, pp. 1-14, 1995.
25. Jerard, R. B., Angleton, J. M., and Drysdale, R. L., "Sculptured surface tool path generation with global interference checking", *Proc. of Design Productivity International Conference*, Honolulu, Hawaii, 1991.
26. Jerard, R.B., Hussaini, S.Z., Drysdale, R.L., and Schaudt, B., "Approximate method for simulation and verification of numerically controlled machining programs", *Visual Computer*, vol. 5, no. 6, pp. 329-348, 1989.

27. Jerard, R.B., Drysdale, R. L., and Magewick, J., "Methods for detecting errors in numerically controlled machining of sculptured surfaces", *IEEE computer graphics & Applications*, vol. 1, pp. 26-39, January, 1989.
28. Jerard, R.B., Angleton, J.M., Drysdale, R.L., and Su, P., "The use of surface point sets for generation, simulation verification and automatic correction of NC machining programs", *Proc. of NSF Design and Manufacturing Systems Conference*, Tempe, Arizona, pp. 143-148, 1990.
29. Jerard, R. B., and Drysdale, R. L., "Methods for geometric modeling, simulation and spatial verification of NC machining", *Product modeling for computer aided design*. Amsterdam: North-Holland. pp. 1-14, 1991.
30. Jiang, H., "The flow approach to swept volume", M.Sc. Thesis, Dept. of Math., New Jersey Institute of Technology, Newark, New Jersey, 1993.
31. Jiang, H., Blackmore, D., and Leu, M.C., "The flow approach to CAD/CAM modeling of swept volumes", *Manufacturing Systems: Design, Modeling and Analysis*, Amsterdam: Elsevier, pp. 341-346, 1994.
32. Keren, D.K., Cooper, D., and Subrahmonia, J., "Describing complicated objects by implicit polynomials", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 38-53, 1994.
33. Kieffer, J., and Litovin, T., "Swept volume determination and interference detection for moving 3-D solids", *ASME J. Mech. Des.*, vol. 113, pp. 456-453, 1991.
34. Korakianitis, T and Pantaropoulos, G.I., "Improved turbine blade design techniques using 4th-order parametric-spline segments", *Comput. Aided Des.*, vol. 25, no. 12, pp. 289-299, 1993.
35. Koren, Y. and Lin, R.S., "Five-axis surface interpolators", *Annals of the CIRP*, vol. 44/1, pp. 379-382, 1995.
36. Kral, Irvin H., *Numerical Control Programming in APT*, Englewood Cliffs, New Jersey: Prentice Hall, 1986.
37. Leavitt, J., and Messcher, W., "Automatic generation of 3-D envelopes", *Proc. NASA Symposium on Computer-aided Geometry Modeling*, pp.145-149, Hampton, Virginia, April 20-22, 1984.
38. Leu, M.C., Park, S. H., and Wang, K. K., "Geometric representation of translational swept volume and its applications", *ASME J. of Eng. for Indust*, vol. 108, pp. 113-119, 1986.



39. Leu, M.C., Blackmore, D., Wang, L. and Pak, K., "Implementation of SDE method to represent cutter swept volumes in 5-axis NC milling", *Proc. Int. Conf. on Intelligent Manufacturing*, pp. 211-220, Wuhan, China, June 14-18, 1995.
40. Leu, M.C., Wang, L., and Blackmore, D., "A NC verification system for 5-axis NC machining with general APT tools", *Annals of the CIRP*, vol. 46/1, 1997. (accepted).
41. Li, S.X., and Jerard, R.B., "5-Axis machining of sculptured surfaces with a flat-end cutter", *Comput. Aided Des.*, vol. 26, no. 3, pp. 165-178, 1994.
42. Lin, R.S. and Koren, Y., "Real-time five-axis interpolator for machining ruled surfaces", *Dynamics Systems and Control*, Volume 2, ASME, 1994.
43. Ling, Z.K., and Chase, T., "Generating the area of a body undergoing planar motion", *J. Mech. Des.*, 1995. (to appear).
44. Liu, X.W., "Five-axis NC cylindrical milling of sculptured surfaces", *Comput. Aided Des.*, vol. 27, no. 12, 1995.
45. Martin, R.R., and Stephenson, P. C., "Sweeping of three-dimensional objects", *Comput. Aided Des.*, vol. 22, no. 4, pp. 223-234, 1990.
46. Menon, J.P., "Automatic NC verification: extended formulation and new results", *Proc of ASME/IEEE Manufacturing International '90 Conference*, Atlanta, Georgia, March 25-28, 1990.
47. Menon, J.P. and Robinson, D.M., "Advanced NC verification via massively parallel raycasting", *Manufacturing Review*, vol. 6, no. 2, pp.141-154, 1993.
48. Menon, J.P. and Voelcker, H.B., "Toward a comprehensive formulation of NC verification as a mathematical and computational problem", *J. of Des. and Manufac.*, vol. 3, pp. 263-277, 1993.
49. Menon, J., Marisa, R., and Zagajac, J., "More powerful solid modeling through ray representation", *IEEE Comput. Graph. & Applic.*, v. 14, pp. 22-35, 1994.
50. Narvekar, A.P., "Representation and application of swept solids for numerically controlled milling", M.S. Thesis, Dept. of Mech. and Aero. Engr., State Univ. of New York at Buffalo, Buffalo, New York, 1991.
51. Oliver, J.H. and Goodman, E.D., "Direct dimensional NC verification", *Comput. Aided Des.*, vol. 22, no. 1, pp. 3-10, 1990.
52. Parida, L., and Mudur, S., "Computational methods for evaluating swept object boundaries", *Visual Computer*, vol.10, pp. 266-276, 1994.

53. Pressman, R.S., and Williams, J.E., *Numerical control and computer-aided manufacturing*, New York: John Wiley & Sons, Inc., 1977.
54. Qin, D., Blackmore, D., and Leu, M.C., "Improved flow approach for swept volumes", *Proc. of Japan-U.S.A. Symposium on Flexible Automation*, 1994.
55. Ruegg, A., "A generalized kinematics model for three-to five-axis milling machines and their implementation in a CNC", *Annals of the CIRP*, vol. 41/1, pp. 547-550, 1992.
56. Sambandan, K., "Graphic simulation and verification of five-axis NC machining", M.S. Thesis, School of Mech. and Aero. Engr., Cornell Univ., Ithaca, New York, 1988.
57. Sambandan, K., "Geometry generated by sweeps of polygons and polyhedra", Ph.D. Dissertation, School of Mech. and Aero. Engr., Cornell Univ, Ithaca, New York, 1990.
58. Takata, S., Tsai, M.D. and Inui, M., "A cutting simulation system for machinability evaluation using a workpiece model", *Annals of the CIRP*, vol. 38/1, pp. 417-420, 1989.
59. Takeuchi, Y., and Watanake, T., "Generation of 5-axis control collision-free tool path and postprocessing for NC data", *Annals of the CIRP*, vol. 41/1: pp. 539-542, 1992.
60. Taubin, G., Cukierman, F., Sullivan, S., Ponce J., and Kriegman, D.J., "Parameterized families of polynomials for bounded algebraic curve and surface fitting", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 3, pp. 287-303, 1994.
61. Voelcker, H.B., and Menon, J.P., "Contemporary CNC machining technology", *Proc. International symposium on Steel Product-Process Integration ed. J.D. Boyd*, pp. 12-33, Annual Conf. of Metallurgists of CIM Halifax, Nova Scotia, Canada, August 20-24, 1989.
62. Wang, L., Leu, M.C., and Blackmore, D., "Kinematics analysis of five-Axis milling machine and its application to NC verification", *Proc. of International Conference on Manufacturing Automation*, Hong Kong, April 28-30, 1997a.
63. Wang, L., Leu, M.C., and Blackmore, D., "Swept volume approach as an integral part of 5-axis NC milling CAD/CAM system", *Proc. of International Conference on Manufacturing Automation*, Hong Kong, April 28-30, 1997b.
64. Wang, L., Leu, M.C., and Blackmore, D., "Generating swept solids for NC verification using the SEDE method", pp. 364-375, *Proc. of Fourth ACM Symposium On Solid Modeling and Applications*, Atlanta, Gorgia, May 14-16, 1997c.

65. Wang, W. P., and Wang, K. K., "Real-time verification of multi-axis NC programs with raster graphics", *Proc. of 1986 IEEE International Conference on Robotics and Automation*, pp. 161-171, April, 1986a.
66. Wang, W. P., and Wang, K. K., "Geometric modeling for swept volumes of moving solids", *IEEE Computer Graphics and Applications*, vol. 6, pp. 8-17, 1986b.
67. Wang, W.P, and Wang, K.K., "Solid modeling for optimizing metal removal of three-dimensional NC end milling", *J. of Manufac. Sys.*, SME, vol. 7, no. 1, pp. 57-65, 1988.
68. Weld, J. D., and Leu, M. C., "Geometric representation of swept volumes with applications to polyhedral objects", *Int. J. Robotics Res.*, vol. 9, pp.105-117, 1990.
69. Wilson, D.G., *The Design of High-Efficiency Turbomachinery and Gas Turbines*, Cambridge, Massachusetts: MIT Press, 1984.
70. Yang, M., and Lee, E., "NC verification for wire-EDM using an R-map", *Comp. Aided Des.*, vol. 28, no. 9, pp. 733-740, 1996.