MULTI-AXIS MILLING OF FLEXIBLE PARTS

BY

FARID ABRARI, B.SC., M.ENG.

NOVEMBER 1998

A THESIS

SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE

DOCTORATE OF PHILOSOPHY

MCMASTER UNIVERSITY

MULTI-AXIS MILLING OF FLEXIBLE PARTS

Doctor of Philosophy (1998) McMaster University

(Mechanical Engineering Department) Hamilton, Ontario

TITLE: Multi-axis milling of flexible parts

AUTHOR: Farid Abrari, B.Sc., M.Eng.
(McMaster University, Hamilton, Ont.)

SUPERVISOR: Dr. M.A. Elbestawi, Chair and Professor, Mechanical Engineering department
Dr. A.D. Spence, Assistant Professor, Mechanical Engineering department

NUMBER OF PAGES: xvi,156

Abstract

Multi-axis milling simulation of flexible parts is a highly interdisciplinary topic. It includes theories and methods in milling mechanics, structural mechanics and geometric simulation. In order to determine the acceptable cutting conditions, process planners faced with the flexible tools and thin walled structures, either use very conservative cutting conditions, or in the case of high volume productions, trial and error methods. In either case, a final *air* cut is usually required to bring the machined surface within the tolerance. Although practical, the above approach results in low productivity. Development of an analysis software capable of predicting the cutting forces, deflections and surface errors in multi-axis milling of the thin wall structures is therefore highly desirable. By eliminating the need for trial and error methods, a dynamic multi-axis process simulator can assist process planners to optimize the cutting condition and tool path for the least geometrical errors. This thesis describes such a simulator, implemented as an integrated CAD/CAM architecture consisting of both geometric and physical simulators.

A static cutting force model is introduced which accounts for the rotation of the a and b axes in multi-axis side milling operations. The proposed formulation is based on a vectorial approach which reduces to a conventional formulation if a three-axis tool path is used. A dynamic multi-axis force model is also developed which is capable of modeling the tool/workpiece structural interaction. The dynamic response of the workpiece to the instantaneous cutting forces is modeled using a special finite element

code. In conjunction with the finite element code, a solid based automatic mesh generation algorithm is also developed. The automatic mesh generator frequently updates the finite element mesh as the workpiece geometry changes. The dynamic parameters of the tool, experimentally found at its tip, are extrapolated for the rest of the cutter elements along the tool axis. The tool elemental parameters are then used to model the dynamic response of the cutter.

Using the effective deflection of the tool/workpiece system, profile of the machined surface is simulated and its geometric deviation from the design surface is computed. This is highly desirable for side milling of ruled surfaces, where the value added is very high.

For the experimental verification of the developed models, a twisted ruled surface was machined using a four-axis milling operation. The thickness of the twisted blade was reduced to 2.0 mm for the case of the dynamic cutting tests. The profile left on the machined surface was compared with the simulation result. In all comparisons a good agreement is seen between the experimental data and simulation results, which verifies the validity of the developed models and techniques.

Acknowledgments

The author wishes to express a sincere gratitude to his research supervisor, Dr. M.A. Elbestawi, and co-supervisor Dr. A.D. Spence for their continuing guidance and support throughout this research.

And also to the members of the supervisory committee: Dr. A. Ghobara and Dr. M. Sklad for their continuing interest. The author also wishes to thank his colleagues at IMMRC for their help at various occasions.

Dedicated to my parents and brother: Abdoallah, Afsar and Majid,
my grandmother Mahboobeh and aunt Saeedeh,
and my fiance Behnak.

Contents

List of Tables

List of Figures

Nomenclature

K_t	Tangential specific cutting constant
K_r	Tangential to radial force ratio
a_a	Axial depth of cut
a_r	Radial depth of cut
τ	Shear stress on the primary shear plane
ϕ	Shear angle
β	Friction angle on the rake face
α_e	Effective rake angle
α	Tool rake angle
R	Cutter radius
ψ	Tooth rotational position
t	Time
T	Tooth period
N_z	Number of teeth
$\underline{\underline{M}}$	Mass matrix
$\underline{\underline{K}}$	Stiffness matrix
$\underline{\underline{C}}$	Damping matrix

$\underline{\underline{\Phi}}$	Modal matrix
$\underline{u}$	Solution vector (x and y)
δ_{ij}	Kronecker delta function
$\underline{\Lambda}$	Eigen vector
u, v	B-spline parameter space variables
$\mathbf{C}(u)$	NURBS curve
$\mathbf{S}(u, v)$	NURBS surface
$N_{i,p}(u)$	B-spline basis function of degree p, where $u_i \leq u \leq u_{i+1}$
i	Inclination angle
F_P	Power component of the cutting force
F_T	Thrust component of the cutting force
F_F	Feed component of the cutting force
r_c	Chip thickness
η_c	Chip flow angle
V_c	Chip velocity vector
V	Cutting velocity
$V - V_c$	Plane containing velocity vectors of material and chip
V_e	Plane containing V and infinitesimal edge
$V - n$	Plane containing the tool axis and normal to V
γ	Angle between $V - V_c$ and V_e planes
ξ	Angle between V_e and V_n planes
$\vec{u}_q$	Tool path velocity at tool element q
$\vec{TA}$	Tool axis direction
$\vec{\Gamma}_q$	local feed direction at tool element q

Chapter 1

Introduction

The milling process is used extensively in a variety of manufacturing areas. Availability of the end mill cutters in many styles and diameters makes them adaptable to a wide variety of cutting operations. The flat, ball and tapered helical ball end mills all have common metal cutting mechanics and kinematics, but with distinct edge geometries required for some certain applications. For example ball end mills are used for machining free form surfaces in automotive, die/mold and aerospace industries. Flat end mills are used in roughing of die cavities, roughing and finishing of stamping dies and pocketing. Aircraft wing structures, fuselage sections, jet engine compressor and turbine blades and precision instrumentation housings all have flexible webs which must be milled using multi-axis machining centers. In roughing operation of these components, tool vibration is mainly responsible for the chip thickness regeneration. However, in semi-finishing and finishing operations, flexibility of the workpiece plays the most dominant role in the process regeneration. The cutter is most flexible at its free end which is in contact with or adjacent to the root of the part, i.e. the most rigid portion of the workpiece. On the contrary, the workpiece flexibility increases towards its free edge, which is close to the clamped part of the tool. In general, cutting forces are dependent on local chip thickness removed from the workpiece, which

continuously varies due to the vibrations of both tool and workpiece structures. The problem is complicated by the partial disengagement of workpiece and tool due to excessive vibrations in the system.

For a complex shaped flexible workpiece, industrial process planners typically resort to a combination of the following techniques to find the practical cutting conditions

- Using trial and error methods
- Using conservative cutting conditions
- Final air (float) cut to bring the part surface within tolerance

Clearly, the above approach results in a lower productivity. In addition, for any new part or any slight change in the original part design, the procedure has to be done all over again. Therefore, for process planning of flexible parts with complex geometric forms it is essential to be able to predict the cutting forces, deflections and surface errors. Such a process simulator can assist process planners to select the best cutting condition and tool path, for the least geometrical errors in a relatively short time and cost. In this thesis, multi-axis side milling of the conventional flat end mills is studied. The proposed methods and techniques are extensible for the peripheral milling of tapered end mills as well.

Recent advancements in the field of computer aided geometry, enables one to accurately model/simulate the geometrical aspects of multi-axis milling operations on complex parts. In this thesis, the functionality of a commercially available geometric kernel, the ACIS solid modeler, is used to model all the geometric aspects of the milling simulation. Geometric simulation, discussed in chapter 3, includes the cutting edge representation, cutter/workpiece engagement computation, multi-axis tool swept volume generation, workpiece geometric representation, tool path planning and

automatic finite element mesh generation. On the other hand, in order to use the available geometric modeling capabilities in the multi-axis milling simulation of the flexible parts, three major tasks were identified:

1. A milling force model is needed, capable of accounting for the rotation of axes a and b. This cutting force model should be able to interact with the structural deflections of the tool and workpiece system.

2. A special finite element code is needed capable of computing the workpiece dynamic response to the instantaneous cutting forces.

3. In conjunction with the finite element code, a solid based automatic mesh generation algorithm is needed to generate the required finite elements as the in-process workpiece geometry is changing.

To address the first task, a new vectorial formulation is introduced in chapter 4, capable of simulating the milling forces of a multi-axis tool path. The proposed force model is an extension to the method of Equivalent Cutting Conditions proposed by Yang and Park [YP91] for the ball end milling process. For the case of a simple tool path, however, the proposed formulation reduces to the conventional formulation of the milling simulation. Due to the vectorial definition, magnitude and direction of the feed parameter along the cutter axis is tool path dependent. This capability is essential for the milling simulation of the multi-axis tool paths. Also it is shown that by representing the spindle speed as a vector tangent to the curve of the cutter deflection it is possible to simulate milling process of the statically deflected tools.

For the second task, a special finite element code is developed which uses the mode superposition technique to solve for the dynamics of the workpiece. At the end of each tool rotational time step, the displacement, velocity and acceleration of each mode is saved and used as the initial condition of the workpiece for the next time

step. Accordingly, the dynamic continuity between the successive dynamic analysis of workpiece is preserved, and as the machining simulation continues a continuous and uninterrupted response of the workpiece is computed. For the time period between the disengagement of one tooth and engagement of the approaching one, a free vibration analysis is performed on the workpiece structure.

For the third task, a solid based automatic mesh generation algorithm is developed in chapter 5 which is capable of generating finite elements for the in-process workpiece. The meshing algorithm is synchronized with the tool motion along its path. Partitioning and cellularization techniques are devised to generate the cell data within the volume of the workpiece. Special algorithms are developed for setting the boundary conditions, extracting contact nodes, numbering the local and global nodes and elements, minimizing of the global bandwidth and writing the input files on proper formats.

As more emphasis is placed on the surface quality of the finished products, it is essential to be able to model the tool/workpiece/cutting process interactions in the multi-axis milling of the flexible parts. To this end a dynamic milling force model is developed in chapter 6 which extends the static model to account for the system deflections. The regeneration mechanism in this model includes vibrations of both the cutter and workpiece. Also, by monitoring the effective system deflection at the Surface Generation Point (SGP), the machined surface profile and error is predicted.

Chapter 7 deals with the experimental verification of the developed models and techniques. A twisted blade was machined using a four-axis milling operation. The thickness of the blade was reduced to 2.0 mm for the case of dynamic cutting tests. The static and dynamic cutting forces, as well as the surface profile and errors, are compared with the simulation results. In all comparisons, a good agreement is seen between the experimental data and the simulation results. The post-processing of the Cutter Location (CL) data for the 5-axis Numerical Code (NC) code generation

is also discussed in this chapter.

The contributions made in the course of this research, and future work, are presented in the final chapter.

Chapter 2

Literature review

2.1 Introduction

Methods in milling mechanics, structural dynamics and computer aided geometric simulation are extensively used in the multi-axis milling simulation of flexible parts. A review on each of the above topics will be covered in this chapter.

2.2 Geometry of the milling operation

Milling is a multiple point, interrupted cutting operation. Because of the multiple teeth, the engagement time of each individual tooth is a fraction of the total time of one single tool rotation. The finished surface, accordingly, consists of a series of elemental surfaces generated by the individual cutting edges of the cutter. Due to the general motion of the cutter relative to the workpiece, the uncut (undeformed) chip thickness is not constant but, for up milling, starts with a zero thickness and increases. In the case of down milling it starts with a finite thickness and decreases to zero.

The very early research in milling mechanics dealt with the chip formation mechanism and spindle power estimation. Martelloti's work on the geometry of the milling process established the definition of the tool path and instantaneous chip thickness [Mar41, Mar45]. Martelloti showed that the true path of a milling cutter tooth is trochoidal, but it can be approximated as circular if the radius of the cutter is much larger than the feed per tooth (see Figure 2.1). For this case the chip load is given by

$$t_c = f \ \sin(\psi) \tag{2.1}$$

where t_c and f are the instantaneous chip load and feed per tooth, respectively. Martelloti also derived an expression for the amplitude of the theoretical irregularities

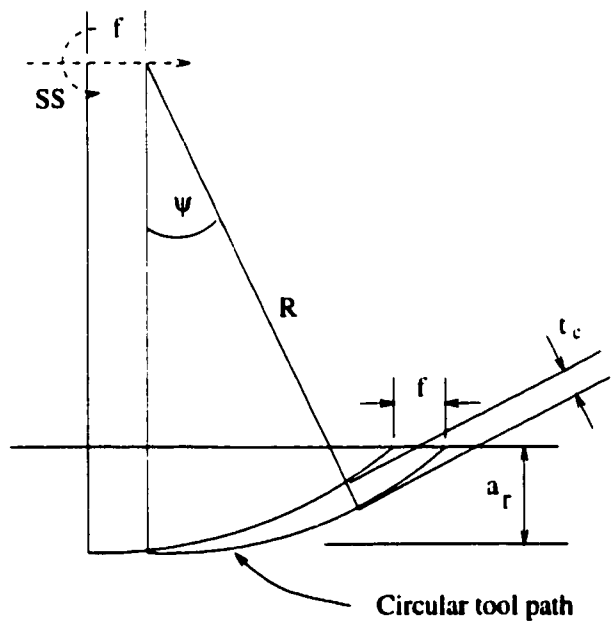


Figure 2.1: Chip thickness variation in the peripheral milling.

(cusps) left on the finished surface from the climb milling (down milling) as follows:

$$H = \frac{f^2}{8R} \tag{2.2}$$

Where H is the height of cusp, f is the feed per tooth and R is the radius of the cutter. The larger the cutter and the smaller the f, the lower the cusp height H. While the frequency of these marks on the surface is equal to that of the teeth, there usually remains a lower frequency wave pattern on the surface of the work which its frequency is equal to that of spindle. The latter is mainly due to the effect of the cutter *runout*.

Geometry of the milling operation, available force models in milling, models of tool and workpiece deflection and machining stability in milling operation will be briefly covered here.

2.3 Milling force models

Force models of milling operation available in the open literature can be divided into two main groups, i.e. rigid and flexible force models. Historically, the rigid force models were the first ones developed in which the milling machine and workpiece are both assumed to be rigid. Accordingly, no feedback was established between the structural deflections and cutting forces. Regardless of flexibility, there are three basic approaches to model cutting forces in metal cutting applications: empirical models, analytical models and semi-empirical models.

2.3.1 Mechanistic force model

The term *mechanistic force model*, which was adopted by many researchers in the analysis of the milling process, is usually referred to as an approach for the milling

force calculation in which the cutting force is related to the undeformed chip thickness through coefficients called *specific cutting pressures*. Accordingly;

$$\begin{aligned} F_T &= K_T\, a_a\, t_c \\ F_R &= K_R\, F_T. \end{aligned} \tag{2.3}$$

where F_T and F_R are the tangential and radial cutting force components, a_a is the axial depth of cut, K_T the specific cutting pressure and K_R is the radial to tangential force ratio. In the above expressions, the cutting force components are linear functions of the undeformed chip thickness t_c. Koenigsberger and Sabberwal [KS66], experimentally showed that the specific cutting pressures are exponential functions of the undeformed chip thickness. In other words, K_T and K_R vary with feed rate, spindle speed as well as axial and radial depths of cut. In order to end up with *constant* specific cutting pressures, Yellowley [Yel85] separated the cutting force generated by the rake face of the tool from that of the cutting edge and flank face. In this approach, although the specific cutting pressures remained constant, two more coefficients were introduced to account for the forces generated by the cutting edge and its flank face. The cutting force model was to be calibrated for these two coefficients for different tool wear. In general, cutting force models developed for the milling process consist the following two fundamental assumptions:

1- The kinematics of milling can be modeled by decoupling the motions of spindle and table. As a result, the cutting edge path can be assumed circular, and equation 2.1 can be used for the chip thickness.

2- The mechanics of machining of any complex process can be modeled by an aggregation of oblique cuts, if the cutting edge is divided into infinitesimal elements. The application of this assumption is to discretize the tool into thin slices, calculate the cutting force applied to each slice of the discretized tool using equations 2.4 and

then sum up the differential cutting forces for all the slices and teeth engaged along each coordinate direction. To gain higher speed in calculation of cutting forces some researchers have explicitly integrated the expressions of cutting force for flat end mills [AS91, JY89].

Also in [AE96] a closed form formulation for the cutting forces of both ball and flat end mills is proposed. This approach provides a new insight to the nature of milling process by replacing the undeformed chip load area with three mutually perpendicular planar areas. As the projected areas navigate through the material, a pair of normal and frictional forces is applied to each projected area. The cutting force components in any tool position is then written by linear combination of those normal and frictional forces. Mathematically, $\{F\} = [K]\{A\}$ where $\{F\}$ is the instantaneous cutting force vector, $\{A\}$ is the projected chip load area (a set of basis functions) and $[K]$ is the *matrix* of the specific pressures and force ratios. The diagonal factors in above relation are associated with the normal forces and the off-diagonal factors are associated with the frictional forces applied to the projected areas. Due to the elimination of numerical integration of the cutting forces the method is very fast.

2.3.2 Analytical force model

Parallel to the mechanistic force models there is another approach for cutting force prediction which is based on an analysis of the mechanics of the metal cutting operation. It defines metal cutting as being a plastic-flow process and tries to derive a set of analytical equations for the cutting forces. However, because of the complexity of the process and presence of a high level of uncertainty associated with some of its aspects, the analytical models can not precisely predict cutting forces in practical cutting conditions. One of the most recognized analytical models is that of Merchant. In this model the cutting force ΔF_R applied to the chip at the tool/chip interface

(under certain conditions and assumptions) is as follows:

$$\Delta F_R = \frac{\tau_s \Delta A_c}{\sin(\phi).\cos(\phi + \beta - \alpha)} \tag{2.4}$$

where ΔA_c is the area of undeformed chip thickness and α is the rake angle the tool. The cutting force is then decomposed to tangential F_S and normal F_N force components (acting on the primary shear plane). Using experimental observations for estimation of the parameters of shear angle (ϕ) and friction angle (β), introduces another class of modeling called the semi-empirical models.

2.3.3 Semi-empirical force model

In this approach, the oblique cutting condition (which is used in practice) is analyzed using orthogonal cutting conditions. Also, empirical equations for shear angle ϕ, friction angle β and shear stress on the shear plane τ_s are used in the analytically derived force equation 2.4. As an example of these types of equations is that of Yang and Park [YP91]. The end turning operation of thin walled tubes was used in developing this set of equations.

$$\begin{aligned} \phi &= 106.7(V.f)^{0.5} + 0.375\alpha + 13.64 \\ \tau_s &= 1.586(V.f)^{-0.25} + 67.703 \\ \beta &= 48.4(V.f)^{0.125} + 25.586 - \phi + \alpha. \end{aligned} \tag{2.5}$$

where V is the cutting speed α is the rake angle and f is feed per tooth. In [YP91], the oblique cutting process of ball end milling has been analyzed as an orthogonal cutting process in the plane containing the local cutting velocity and chip flow. The cutting forces generated in the orthogonal plane is then transformed to the oblique cutting condition based on the assumption that the direction of the friction force is

collinear with the that of the chip flow on the rake face. With availability of faster and more powerful computers, and increasing demand for precision machining, the flexible force models were introduced. Flexibility in a model can be either static or dynamic.

2.3.4 Statically flexible force models

The static deflections of the tool and the workpiece (a thin rectangular web) was studied in [KDS82] and [SD86] where a mechanistic force model of the end mill was used to calculate the cutting forces. In [SD86], the chip load is calculated such that it balances the cutting forces for the resulting system deflections. The tool is modeled by a cantilever beam rigidly supported by the tool holder. The workpiece is modeled by the rectangular plate elements clamped on the three edges with the fourth edge free to deflect. The developed model, provides accurate surface error predictions as long as the system deflections are essentially static. In [Fau64] it is suggested that if the tooth frequency is 10 times less than the lowest natural frequency of the part, then the cutting force may be assumed static. For a thin walled flexible structure, this condition can easily be violated. In the last 15 years, there have been few workers who studied the subject of flexible part machining. In [KDS82], milling of a Clamped-Clamped-Clamped-Free (CCCF) plate with a flexible end mill is studied. Although the finite element method is used to model the part deflections, the interaction between milling forces and the structural deflections is neglected. Accordingly, the model can still be considered as a rigid model.

2.3.5 Dynamically flexible Force models

In [SE90] the model developed in [KDS82] is improved to include the dynamic milling forces and the regeneration mechanism. The mechanism of regeneration in chip thickness calculation makes this model the most comprehensive and accurate one. Accordingly, the cutting force is modeled as:

$$F = K_T a_a (f \sin(\psi) + \delta(t) - \delta(t - T)) \tag{2.6}$$

where the quantity $f \sin(\psi(t))$ is the instantaneous chip thickness in the absence of any deflection in the system, $\delta(t)$ is the instantaneous dynamic deflection in the system (primary feedback) and the quantity $\delta(t - T)$ accounts for the undulations left from the cutting action of the previous teeth (secondary feedback). In [SE90] a $3 - D$ finite element model with the brick elements is used to model the dynamics of a CFFF plate. The peripheral milling of flexible plates is also studied in [Mon90] and [AMB92], where the tool is assumed to be rigid and the plate is modeled by finite element analysis. Finally, milling of thin ribs is studied in [AAK87] where an imprint of the high frequency structural vibrations is predicted by the model. In all above studies, the workpiece geometry is exclusively restricted to a thin plate and the tool path to a straight line. This clearly prevents the applicability of the developed methods and the conclusions obtained to the general cases where the workpiece has a complicated shape and the tool path is not a straight line.

2.4 Structural mechanics

In reality both the cutter and workpiece structures deflect and vibrate under the milling forces. Static deflections cause macro dimensional errors which may violate the tolerance requirements on the machined surfaces. The tool and part vibrations,

on the other hand, result in poor surface quality and chipping the cutting edges. To meet the required dimensional accuracy in the milling of flexible parts the effect of the structural deformations need to be considered. The following section is a brief review on the models of flexible part and flexible tool.

2.4.1 Structural modeling of the workpiece

Free vibration of a multi degree freedom system is governed by [Hum90]

$$\underline{\underline{M}}\,\ddot{\underline{u}} + \underline{\underline{C}}\,\dot{\underline{u}} + \underline{\underline{K}}\,\underline{u} = 0 \tag{2.7}$$

where $\underline{\underline{M}}, \underline{\underline{C}}$ and $\underline{\underline{K}}$ are *matrices* of mass, viscous damping and stiffness of the system. $\underline{u}$ is the solution *vector* of the system. For the undamped case equation 2.7 reduces to the following:

$$\underline{\underline{M}}\,\ddot{\underline{u}} + \underline{\underline{K}}\,\underline{u} = 0 \tag{2.8}$$

The solution for the above equation is proposed to be:

$$\underline{u} = \underline{q}\sin(\omega t + \theta) \tag{2.9}$$

where $\underline{q}$ the vector of amplitudes, ω the frequency of vibrations and θ the phase angle are all unknowns, yet to be determined. Substituting equation 2.9 into 2.8 yields to: $(\underline{\underline{K}} - \omega^2\underline{\underline{M}})\underline{q}\sin(\omega t + \theta) = 0$ which is always satisfied if : $(\underline{\underline{K}} - \omega^2\underline{\underline{M}})\underline{q} = 0$. Alternatively,

$$\underline{\underline{K}}\,\underline{q} = \omega^2\underline{\underline{M}}\,\underline{q}. \tag{2.10}$$

Equation 2.10 represents a linearized eigenvalue problem. A linearized eigenvalue problem can be converted to a standard form by pre-multiplying both sides of 2.10 by $\underline{\underline{M}}^{-1}$, i.e. $\underline{\underline{M}}^{-1}\underline{\underline{K}}\underline{q} = \omega^2(\underline{\underline{M}}^{-1}\underline{\underline{M}})\underline{q}$ which in turn reduces to : $\underline{A}\underline{q} = \lambda\underline{q}$ where λ is equal to ω^2. In general we have : $\underline{\underline{K}}\underline{q}_i = \lambda_i\underline{\underline{M}}\underline{q}_i$ where $\underline{q}_i$ and λ_i are the ith eigenvector and eigenvalue of the system. It can be shown that:

$$\underline{q}_i^T\underline{\underline{M}}\underline{q}_j = \underline{q}_i^T\underline{\underline{K}}\underline{q}_j = 0 \qquad \{\text{if} : i \neq j\} \tag{2.11}$$

which represents the condition of orthogonality among eigenvectors. Scaling of the eigenvectors is called normalization. The method in which eigenvectors satisfy the condition $\underline{\underline{\phi}}_i^T\underline{\underline{M}}\underline{\underline{\phi}}_i = 1$ is called mass orthonormalization. Henceforth, $\underline{\underline{\phi}}_i$ will be used to denote the ith mass orthonormalized eigenvector. In this notation $\underline{\underline{\phi}}_i^T\underline{\underline{M}}\underline{\underline{\phi}}_i = \delta_{ij}$ and $\underline{\underline{\phi}}_i^T\underline{\underline{K}}\underline{\underline{\phi}}_i = \lambda_i\delta_{ij}$ where:

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{Otherwise} \end{cases} \tag{2.12}$$

is the Kronecker delta. The eigenvector of a generalized eigenproblem are called *modes shapes*. The smallest eigenvalue is called the *fundamental frequency* and corresponding mode shape is called the *fundamental mode*. When mass orthonormal mode shapes are used by arranging side by side in an $N \times N$ matrix, it is called *modal matrix* and denoted by $\underline{\underline{\Phi}} = [\underline{\Phi}_1\underline{\Phi}_2...\underline{\Phi}_N]$. It can be shown that [Hum90]

$$\underline{\underline{\Phi}}_T\underline{\underline{M}}\underline{\underline{\Phi}} = \underline{\underline{I}} \tag{2.13}$$

$$\underline{\underline{\Phi}}_T\underline{\underline{K}}\underline{\underline{\Phi}} = \underline{\underline{\Lambda}} \tag{2.14}$$

where $\underline{\underline{\Lambda}}$ is called the *spectral matrix* is a diagonal matrix of eigenvalues λ_i. Since the eigenvectors of a symmetric matrix of order N are *independent*, they constitute a

complete set of basis vectors in an N-dimensional space and can be used to represent any N^{th} order vector. Suppose the eigenproblem is defined as $\underline{\underline{K}}\underline{q} = \lambda\underline{\underline{M}}\underline{q}$, then if eigenvectors are mass orthonormalized, any arbitrary vector x can be represented as follows:

$$x = c_1\underline{\Phi}_1 + c_2\underline{\Phi}_2 + ... + c_N\underline{\Phi}_N \tag{2.15}$$

by pre-multiplying both sides of 2.15 by $\underline{\Phi}_i{}^T\underline{\underline{M}}$ we get: $\underline{\Phi}_i{}^T\underline{\underline{M}}\underline{x} = c_i$.
(Note that $\underline{\Phi}_i{}^T\underline{\underline{M}}\underline{\Phi}_i = 1.0$). Equivalently:

$$\underline{c} = \underline{\underline{\Phi}}^T\underline{\underline{M}}\underline{x} \tag{2.16}$$

The representation of a vector like x as a superposition of the eigenvectors of a matrix as in 2.15 by coefficients of 2.16 is referred to as the *expansion theorem.* The expansion theorem is used in the mode superposition technique for the solution of the multi degree of freedom systems as follows.

2.4.2 Solution of damped free vibration

For a multi degree system with governing equation 2.7 the solution is [Hum90] $\underline{u} = \underline{q}e^{st}$ (with $\dot{\underline{u}} = s\underline{q}e^{st}$ and $\ddot{\underline{u}} = s^2\underline{q}e^{st}$). Substitution of the solution into 2.7 leads to $(s^2\underline{\underline{M}} + s\underline{\underline{C}} + \underline{\underline{K}})\underline{q} = 0$ with nontrivial solution given by:

$$det[s^2\underline{\underline{M}} + s\underline{\underline{C}} + \underline{\underline{K}}] = 0 \tag{2.17}$$

The equation 2.17 is called the *Quadratic eigenvalue problem.* If the system has N degrees of freedom then 2.17 provides $2N$ eigenvalues s and eigenvectors $\underline{q}$. The general solution to the free vibration problem is obtained by superposition of the $2N$

solutions obtained by substituting for $\underline{q}$ and s in the following:

$$\underline{u} = \sum_{n=1}^{2N} \alpha_n \underline{q}_n e^{s_n t} \tag{2.18}$$

where α_n is an arbitrarv constant that should be determined using initial conditions. The quadratic eigenvalue problem of equation 2.17 can be transformed into a linearized form as follows. A vector x is defined of vector of unknown velocities and displacements such that:

$$x = \begin{bmatrix} \dot{\underline{u}} \\ \underline{u} \end{bmatrix} \tag{2.19}$$

Equations of motion of the system in augmented form can be written as follows:

$$\begin{aligned} \underline{\underline{M}}\dot{\underline{u}} - \underline{\underline{M}}\dot{\underline{u}} &= 0 \\ \underline{\underline{M}}\ddot{\underline{u}} + \underline{\underline{C}}\dot{\underline{u}} + \underline{\underline{K}}\underline{u} &= 0.0 \end{aligned} \tag{2.20}$$

The latter equation can be alternatively written as follows:

$$A\dot{x} = Bx \tag{2.21}$$

where $A = \begin{bmatrix} 0 & -\underline{\underline{M}} \\ -\underline{\underline{M}} & -\underline{\underline{C}} \end{bmatrix}$ and $B = \begin{bmatrix} -\underline{\underline{M}} & 0 \\ 0 & \underline{\underline{K}} \end{bmatrix}$. Solution for 2.21 is $x = ve^{st}$. Substituting in 2.21 gives: $BV = sAV$. Pre-multiplying both sides of latter by A^{-1} leads to:

$$A^{-1}BV = sV \tag{2.22}$$

where $A^{-1}B = \begin{bmatrix} -\underline{\underline{M}}^{-1}\underline{\underline{C}} & -\underline{\underline{M}}^{-1}\underline{\underline{K}} \\ I & 0 \end{bmatrix}$. Here again eigenvectors are orthogonal with respect to A and B matrices:

$$V_m^T A V_n = V_m^T B V_n = 0 \tag{2.23}$$

In the case of mass orthonormalization (here A is the mass matrix) one has to satisfy the following: $V_m^T A V_m = 1.0$ where $m = 1, ..., 2N$. Henceforth:

$$V^T A V = I \tag{2.24}$$

$$V^T B V = S \tag{2.25}$$

where I is the unity matrix and S is the *spectral matrix* (diagonal matrix of eigenvalues s). Using the expansion theorem, the solution for the damped free vibration equation is obtained by the following transformation:

$$x = \sum_{n=1}^{2N} V_n y_n = V y \tag{2.26}$$

Substituting 2.26 in to 2.21 yield to: $\dot{x} = V\dot{y} \Rightarrow AV\dot{y} = BVy$. Pre-multiplying both sides by V^T leads to:

$$V^T(AV)\dot{y} = V^T BVy \Rightarrow \dot{y} = Sy \tag{2.27}$$

Since S is diagonal we will have: $\dot{y}_n = s_n y_n$where$\{n = 1, 2, ..., 2N\}$. Solution for the latter is: $y_n = d_n e^{s_n t}$ where d_n is dependent on initial conditions x_0 through $x_0 = V y_0$. Since $y_0 = d$ then $x_0 = Vd$ which yields

$$d = V^T A x_0 \tag{2.28}$$

For the quadratic eigenproblem one therefore has to have eigenvalues S and eigenvectors V. By then having $x_0 = \begin{bmatrix} \dot{\underline{u}}_0 \\ \underline{u}_0 \end{bmatrix}$ (initial values) known, constructing matrix $\underline{\underline{A}}$ (equation 2.21,) finding the modal matrix $\underline{\underline{V}}$ (solving equation 2.22) and finding the vector of coefficients $\underline{d}$ (using equation 2.28,) the solution for the system can be found in normal coordinates, i.e. $\underline{y}_n = \underline{d}_n e^{s_n t}$. Finally, solution of the problem in natural coordinates can be found using $\underline{x} = \underline{\underline{V}}\,\underline{y}$.

In practice, however, this method is very costly, and usually transformation of coordinates associated with an undamped system is used, *hoping* that the transformation matrix $\underline{\underline{\Phi}}^T \underline{\underline{C}}\,\underline{\underline{\Phi}}$ is diagonal:

$$\underline{\underline{M}}\,\ddot{\underline{u}} + \underline{\underline{C}}\,\dot{\underline{u}} + \underline{\underline{K}}\,\underline{u} = 0.0$$

using the transformation: $\underline{u} = \underline{\underline{\Phi}}\,\underline{y} \Rightarrow \ddot{\underline{u}} = \underline{\underline{\Phi}}\,\ddot{\underline{y}}$ and $\dot{\underline{u}} = \underline{\underline{\Phi}}\,\dot{\underline{y}}$,

$$\underline{\underline{\Phi}}^T \underline{\underline{M}}\,\underline{\underline{\Phi}}\,\ddot{\underline{y}} + \underline{\underline{\Phi}}^T \underline{\underline{C}}\,\underline{\underline{\Phi}}\,\dot{\underline{y}} + \underline{\underline{\Phi}}^T \underline{\underline{K}}\,\underline{\underline{\Phi}}\,\underline{y} = 0.0 \tag{2.29}$$

For a system with a *proportional damping*, i.e. $\underline{\underline{\Phi}}^T \underline{\underline{C}}\,\underline{\underline{\Phi}}$ being diagonal, then equation 2.29 is uncoupled:

$$\ddot{y}_0 + c_n \dot{y}_0 + \omega_n^2 y_n = 0.0 \qquad \{n = 1, 2, ..., 2N\} \tag{2.30}$$

with the solution:

$$y_n = e^{-\zeta_n \omega_n t} \left(\frac{\dot{y}_{0n} + y_{0n}\zeta_n\omega_n}{\omega_{0n}} \sin(\omega_{D_n} t) + y_{0n} \cos(\omega_{D_n} t) \right) \tag{2.31}$$

where $\omega_{D_n} = \omega_n \sqrt{1 - \zeta_n^2}$ and $\zeta_n = \frac{c_n}{2\omega_n}$ are the damped frequency and damping ratio of the nth mode.

2.4.3 Force dynamic response of multi degree systems

The governing equation system is as follows:

$$\underline{\underline{M}}\,\underline{\ddot{u}} + \underline{\underline{C}}\,\underline{\dot{u}} + \underline{\underline{K}}\,\underline{u} = \underline{P}(t) \tag{2.32}$$

As before one may use the transformation:

$$\underline{u} = \sum_{n=1}^{N} \underline{\phi}_n \underline{y}_n = \underline{\underline{\Phi}}\,\underline{y} \tag{2.33}$$

where $\underline{\underline{\Phi}}$ is the matrix of the *undamped* mode shapes and $\underline{y}$ is the vector of normal coordinates. Using transformation of equation 2.33 in equation 2.32 leads to:

$$\underline{\underline{M}}^*\underline{\ddot{y}} + \underline{\underline{C}}^*\underline{\dot{y}} + \underline{\underline{K}}^*\underline{y} = \underline{p}^* \tag{2.34}$$

where $\underline{\underline{M}}^* = \underline{\underline{\Phi}}^T\underline{\underline{M}}\,\underline{\underline{\Phi}} = \underline{\underline{I}}$ (if mode shapes are normalized to be mass orthonormal) $\underline{\underline{C}}^* = \underline{\underline{\Phi}}^T\underline{\underline{C}}\,\underline{\underline{\Phi}}$ (could be diagonal if the damping of the system is proportional) $\underline{\underline{K}}^* = \underline{\underline{\Phi}}^T\underline{\underline{K}}\,\underline{\underline{\Phi}}$ and $\underline{p}^* = \underline{\underline{\Phi}}^T\underline{P}(t)$. Then equation 2.34 reduces to a set of N uncoupled equations:

$$\ddot{y}_n + 2\zeta_n\omega_n\dot{y}_n + \omega_n^2 y_n = \frac{p_n}{M_n} \tag{2.35}$$

where $\frac{C_n}{M_n} = 2\zeta_n\omega_n$ and $\frac{K_n}{M_n} = \omega_n^2$. The complete solution for 2.35 consists of free vibration and forced response of the system is as follows:

$$y_n = e^{-\zeta_n\omega_n t}\left(\frac{\dot{y}_{0n} + y_{0n}\zeta_n\omega_n}{\omega_{0n}}\sin(\omega_{D_n}t) + y_{0n}\cos(\omega_{D_n}t)\right) +$$

$$\frac{1}{M_n\omega_{D_N}}\int_0^t p_n(\tau)e^{-\omega_{D_n}\zeta_n(t-\tau)}\sin\omega_{D_n}(t-\tau).d\tau \tag{2.36}$$

where $\dot{y}_{0n}$ and y_{0n} are initial conditions of each mode and $\omega_{D_n} = \omega_n\sqrt{1-\zeta_n^2}$. For a complete solution of the system, equation 2.36 has to be solved for each of the N modes and then, using $\underline{u} = \underline{\underline{\Phi}}\underline{y}$, the physical response of the system to be determined.

2.4.4 Finite element and vibrational analysis

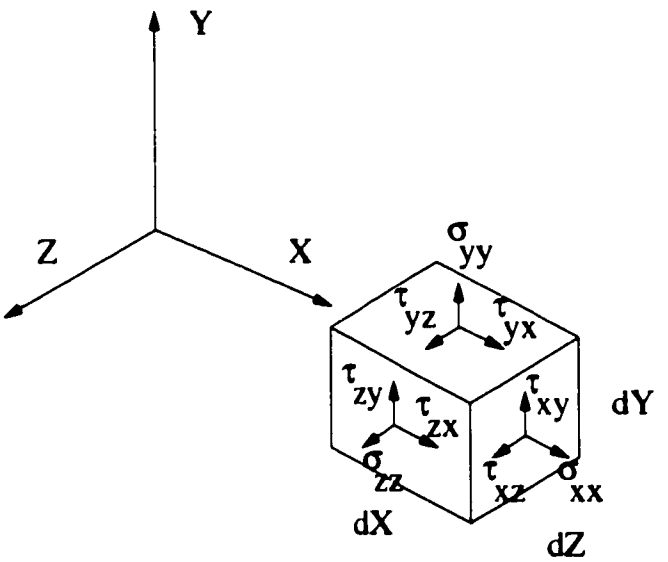


Figure 2.2: Stress tensor on an infinitesimal element.

An infinitesimal element in Cartesian coordinates; where the edges are of the lengths dx,dy and dz is shown in Figure 2.2. From the equilibrium condition relations among the shear stresses are as follows:

$$\tau_{xy} = \tau_{yx}, \quad \tau_{yz} = \tau_{zy}, \quad \tau_{zx} = \tau_{xz}$$

By definition, the longitudinal strains are:

$$\epsilon_x = \frac{\partial u}{\partial x}, \quad \epsilon_y = \frac{\partial v}{\partial y}, \quad \epsilon_z = \frac{\partial w}{\partial z},$$

where u, v and w are the translation functions along the x, y and z directions. On the other hand, the shear strains by definition are:

$$\begin{aligned}
\gamma_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = \gamma_{yx} \\
\gamma_{yz} &= \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} = \gamma_{yz} \\
\gamma_{zx} &= \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} = \gamma_{xz}
\end{aligned}$$

The tensors of stress and strain can be arranged in two vectors $\underline{\sigma}$ and $\underline{\epsilon}$ as follows:

$$\underline{\sigma} = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix} = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{bmatrix}$$

$$\underline{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{bmatrix} = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix}$$

From the continuity condition the relations between stresses and strains are:

$$\begin{aligned}
\epsilon_x &= \frac{1}{E}(\sigma_x - \nu\sigma_y - \nu\sigma_z), \quad \gamma_{xy} = \frac{\tau_{xy}}{G} \\
\epsilon_y &= \frac{1}{E}(\sigma_y - \nu\sigma_x - \nu\sigma_z), \quad \gamma_{yz} = \frac{\tau_{yz}}{G} \\
\epsilon_z &= \frac{1}{E}(\sigma_z - \nu\sigma_y - \nu\sigma_x), \quad \gamma_{zx} = \frac{\tau_{zx}}{G}
\end{aligned}$$

where $G = \frac{E}{2(1+\nu)}$. Consequently, the constitutive matrix will be as follows:

$$\underline{\underline{C}} = \frac{1}{E}\begin{bmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(1+\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(1+\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(1+\nu) \end{bmatrix}$$

Hence, the stress strain relation within the continua can be written as: $\underline{\epsilon} = \underline{\underline{C}}\underline{\sigma}$.

Equation of motion for finite elements

The principle of virtual work is used to develop equation of motions. Let the time-varying generic displacements $\underline{U}(t)$ within a finite element be expressed as a column vector:

$$\underline{U} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

also written as :$\underline{u}(t) = \{u, v, w\}$. The time varying body forces are as follows:

$$\underline{b} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}$$

also written as: $\underline{b}(t) = \{b_x, b_y, b_z\}$. If the element had n_n number of nodes then the nodal displacement vector $\underline{q}(t)$ will be: $\underline{q}(t) = \underline{q_i}(t) \quad i = 1, 2, \cdots, n_n$ where:

$$\underline{q_i} = \begin{bmatrix} q_{xi} \\ q_{yi} \\ q_{zi} \end{bmatrix} = \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix} \quad i = 1, 2, \cdots, n_n \tag{2.37}$$

Time varying nodal forces $\underline{P}(t)$ could be written as:

$$\underline{p_i} = \begin{bmatrix} p_{xi} \\ p_{yi} \\ p_{zi} \end{bmatrix}$$

also written as: $\underline{p}(t) = \{p_{xi}, p_{yi}, p_{zi}\} \quad i = 1, \cdots, n_n$. If $\underline{\underline{f}}$ is the matrix of the displacement shape functions, then: $\underline{U}(t) = \underline{\underline{f}}\,\underline{q}(t)$ where $\underline{\underline{d}}$ is the matrix of the linear differential operator. For the vector of strains one will have:

$$\underline{\epsilon}(t) = \underline{\underline{d}}\,\underline{u}(t) = \underline{\underline{d}}\,\underline{\underline{f}}\,\underline{q}(t) = \underline{\underline{B}}\,\underline{q}(t) \tag{2.38}$$

where $\underline{\underline{B}} = \underline{\underline{d}}\underline{\underline{f}}$. Finally, the following relation can be written for the vector of stress components:

$$\underline{\sigma} = \underline{\underline{E}}\underline{\epsilon} = \underline{\underline{E}}\underline{\underline{B}}\underline{q}(t). \tag{2.39}$$

2.4.5 Virtual work principle

If a general structure in dynamic equilibrium is subjected to a system of small virtual displacement within a compatible state of deformation, the virtual work of external actions (forces) is equal to virtual strain energy of internal stresses, i.e. $\delta u_e = \delta w_e$. Denoting virtual displacement of the nodes to be: $\delta \underline{q} = \{\delta q_i\} \quad i = 1, \cdots, n_n$ then the virtual displacement within the element will be: $\delta \underline{u} = \underline{\underline{f}}\delta \underline{q}$. From equation 2.38 virtual strain generated by the virtual displacement will be: $\delta \underline{\epsilon} = \underline{\underline{B}}\delta \underline{q}$. Using the virtual strain then the virtual internal energy generated in the element will be:

$$\delta u_e = \int_V \delta \underline{\epsilon}^T \underline{\sigma} dV \tag{2.40}$$

where V is the volume of the element. The forces applied to the element are :

- body forces, i.e. $b_i(t).dV \quad i = x, y, z$.
- internal forces, i.e. $\rho \ddot{u} dV, \rho \ddot{v} dV \rho \ddot{w} dV$ due to accelerations $\ddot{u}, \ddot{v}$ and $\ddot{w}$.
- real external forces $\underline{p}(t)$.

Hence, the virtual work of forces will be:

$$\delta w_e = \delta \underline{q}^T \underline{p}(t) + \int_V \delta \underline{u}^T \underline{b}(t) - \int_V \delta \underline{u}^T \rho \ddot{u} dV \tag{2.41}$$

Equating $\delta u_e = \delta w_e$ from the equations 2.40 and 2.41 leads to:

$$\int_V \delta \underline{\epsilon}^T \underline{\sigma} dV = \delta \underline{q}^T \underline{p}(t) + \int_V \delta \underline{u}^T \underline{b}(t) - \int_V \delta \underline{u}^T \rho \ddot{\underline{u}} dV \tag{2.42}$$

substituting $\ddot{\underline{u}} = \underline{\underline{f}}\ddot{\underline{q}}$, $\delta \underline{u}^T = \delta \underline{q}^T \underline{\underline{f}}^T$ and $\delta \underline{\epsilon}^T = \delta \underline{q}^T \underline{\underline{B}}^T$ leads to the following:

$$\underline{\underline{M}}\ddot{\underline{q}} + \underline{\underline{K}}\underline{q} = \underline{p}(t) + \underline{p}_b(t) \tag{2.43}$$

where:

$$\underline{\underline{M}} = \int_V \delta \underline{u}^T \rho \ddot{\underline{u}} dV = \int_V \rho \underline{\underline{f}}^T \underline{\underline{f}} dV \tag{2.44}$$

$$\underline{\underline{K}} = \int_V \delta \underline{\epsilon}^T \underline{\sigma}(t) dV = \int_V \underline{\underline{B}}^T \underline{\underline{E}}\,\underline{\underline{B}} dV \tag{2.45}$$

$$\underline{p}_b = \int_V \delta \underline{u}^T \underline{b}(t) dV = \int_V \underline{\underline{f}}^T \underline{b}(t) dV$$

2.4.6 Structural modeling of the tool

The structure of a machine tool, like any other continuous system has an infinite number of degrees of freedom and corresponding number of modal shapes and natural frequencies. However, experimental evidence show higher modes have less contribution in the resulting vibration of the structure. So, for an acceptable accuracy all but a few lower modes can be neglected [TI81]. Accordingly, vibration of the machine/tool can be modeled as shown in Figure 2.3. If the cutting forces are applied to the system of Figure 2.3 because of the relative vibration between the tool and the workpiece, i.e. x and y deflections, a wavy surface will be left behind. As the next tooth cuts into the undulated surface of the work the vibration of the system undergoes a change. If the changes are towards the diminishing of the vibrations (or reducing the amplitude) it is a stable cutting condition and if the vibration grows

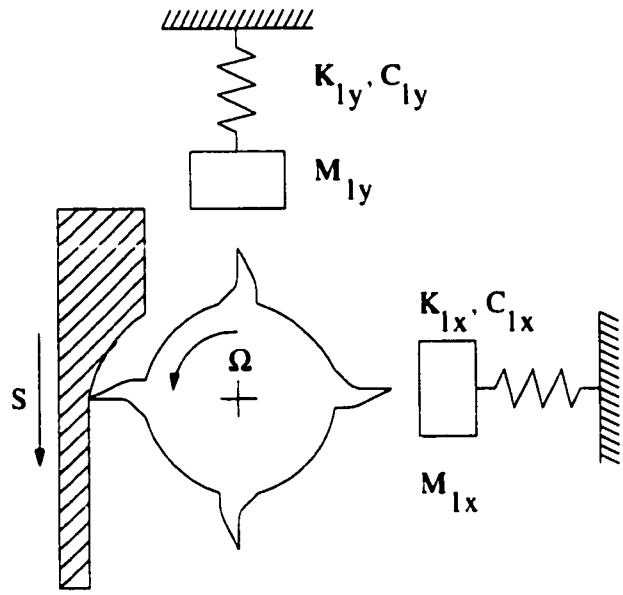


Figure 2.3: Structural model for the tool dynamic deflection.

gradually and stabilizes at a high amplitude, it is regarded as an unstable cutting condition or *chatter*. The necessary modal parameters (stiffness, mass and damping) of the tool and the spindle are obtained using impact testing. For the case that the chatter vibrations are expected to occur at the workpiece mode frequencies (due to higher flexibility) then the dynamic modeling of the cutter may not be necessary. For this case a static deflection model is suggested [AMB92]. The cutter is modeled as a cantilever beam with linear springs at the fixed end to account for the stiffness at the collet(see Figure 2.4). Then, the static deflection of the end mill under the cutting forces can be calculated from beam theory [TM40]. The deflection in x and y direction at axial position z_k caused by force applied at z_m is given by the cantilever beam formulation as:

$$\delta(k,m) = \begin{cases} \frac{F_m \nu_m^2}{6EI}(3\nu_m - \nu_k) + \frac{F_m}{k_c}, & 0 < \nu_k < \nu_m \\ \frac{\Delta F_{y,m} \nu_m^2}{6EI}(3\nu_k - \nu_m) + \frac{\Delta F_m}{k_c}, & \nu_m < \nu_k \end{cases} \tag{2.46}$$

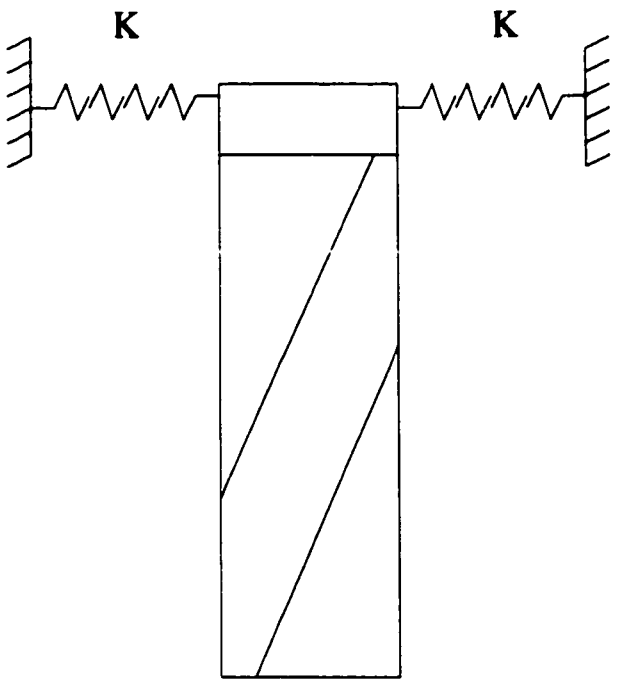


Figure 2.4: Structural model for the tool static deflection.

where E is the Young Modulus, I is the area moment of inertia of the tool, k_c is the experimentally measured tool clamping stiffness in the collet and $\nu_k = l - z_k$, l being the gauge length of the cutter. The area moment of the tool is calculated by using an equivalent tool radius of $R_e = sR$, where $s \simeq 0.8$ is the scale factor due to flutes [KV90].

2.5 Three dimensional geometric modeling

The desire to create symbolic models of the physical world has motivated generations of mathematicians. However, it is only comparatively recently that the knowledge of geometry accrued over the years has been used in automated systems known as *geometric modelers*. Three dimensional geometric modeling can be divided into three groups, namely wire-frame, surface and solid modeling [LaC94, Zei91]. In this section

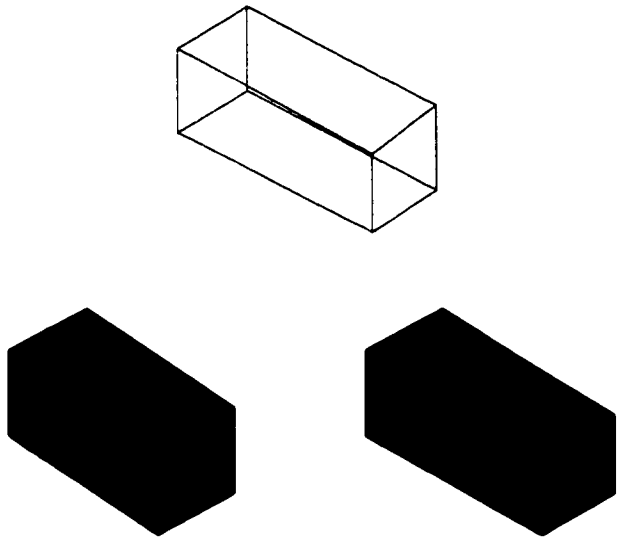


Figure 2.5: Ambiguity in wire-frame interpretation of a simple block.

a brief description of each group is presented, followed by the NURBS representation of free form surfaces.

Early automated drawing packages were wire-frame modeling systems. That is, systems in which only the edges and vertices of objects are represented. Because of ambiguities that may arise in interpreting wire-frame drawings, wire-frames are not the preferred object representation (see Figure 2.5.) On the other hand, wire-frame objects have small storage requirements and can be accessed and displayed quickly. For this reason, many modeling systems retain the capability to generate wire-frame drawings. Surface modeling, on the other hand, provides a more complete and less ambiguous representation than the wire-frames. The mathematical roots of surface design algorithms are in the field of approximation theory. Surface design with splines originated in the automobile industry (car body design), in the ship-building industry (ship hull design), and in the aircraft industry (design of wings and fuselage.) Free-form surface design in these areas has led to the field of Computer Aided Geometric Design (CAGD). However, due to the lack of topology in surface modeling (capability

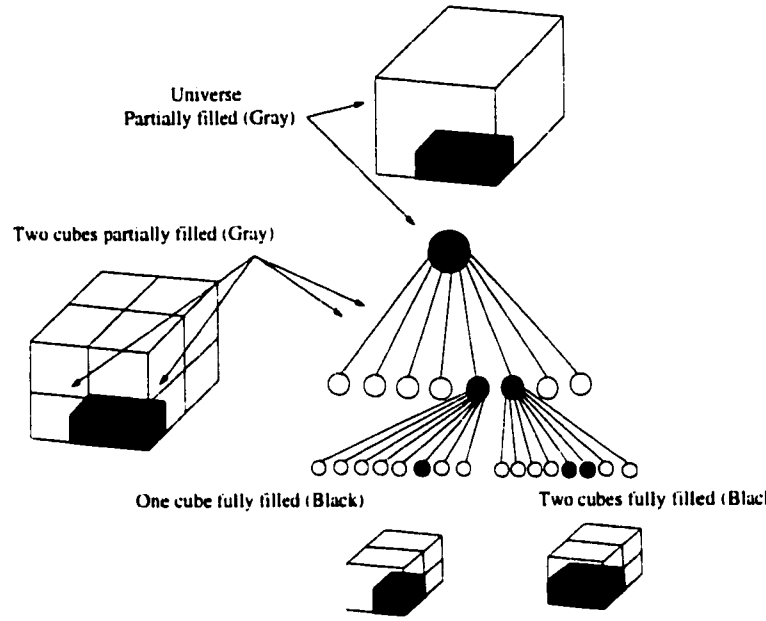


Figure 2.6: Octree representation

of establishing spatial relationship among the different entities), surface models are difficult to modify.

Solid modeling is the outgrowth of several convergent developments, including automatic drafting systems and free-form surface design. An important contributor for the solid modeling has been the free-form surface design, using parametric surface patches and various types of spline surfaces. There are three dominant representation schemes used in solid modeling, i.e. spatial enumeration using octrees, constructive solid modeling (CSG) and boundary representation (B-rep).

In the spatial enumeration approach the entire modeling universe is just a very big cube (the root node in Figure 2.6.) Based on the fact that in a regular division of the space, each cube can be divided into each smaller cubes, an *octree* structure is constructed. Accordingly, the root node defines the universe and the next eight children represent the eight cubical octants of the universe. This subdivision continues

to the required level of resolution. Since the octree method does not provide an exact representation of the object, it has limited applications.

In CSG, a solid is represented as a Boolean expression of *primitive* solid objects. The primitive structures include blocks, spheres, cones, toruses or any other solid object with an analytical representation (see Figure 2.7.) CSG objects are usually stored

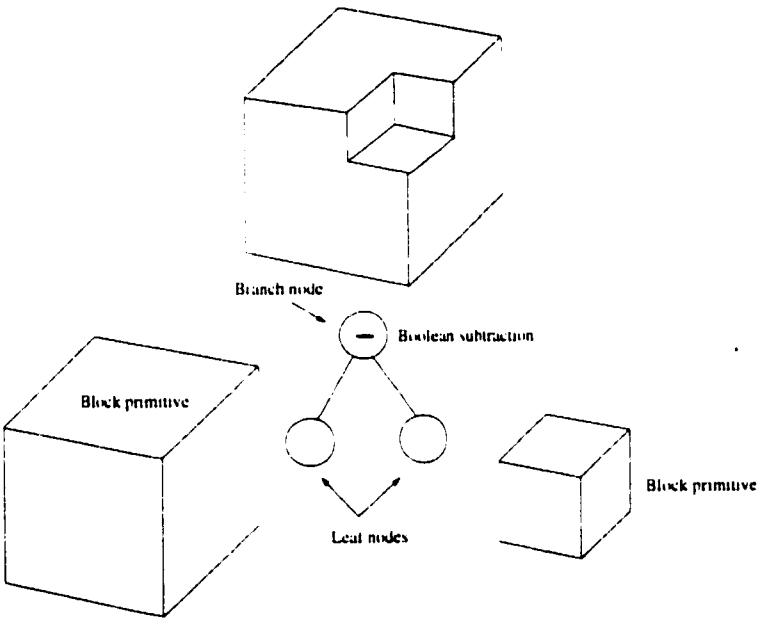


Figure 2.7: CSG representation

in a tree data structure with the leaf and branch nodes representing the primitives and Boolean operations, respectively. Although the CSG representations are very compact, they are limited to the geometry defined by the primitives which limits the CAM applications of the CSG solid models. On the other hand, B-rep describes only the oriented surfaces of a solid as a data structure composed of vertices, edges and faces. The orientation convention permits one to decide on which side of the surface the interior of solid is located. Description of an object using B-rep has two parts, *topological* and *geometric*. Briefly, the topological description specifies VERTEX,

EDGE, and FACE entities in an abstract form and indicates their incidences and adjacencies. The geometric representation, on the other hand, specifies the *equations* of the surfaces associated with each FACE of the object. These equations are used to embed each FACE in 3D-space. Among these equations are the NURBS (Nonuniform Rational B-Spline.) An overview of the NURBS curve and surface formulation, which is widely used in B-rep solid modeling, is covered next.

2.5.1 NURBS representation of curves and surfaces

A pth degree NURBS curve is defined by [PT97]

$$\mathbf{C}(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i \mathbf{P}_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \tag{2.47}$$

where the $\mathbf{P}_i$ are the *control points*, the w_i are the *weights* and the $N_{i,p}(u)$ are the pth B-spline basis functions defined on the non-periodic knot vector

$$U = \{\underbrace{a, \ldots, a}_{p+1}, u_{p+1}, \ldots, u_{m-p-1}, \underbrace{b, \ldots, b}_{p+1}\} \tag{2.48}$$

The ith B-spline basis functions of degree p, denoted by $N_{i,p}(u)$, is defined as

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u \leq u_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{2.49}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{2.50}$$

Figure 2.8 is graphical representation of basis functions of degree 3. The following is an intuitive definition for a surface which is the foundation of the NURBS surface

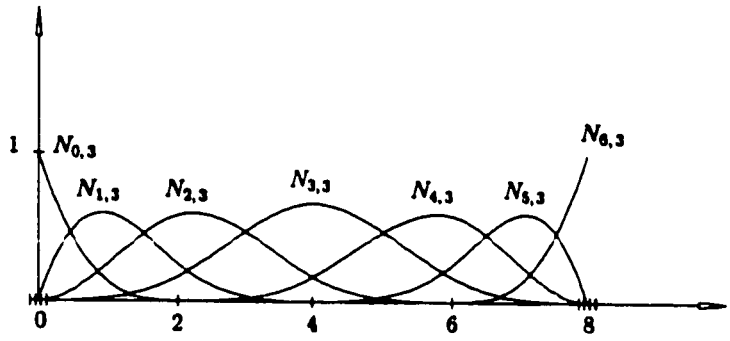


Figure 2.8: Cubic B-spline basis functions [PT97].

definition. *A surface is the locus of a curve that is moving through space and thereby changing its shape.* Accordingly, a NURBS surface of degree p in the u direction and degree q in the v direction is defined as a bivariate vector-valued piecewise rational function

$$\mathbf{S}(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}\mathbf{P}_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}} \tag{2.51}$$

where $\mathrm{P}_{i,j}$ form a bidirectional control net, $w_{i,j}$ are the weights, and $N_{i,p}(u)$ and $N_{j,q}(v)$ are the non-rational B-spline basis functions defined on the knot vectors

$$U = \{\underbrace{0,\ldots,0}_{p+1}, u_{p+1},\ldots,u_{m-p-1}, \underbrace{1,\ldots,1}_{p+1}\} \tag{2.52}$$

$$V = \{\underbrace{0,\ldots,0}_{q+1}, u_{q+1},\ldots,u_{m-q-1}, \underbrace{1,\ldots,1}_{q+1}\} \tag{2.53}$$

Chapter 3

Milling geometric simulation

Geometric simulation of the machining process, in particular the milling operation, is among the applications of solid modeling. There are three main approaches for the Boolean operation between the tool and workpiece. The image based approach, which encompasses the work of Wang [Wan88], Takata et al. [TTS89], Yamazaki et al [YKSS91], uses the ray casting technique to reduce the 3D Boolean operation to one dimension. The surface based (point-vector) approach, developed by Fussell and Ersoy [FE92], uses workpiece surface points and normal vectors to intersect with the solid model representation of the swept volume. The removed volume of material then can be used for the average force calculation. Dewaele and Kinzel [DK89], use solid representations of both the tool and workpiece to determine their contact face. In this thesis, the workpiece solid model and the tool swept volume (solid) are used for the geometric simulation of the milling process. The cutting edges of the various end mill cutters, required for the simulation of the cutting process, are modeled using B-splines.

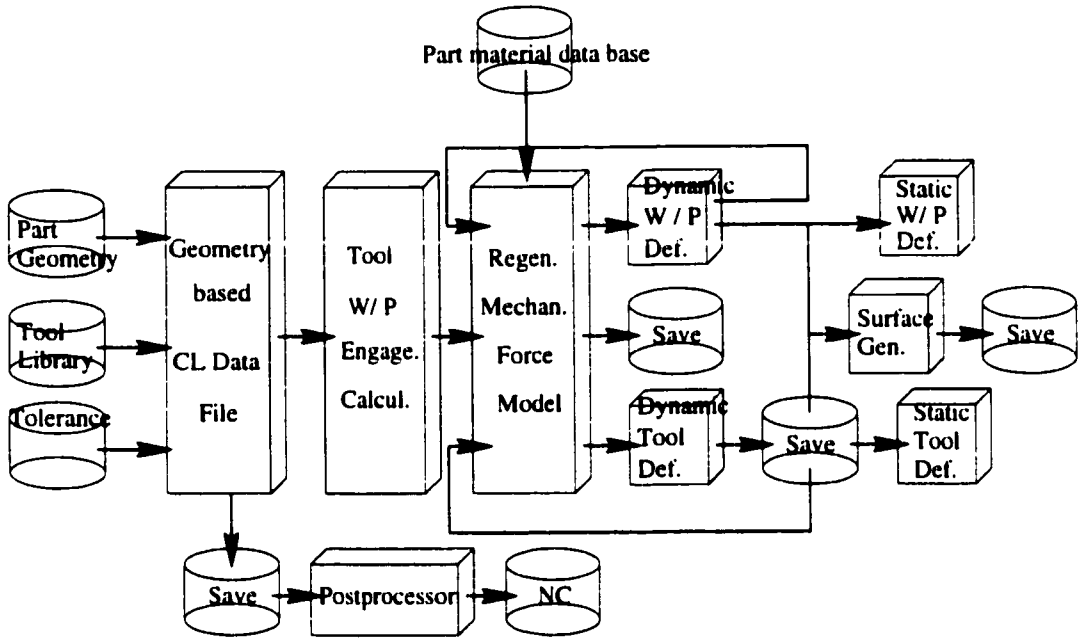


Figure 3.1: CAD/CAM architecture for the side milling of flexible parts.

3.1 Integrated CAD/CAM architecture for side milling

For side milling simulation of the flexible parts, an integrated Computer Aided Design (CAD)/ Computer Aided Manufacture (CAM) architecture is proposed (see Figure 3.1.) The part and tool geometries, part material data base, NC interpolation tolerance, and cutting conditions are the user inputs to the software. The instantaneous cutting forces and deflections, the surface generated on the flexible workpiece and the CL data file are the outputs of the software. The required NC code for the actual machining is then generated by the post-processing of the CL data file. A complete geometric simulation of the milling operation consists of the following geometric representations:

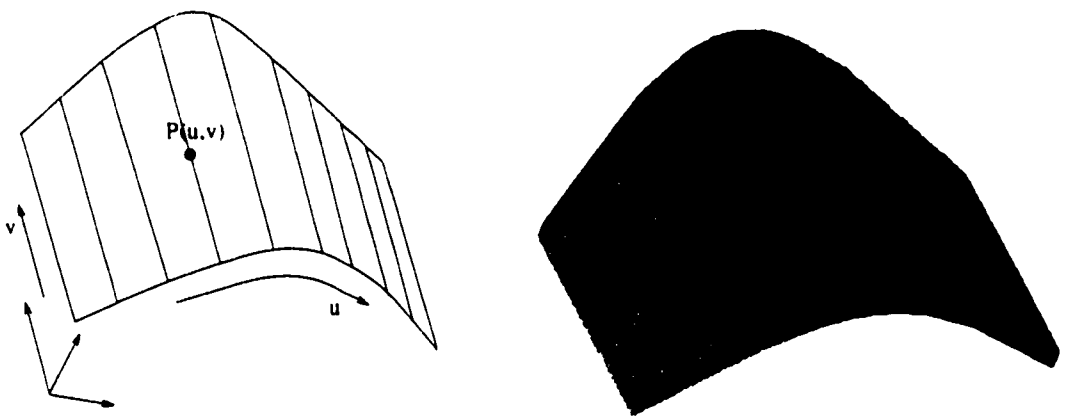
1. Boundary representation (B-rep) of the workpiece geometry,

2. CL data generation,

3. B-rep of the tool swept volume,
4. Boolean subtraction of the tool/workpiece B-reps,
5. Representation of the cutting edge, extraction of the contact face and the engaged segments of the cutting edge.

In this thesis, all the geometric aspects of the milling simulation are implemented using the ACIS 3D B-rep geometric kernel [Inc95].

3.2 Workpiece geometric representation



(a) Ruled surface (b) Part solid model

Figure 3.2: A ruled surface and its part solid model.

Peripheral milling in turbo-machinery blade production eliminates the necessity of forms and molds and, hence reduces the cost. More importantly, avoiding a specific manufacturing method, the blade design can be optimized based on the flow characteristics only [REW89a]. Among the geometric requirements for the milling simulation of the turbo-machinery blades is a solid model representation of blade geometry. In the past decade, there has been a strong interest in developing a general

formulation for the blade surface design [Cas82, SM73]. To achieve a general and more flexible method for the blade surface design, a ruled surface based mathematical model is introduced in [REW89b]. Ruled surfaces allow curvature along one space parameter u (see Figure 3.2(a).) The curvature along the other parameter v is zero. This feature of ruled surfaces makes them ideal for peripheral milling. After designing the surface geometry of a blade as a ruled surface, the procedure of constructing its solid model is as follows:

- *One sided* solid sheets are constructed using the boundary faces of the blade (including the non-operational surfaces of the part)
- The one sided sheets are then orderly *stitched* together. The outside directions of the boundary faces should match with those of the real solid object.

An arbitrary form ruled surface and its solid model are shown in Figures 3.2(a) and 3.2(b).

3.3 Cutting edge representation

Shown in Figure 3.3 is the geometry of a tapered end mill with a *constant* helix angle. The origin of the coordinate system is at the apex of the cone. Half of the cone apex angle is denoted by the angle Ω and the cutting edge helix angle (measured on the cone) is denoted by the angle i. The position vector of a point P on the edge can be expressed as follows:

$$\vec{r}(\psi) = R(\psi)\cos\psi\ \vec{i} + R(\psi)\sin\psi\ \vec{j} + Z(\psi)\ \vec{k}$$

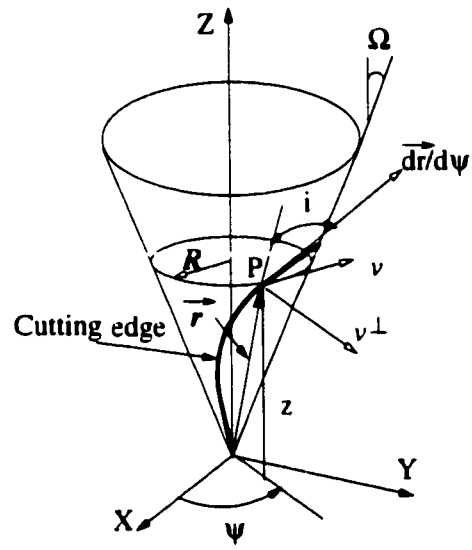


Figure 3.3: Cutting edge geometry of a tapered end mill.

where ψ is the rotational position of the point P and $R(\psi)$ is the local radius of the tool at that point. Obviously, $Z(\psi) = R(\psi)/\tan(\Omega)$ which leads to:

$$\vec{r}(\psi) = R(\psi)\cos\psi\ \vec{i} + R(\psi)\sin\psi\ \vec{j} + \frac{R(\psi)}{\tan\Omega}\ \vec{k} \tag{3.1}$$

From the definition of the helix angle one can write:

$$\sin(i) = \frac{1}{\left|\frac{d\vec{r}}{d\psi}\right|}\frac{d\vec{r}}{d\psi}\cdot\vec{v} \tag{3.2}$$

Also for the given position ψ, the unit vectors $\vec{v}^{\perp}$ and $\vec{v}$ can be expressed as follows:

$$\begin{aligned} \vec{v}^{\perp} &= \cos(\psi)\ \vec{i} + \sin(\psi)\ \vec{j} \\ \vec{v} &= -\sin(\psi)\ \vec{i} + \cos(\psi)\ \vec{j}. \end{aligned} \tag{3.3}$$

Using equation 3.1, the derivative of the position vector $\vec{r}$ with respect to parameter ψ is:

$$\frac{d\vec{r}}{d\psi} = (R'\cos\psi - R\sin\psi)\,\vec{i} + (R'\sin\psi + R\cos\psi)\,\vec{j} + \frac{R'}{\tan\Omega}\vec{k} \tag{3.4}$$

The length of the vector $d\vec{r}/d\psi$ will be:

$$\left|\frac{d\vec{r}}{d\psi}\right| = \left[R^2 + \frac{R'^2}{\sin^2\Omega}\right]^{1/2} \tag{3.5}$$

Where R' is the derivative of R with respect to the parameter ψ. Substituting equations 3.3, 3.4 and 3.5 in to equation 3.2 and simplifying one gets:

$$\frac{dR}{R} = \frac{\sin\Omega}{\tan i}d\psi.$$

which has the following solution:

$$\ln R = \frac{\sin\Omega}{\tan i}\cdot\psi + C \tag{3.6}$$

Let the radius of the tapered end mill be denoted by R_n, i.e. the nominal radius, and further assume $\psi = 0.0$ for that position. Using the above boundary condition in equation 3.6 we get $C = \ln R_n$. Substituting for C in 3.6 results in the following equation for R:

$$R = R_n e^{\frac{\sin\Omega}{\tan i}\psi} \tag{3.7}$$

Finally, substituting equation 3.7 in 3.1 yields the equation of the cutting edge for tapered end mills with constant helix angle as follows:

$$\vec{r}(\psi) = R_n e^{\frac{\sin\Omega}{\tan i}\psi} \cos\psi \ \vec{i} + R_n e^{\frac{\sin\Omega}{\tan i}\psi} \sin\psi \ \vec{j} + \frac{R_n e^{\frac{\sin\Omega}{\tan i}\psi}}{\tan\Omega} \vec{k} \tag{3.8}$$

For a conventional end mill with radius R the equation 3.8 reduces to the following:

$$\vec{r}(\psi) = R\cos\psi \ \vec{i} + R\sin\psi \ \vec{j} + \frac{R\ \psi}{\tan i} \ \vec{k} \tag{3.9}$$

For ball end mills with the nominal radius R an equation similar to 3.8 can be developed in spherical coordinate system as follows [Abr93]:

$$\vec{r}(\psi, \theta) = R\sin(\theta)\cos(\psi)\vec{i} + R\sin(\theta)\sin(\psi)\vec{j} - R\cos(\theta)\vec{k} \tag{3.10}$$

where $\psi = \mp\tan i \mp \frac{2}{\sqrt{3}}\tanh^{-1}(\sqrt{3}\tan i) \pm \frac{6}{\sqrt{3}}\tanh^{-1}(\frac{\sqrt{3}}{3}\tan i)$. In the development of this equation the variation of the inclination angle along the cutting edge is approximated by $i = \theta/6$. Equivalently, the ball end mill cutting edge equation can be written by [YP91];

$$\vec{r}(\theta) = R\cos\alpha_n \sin\theta \ \vec{i} + R\sin\alpha_n \cos\alpha_n(1 - \cos\theta) \ \vec{j} - R(\cos^2\alpha_n \cos\theta + \sin^2\alpha_n) \ \vec{k} \tag{3.11}$$

In the geometric simulation of the milling operation, a *B-spline* curve is usually used to represent the cutting edge of the tool [EMSE96]. The B-spline curves shown in Figure 3.4 model the cutting edges of a typical tapered end mill. Geometric modeling of the end mill cutting edges involves two steps. First, for a given tool type/size, a set of interpolation points are generated using one of the vector functions 3.8, 3.9 or 3.11. A spline curve is then *interpolated* using the generated set of points and the

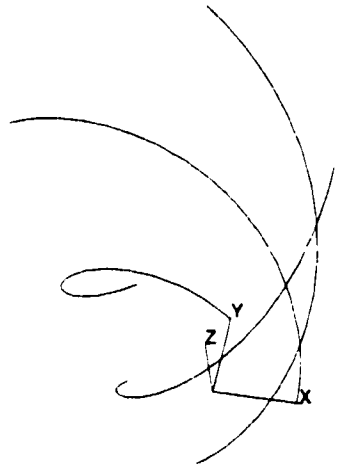
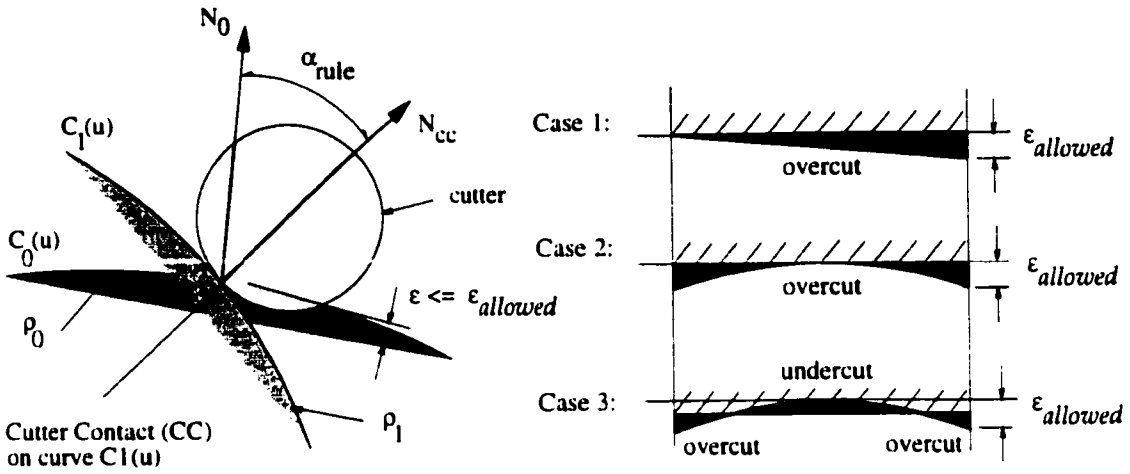


Figure 3.4: B-spline representation of tapered end mill cutting edges.

boundary slopes. The kinematics of a multi-axis milling machine, i.e. rotation of the spindle and translation/orientation of the machine table, can be modeled by the rotational and translational/rotational motion of the cutting edges, respectively.

3.4 CL data generation

The Single Point Offset (SPO) method is used in this thesis for the CL data generation. The CL data file contains information regarding the tool position/orientation relative to the machining surface, i.e. Workpiece Coordinate System (WCS). In SPO setting, tool is placed at a tangent to one of the two rails of the surface, as shown in Figure 3.5(a). In the case of non-developable ruled surfaces, i.e. surfaces that can not be deformed to a planar shape without changing length measurements in them, tool orientation (setting) should be calculated for a minimum interference. To distribute the machining error along the ruled direction, the Cutter Contact point (CC) can be placed on the middle of the ruled line (see case 2 in Figure 3.5(b)). Moreover, to decrease the cutting error evenly (case 3 in Figure 3.5(b)), the calculated tool center



(a) Single point offset

(b) Error distribution

Figure 3.5: Single Point Offset in side milling of ruled surfaces.

can be shifted along the surface normal at CC point by $\epsilon_{allowed}/2$. The surface error distribution for all three cases is shown in Figure 3.5(b). According to [Liu95], the machining interference for the single point offset orientation is

$$\epsilon_{Max} = -R_t(1 - \cos\alpha_{rule}) \tag{3.12}$$

Computations of CL data in SPO method is based on this observation that the axis of the tool is to be parallel with the generatrix of the surface, which leads to the following CL position

$$\vec{r_0} = \vec{r_{cc}} + R\,\vec{N_{cc}} \tag{3.13}$$

where $\vec{r_o}$ is the position vector of the tip of the tool, $\vec{r_{cc}}$ and $\vec{N_{cc}}$ are the positional and surface normal vectors at the cutter contact point. The orientation of the rule, gives the orientation of the tool on the surface. To generate the CL data, the parameter space of the machining surface is discretized along the parameter in which the surface

is *not* linear. The resolution of the discretization is such that the error introduced through the linear motion of the tool between two adjacent iso-parameters of the surface is smaller than the user defined machining tolerance. For side milling of the non-developable ruled surfaces with curvilinear rails, linear approximation error is avoided by increasing the number of CC points along the tool path. For the five axis side milling, increasing the number of cutter locations does not result in memory overflow of the Computer Numerical Control (CNC) controller, because there is usually only one or two tool passes [Liu95].

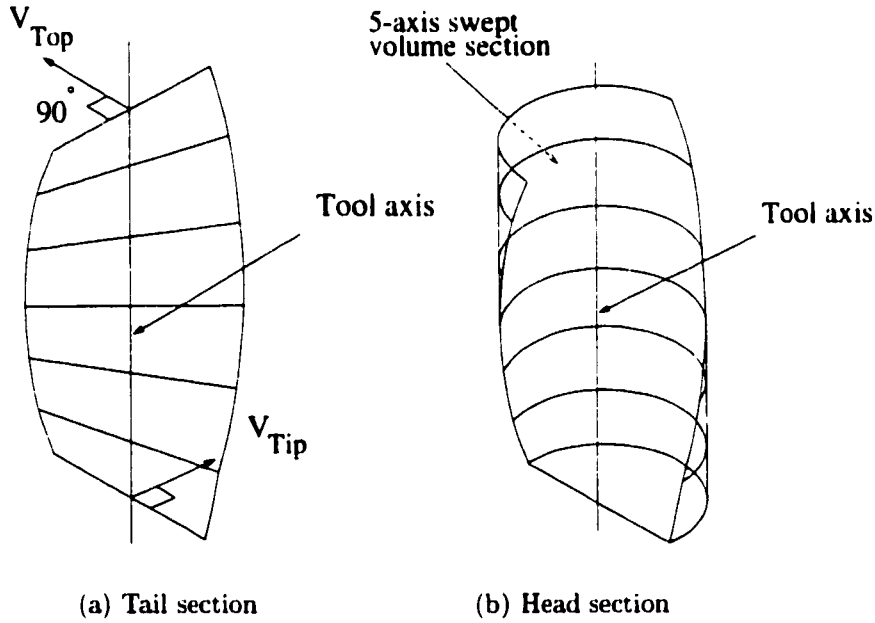


Figure 3.6: Swept volume sections in multi-axis milling

3.5 Multi-axis tool swept volume representation

Multi-axis milling technology is significant for a group of industrial parts (e.g., pump and compressor impellers), that are only machinable using this method. Accordingly, geometrical modeling of the multi-axis tool swept volume becomes an essential part

of any multi-axis milling simulation. A smooth sculptured object whose surface is best described as the trajectory of *cross-section* curves swept along a *profile* curve is called a *sweep object.* The method of constructing smooth surfaces from a sequence of cross-sectional curves (without profile curve) is named *lofting.* In the Computer Aided Graphic Desing (CAGD) community, however, the same lofting is given different names depending on how the interpolating surface is constructed. If the section curves are blended with Hermite function, the resulting surface is called a *Hermite blended cross-sectional surface.* If the section curves are B-Spline curves, and they are blended using B-Spline basis functions, the method is called *skinning.* Geometric simulation of the metal removal process in the milling operation, requires Boolean subtraction of the workpiece solid model with that of the swept tool. The geometric complexity of the tool swept solid (or volume) is mainly dependent on the tool path. While most of the solid based geometric engines provide a built-in function for the sweeping of a section along a 2D path, an advanced skinning algorithm is required for the multi-axis swept volume generation. Moreover, while only four of the faces in a three axis swept volume are non-analytic [IE98], all the bounding faces of a four or five axis swept volume are non-analytic. For a given tool type/geometry/path, the construction of a multi-axis tool swept solid consists of the following steps:

1. For any tool position/orientation given in CL data file, the corresponding sweep section is constructed first. This consists of the following two steps:

    - For each section, a series of half-circle (or line segment) edges are constructed *normal* to the tool axis (see Figures 3.6(a) and 3.6(b).) These edges are also normal to the *local* tool path velocity vector. Radii of circular edges or the length of the linear ones, depend on the tool type/size.
    - Adouble sided solid sheet (both directions outside) is then constructed by the skinning of these edges.

2. A copy of the first and last sections are created/modified as one sided sheets.

3. The sheet sections are skinned to generate a hollow duct of the swept solid.

4. The tail and head sheets are stitched to the duct to generate the solid object.

A schematic of a multi-axis swept solid of a flat end mill is shown in Figure 3.7.

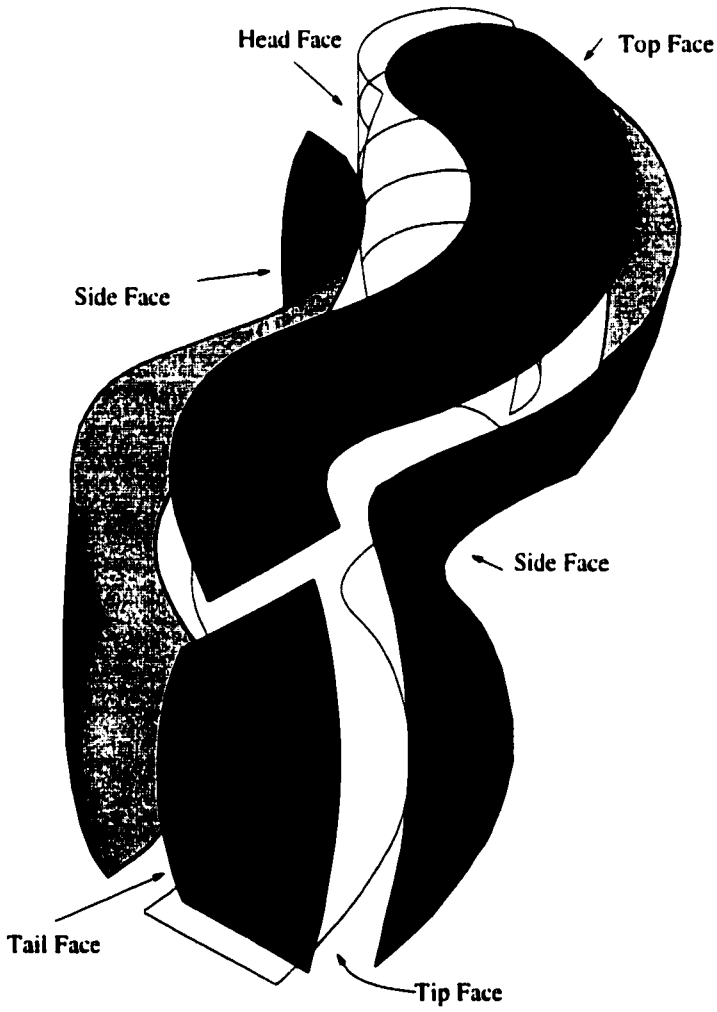


Figure 3.7: A multi-axis flat end mill swept volume and its boundary faces

3.6 In-process workpiece geometry

Finally, the Boolean subtraction of the tool swept solid from that of the workpiece generates the in-process workpiece geometry. A four axis swept volume of a 1 inch

Figure 3.8: 4 axis swept volume of a 1 inch flat end mill.

end mill is shown in Figure 3.8. The Boolean subtraction of the shown tool swept solid model from the workpiece solid model, generates the in-process workpiece geometry, as shown in Figure 3.9.

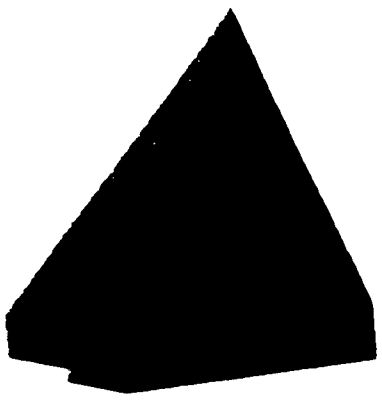


Figure 3.9: 4 axis workpiece cut by one inch flat end mill.

Chapter 4

Automatic mesh generation

4.1 Introduction

The advancing technology of computing hardware and software is well manifested in current CAD systems employing solid modeling. One of the most important downstream applications of solid modeling is to provide a unified geometrical database for all activities in the area of Computer Aided Engineering (CAE), such as finite element analysis, design, and CNC machining. For instance, the finite element method has been proven to be a powerful and versatile analysis tool, but its usefulness is impeded by the need to generate a proper mesh, which is a time consuming and error prone task. In recognition of this problem a large number of methods and algorithms have been developed to automate the mesh generation task. A great majority of these methods and algorithms, however, do not use the CAD database to generate an FEA mesh. Buell and Bush [BB73], Thacker [Tha80] and Ho-Le [HL88] provide a comprehensive review of non-solid model based mesh generation methods.

An essential part of developing a software capable of simulating workpiece flexibility in milling, is to develop an automatic mesh generation module. The mesh

generator must create a valid hexahedral finite element mesh as the in-process workpiece geometry is changing. Moreover, since the input geometry of the workpiece is not known in advance, it should also be able to handle complex shapes.

The remainder of this chapter is organized as follows. First the available 2D and 3D mesh generation techniques and algorithms produced by some research groups and companies for CAD applications are discussed. Next, the available algorithms for the FEA mesh generation on solid models are reviewed, including algorithms developed in General Motors Research Labs [CFF85], Renselaer Polytechnic Institute [YS83], and Sandia National Laboratories [Cha86]. At the end of the chapter, a solid based automatic mesh generation algorithm developed at McMaster University using ACIS solid modeler [Inc95] is introduced. Technical problems and devised solutions encountered during development of the software are also discussed.

4.2 Node First Algorithms

An important group of mesh generation algorithms are those that need nodal points (or sites) first, and generate a mesh by connecting the nodes together. The methods that use this approach require the user to manually divide the boundary, as well as interior of region to be meshed, by providing a set of points. For these algorithms the task of mesh generation, for given planar objects, reduces to determining the most suitable triangulation of a given set of points $P = \{p_1, p_2, ..., p_n\}$.

For a set of given points, Voronoi regions are defined such that: $V(p_i) = \{x : | p_i - x | \leq | p_j - x |, \forall i \neq j\}$. Voronoi regions of a set of points are shown in Figure 4.1. In the theory of computational geometry it is shown that for every plane graph there is another associated graph called *dual graph* that contains the same *information* as the plane graph but their FACES and NODES are interchanged. See Figure 4.2 and Figure 4.3. Delaunary proved that the dual of the Voronoi diagram (like what is

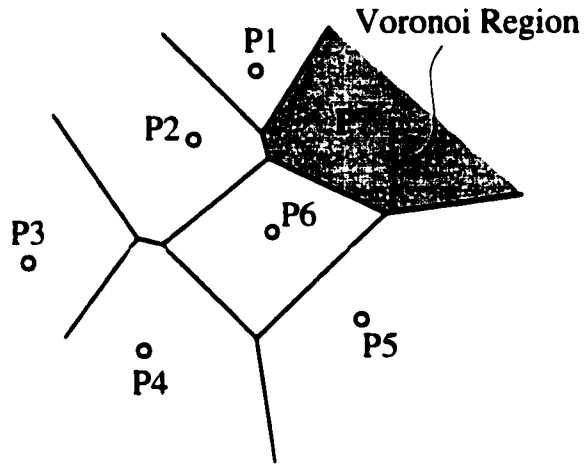


Figure 4.1: Voronoi regions of a set of points

shown in Figure 4.1), when it is drawn with straight edges, (not like the case shown in Figure 4.3) will triangulate the given set of points. In Figure 4.4 a superposition of Voronoi diagram and its Delaunary triangulation is shown. There are well developed algorithms and data structures (such as Quad-edge data structure) for this set of the problems [BS85]. Most of these algorithm are best suit to planar graphs rather than 3D solid objects (solid models). However, because the task of mesh generation on solid models usually can be reduced to triangulation of planar graphs, an understanding of these ideas is required.

4.3 Adapted Mesh Algorithms

Another group of mesh generation algorithms are those that a mesh template is pre-generated, and then adapted to the object being meshed. The main approaches are as follows :

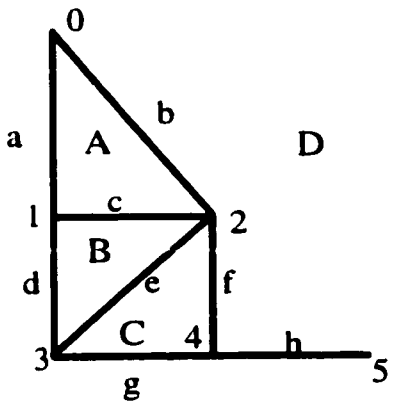


Figure 4.2: A simple plane graph

4.3.1 Conformal Mapping Approach

In this approach a polygon Q that has the same number of vertices as the simply-connected region P to be meshed is constructed such that it can be readily meshed. Then a conformal mapping from polygon Q to polygon P maps the already generated mesh in Q into P. The mathematics involved in this method is extensive and it is mostly useful for 2D problems. Besides in this approach element shape and mesh density is difficult to control [BH82].

4.3.2 Mapped Element Approach

In this approach the object being meshed is required to be manually subdivided into four-sided regions. Although, this approach is not fully automated, it is still the mainstay of many commercial softwares. For any region, using Coons patch terminology [YS83], a normalized unit square in the parametric space (u, v) is mapped onto the region as following

$$p(u, v) = (1 - v)f_1(u) + vf_2(u) + (1 - u)g_1(v) + ug_2(v)$$

where : $0 \leq u \leq 1$ and $0 \leq v \leq 1$.

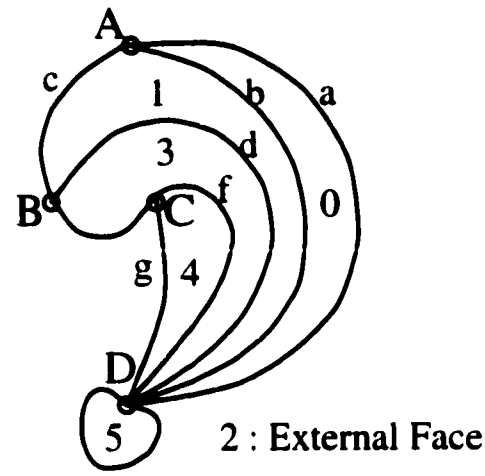


Figure 4.3: Dual graph of Figure 4.2.

In such a transformation from (u, v) space to the sub-region being mapped, the unequally spaced iso-parameter grid lines in (u, v) plane will be mapped into the meshes in the sub-region. When the above formulation is written in terms of projector theory [Gor83] it is called the transfinite mapping technique. Two special cases of transfinite mapping are iso-parametric mapping [ZP71] and discrete transfinite mapping [HSA81]. In iso-parametric mapping the boundary curves of quadrilaterals are described by Lagrange polynomials. In discrete transfinite mapping boundary curves are represented by sets of points located on curve with a unique coordinate associated with each point on the list. Splines are also used in this family of mesh generation technique as in Yildir and Wexler [YW83].

4.4 Grid Based Approach

In this approach a grid of elements (rectangular or triangular) is generated first. The grid is then superimposed on the object. Grid cells that fall outside of the object are discarded. Grid cells that intersect the object boundary are adjusted or trimmed

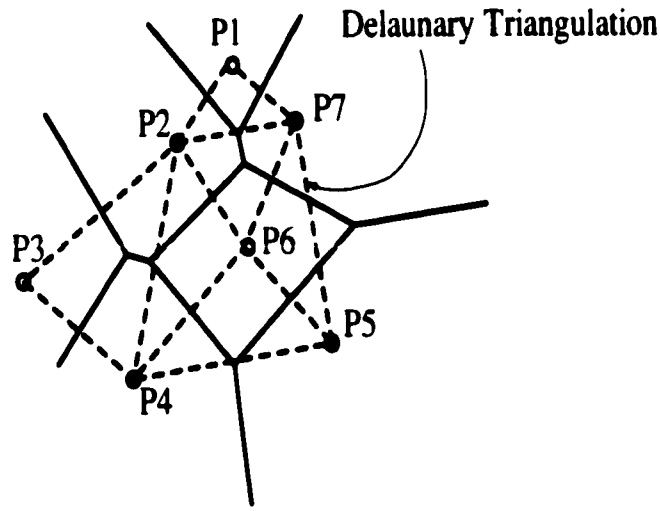


Figure 4.4: Delaunary Triangulation

so that they fit into the object. In this approach, the interior element are excellent, however the elements adjacent to the boundary of the object could be distorted. Yerry and Shephard [YS83] have shown that a quadtree data structure can be modified to be more suitable to grid based mesh generation.

4.5 Formex Algebra

Formex algebra is a mathematical system that provides a basis for solution of problems of data preparation for FEA and graphics in computer aided design process [Noo84]. In this approach a set of basic operations are defined on mathematical objects called *formices*. For example, in a finite element mesh consisting of some pattern of beam elements one element at some certain orientation is selected to be a formex F_1. Then pre-defined operations are used to combine simple formices together to form complex formices. For example, combination of few simple formices using the ***duplus*** operator # leads to a complex formex $F = F_1\#F_2\#F_3\#F_4$. Geometry of the result formex is

more complex than those of any involved formices. Combination of complex formices eventually leads to the pattern required for elements in the mesh generation task.

Regardless of the method of mesh generation used, the following issues are always required:

- conversion of element types

  If meshes are generated for only one type of element, the generated elements can be converted to another type as desired. Figure 4.5 and Figure 4.6 show conversions from brick to tetrahedra and from tetrahedra to bricks, respectively [Hig83].

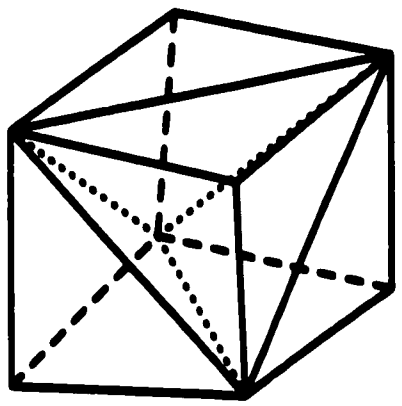


Figure 4.5: Conversion from brick element to tetrahedra

- mesh smoothing

  To improve the mesh quality, sometimes it is necessary to apply a mesh smoothing technique on it. Hermann [Her76] proposes to reposition an internal node i as follows

$$p_i = 1/N(2 - W)\sum_{n=1}^{N}(p_{nj} + p_{nl} - wp_{nk})$$

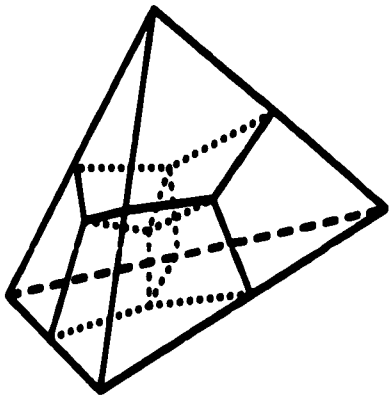


Figure 4.6: Conversion from tetrahedra element to brick

where N is the number of elements around the node i and w is the weighting factor, $0 \leq w \leq 1$. If w=0, it is called *Laplacian* smoothing and if w=1 it is called *iso-parametric* smoothing.

- mesh conformity

  Most of the finite element packages require a mesh to be conforming, where adjacent elements share a whole edge or a whole face. On the other hand, one needs to accommodate changes in element sizes from region to region, and at the same time preserve the conformity of the mesh, see Figure 4.7. If mesh conforming is allowed to be violated, mesh refinement as well as size transition become easier. Otherwise it would be difficult specially for quadrilateral elements, or brick elements in 3D.

4.6 Mesh generation using geometric modelers

Since a finite element mesh communicates the geometry of the domain to be analyzed with the solver, it seems appropriate that it be extracted directly from the geometry of the workpiece. In this case, the problem of mesh generation reduces to conversion

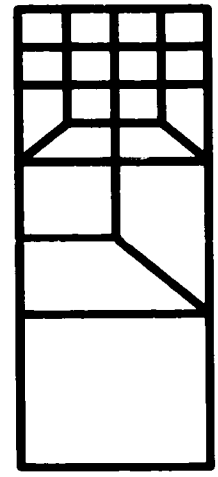


Figure 4.7: Mesh refinement

of the workpiece geometry to a form understood by the finite element solver. Two important approaches are introduced to this problem:

4.6.1 Grid Based Approach

This approach involves dividing up the space enclosed by the object into smaller shapes using the *octree* or *quadtree* data structures [YS84]. The octants are then broken up into tetrahedra by using rather complicated algorithms.

4.6.2 Geometry Decomposition Algorithm

This approach involves cutting the surface of the volume of the given object into sub-surfaces or sub-volumes. The process of subdivision is continued untill acceptably sized finite elements are obtained [VP82]. The technique is based on two laws of analytical geometry:

1. Every polygon is divisible into triangles.

2. Every polyhedron is divisible into tetrahedra.

If one can reduce a planar graph to a polygon, one will be able to create triangles in it. Similarly, if one can reduce a volume to a polyhedron, one can always divide it into some tetrahedra.

- Meshing a Surface

  The first step in this task is to reduce the surface into a polygon. To achieve this, the surface is first projected to a plane. Nodes are then created on the boundary of the curves. Now the objective is to divide the polygon into smaller ones using the best split line. The rules involved in this splitting task are as follows:

  1. The split lines should be created through concave vertices.
  2. The split lines should make angles close to 60 and 120 degrees with respect to the boundary.
  3. The split lines should have the smallest possible length.
  4. A minimum number of nodes should be created on a split line.

  To exclude *invalid* split lines (*Art Gallery Problem* [O'R94]) a graph of the sight angles of all vertices as seen from a given starting vertex P, versus s, i.e. the circumference distance of all vertices from P, is plotted. (See Figure 4.8.) The presence of *dips* in this plot is a sign of the presence of the regions in the polygon with respect to vertex P; which in turn leads to detection of invalid splitting line. The process of dividing polygons into smaller ones is continued untill all remaining polygons have only 3 vertices. For the case of Quadrilateral elements, all polygons must have 4 vertices at the end.

- Meshing a Volume

  Conceptually, the method is the same as meshing surfaces but the calculations

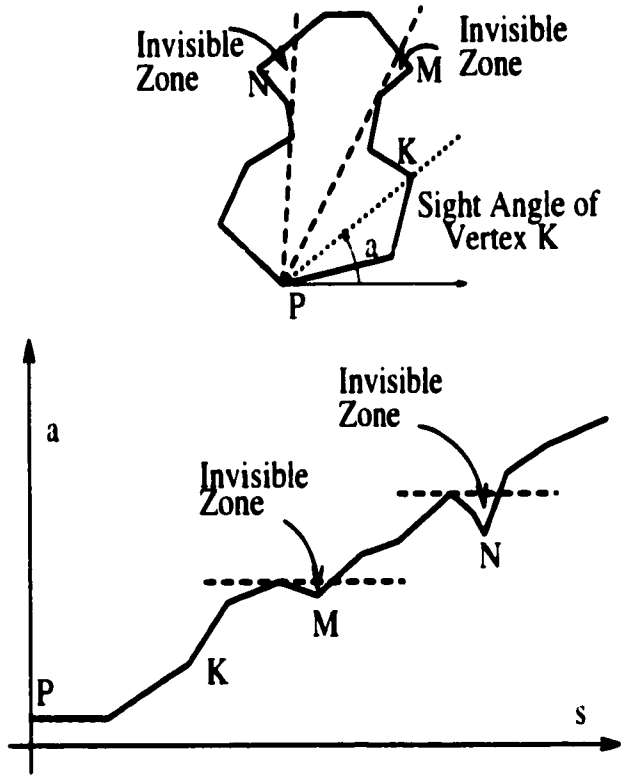


Figure 4.8: Invalid split line detection.

involved are more extensive than before. At the first step, all of the surfaces of the volume are meshed using the procedure described above. This reduces the volume to a *polyhedron*. The meshing is then continued in a fashion similar to surface meshing, i.e. by finding the best splitting the surface for the polyhedron. Except for a starting vertex, non visible sub-volumes and invalid splitting surfaces are found (if any). The fight for the best splitting surface is continued based on the similar rules mentioned in the previous section and used to divide the volume into two sub-volumes. The division process continued for all the sub-volumes till all the sub-volumes created have 4 or 6 faces. The number of nodes generated on any splitting surface is determined based on the

element density required at that region. TRIQUAMESH is the name of a mesh generation program that works based on the above approach and is used by I-DEAS to generate beam, shell and solid finite elements for arbitrarily shaped objects [Amr86].

4.6.3 Mapped Element Approach

At the Sandia National Laboratories, an automatic mesh generation program was developed which is capable of generating hexahedral elements for 3D finite element analysis. In this approach, the solid model to be meshed is decomposed, using the primitives and Boolean operations of the solid modeler into a set of *finite element primitives*. For the simple finite element primitives, a transfinite mapping [HSA+81] is then derived by utilizing the parametric representation of the surfaces of the object to be meshed. In this process, two nodes on two *opposite* faces are located. The interior points (nodes) of the mesh are then generated through a *lofting* of the meshes on these faces. The validity of the generated interior nodes is checked through point classification. For a complex type of primitives, such as a torus, a decomposition technique is used to generate a mesh on them. This decomposition of the finite element primitives into meshes is transparent to the user and automatically done by the solid modeler. However, decomposion of the original part into finite element primitives requires user interaction.

4.7 Automatic brick finite element generation

Although for most 2D mesh generation problems there are solutions proposed in the open literature, the 3D mesh generation is still a challenge. Unfortunately, the generation of hexahedral element meshes is algorithmically much more complex than the generation of tetrahedral element meshes, and very few algorithms have been

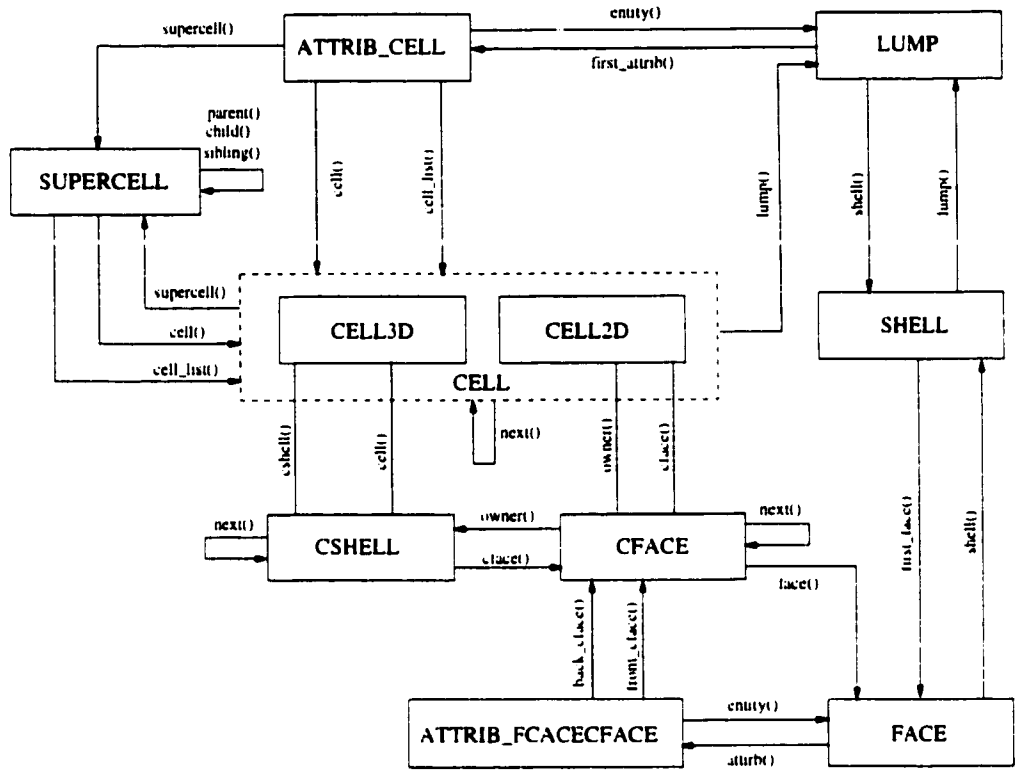


Figure 4.9: Supported data structure by the ACIS Cellular Topology Husk

developed in the past. The following is the methodology proposed for this task and its implementation procedure. First the cellular topology husk of ACIS is introduced and then characteristics of mesh generation algorithms suitable for CAD and CAM application are discussed. At the end, the proposed algorithm and its implementation and other related issues are covered.

4.7.1 ACIS solid modeler - Cellular Topology Husk

The ACIS geometric solid modeler provides an auxiliary data structure through its cellular topology husk. This auxiliary data structure allows the user to organize models with mixed dimensionality. Dimensionality refers to the number of dimensions

occupied by entities within the model. *Sheets* are 2D and *solids* are 3D, and when both 2D and 3D objects occupy the same model, the model has a mixed dimensionality. In general, by using this husk an attribute class; called ATTRIB_CELL, hangs from a lump pointer and contains a list of all cells attributed to that lump [Inc95]. The supported data structure by the cellular topology husk is shown in Figure 4.9. By a proper interrogation of the data structure of a cellularized object, it is possible to access every cell geometry detail, including the cell vertices. These cell vertices, which later will serve as finite element nodes, are used to generate an associativity table for the generated cells (which later will server as finite elements).

4.7.2 CAD versus CAM automatic mesh generators

In recent years some automatic finite element mesh generation algorithms have been developed for CAD systems where geometry of the model is constructed and maintained using mainly solid models. However, very little effort has been devoted to development of an automatic mesh generation algorithm suitable for the CAM environment. In CAD applications, although the geometry of the part is not known for the meshing algorithm in advance, it is not subjected to variation either. This situation is similar to a case in which a mechanical part is meshed for a static or dynamic analysis. When the mesh is generated it is not subjected to a change later on. In CAM applications, on the other hand, not only is the part geometry not known a priori, but, because of the machining or forming process it is subjected to variation in time as well. Moreover, while in CAD applications, a general mesh density control is sufficient for acceptable mesh quality, in a CAM environment a strict mesh quality control mechanism is required to assure generation of the desired number and order of elements at the contact zone of the tool/workpiece system. All of these issues complicate the algorithm of mesh generation for CAM applications.

4.7.3 The algorithm and its implementation

A special purpose solid based automatic mesh generation algorithm is introduced in this section, which allows for the dynamic analysis of the workpiece structure during the peripheral milling operation. To generate the finite element mesh on the in-process workpiece, this code is called by the dynamic multi-axis force routine whenever an appreciable change in the dynamics of the workpiece is deemed to happen. Inputs to the algorithm are as follows:

- A solid model of the updated workpiece in the form of an ACIS SAT file. Each updated workpiece (object) corresponds to the geometry of the workpiece at some point along the tool path. Accordingly, a sequence of these objects contains all the information required for the geometrical simulation of the metal removal process in the milling operation.

- The type of cutter used in machining, i.e. flat, tapered or ball end mill and their associated geometrical data, such as radius, helical angle and so on. These information are required for the grouping of the nodes which are located on the contact face of the tool/workpiece system.

- Cutter location data file. This file contains all information regarding position/orientation of the tool throughout the tool path. Information for the generation of one of the crucial partitioning layers is extracted from this file.

- Meshing quality control parameters:

  1. Number of elements along the tool axis on the contact face.
  2. Number of elements along the arc of contact on the contact face.
  3. Number of elements along the tool path (include item 2).
  4. Number of elements along the thickness of the workpiece.

5. Number of elements between fixture level and tip of the tool.

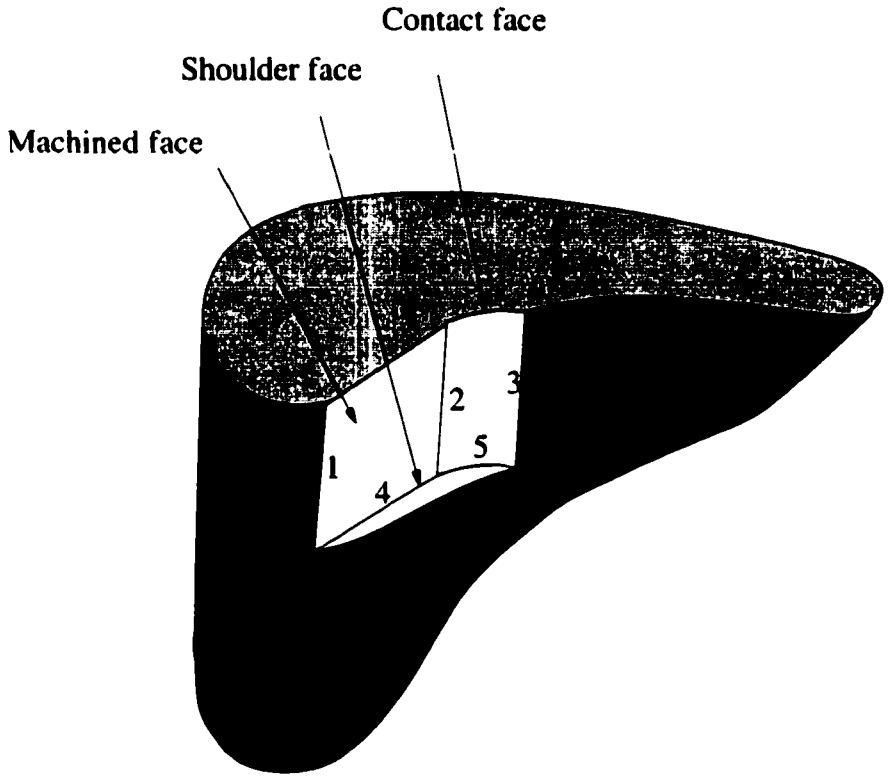


Figure 4.10: Geometry of the updated workpiece

4.7.4 Critical partitioning of the in-process workpiece

Figure 4.10 represents the geometry of the workpiece at sometime after the start of machining. Geometrical simulation of the milling operation always adds three faces to the data structure of the original workpiece. Namely, the added faces are the *machined*, *contact* and *shoulder* faces. Intersection of these faces with the boundary of the workpiece and each other creates five critical curves labeled 1 to 5 in Figure 4.10. The important issue here is to realize that there are always five critical partitioning planes that are passing through these curves, and that the process of cellularization of

the workpiece should start with the partitioning of the workpiece using these critically positioned planes. Orientation of these planes are also dependent on the tool path which is one of the inputs to the algorithm. The critical planes 1 to 3 are generated such that not only they pass through the critical curves 1 to 3 but also such that they are *more or less* normal to the local tool path, see Figure 4.11. The critical

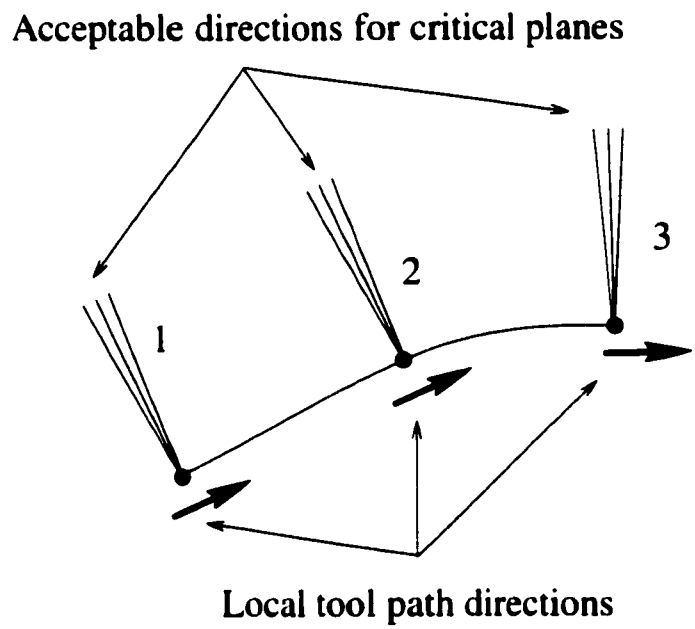


Figure 4.11: Acceptable orientation for the critical planes

planes 4 and 5 both are passing through the curves 4 and 5. One of them, however, is *normal* to the local direction of the tool axis while the other is *parallel* to it. It can be seen that the union of the critical planes shown in Figure 4.12 followed by a Boolean subtraction of it from the workpiece geometry partitions the workpiece into 12 sub-volumes. At this stage, the critical partitioning is done and cellularization of the sub-volumes can proceed as follows. Depending on the mesh quality control parameters specified by the user, a series of copying/translation/rotation operations are performed on the critical planes to generate yet another cellularization structure

for the sub-volumes. For example, if the user wishes to have 4 elements on the contact

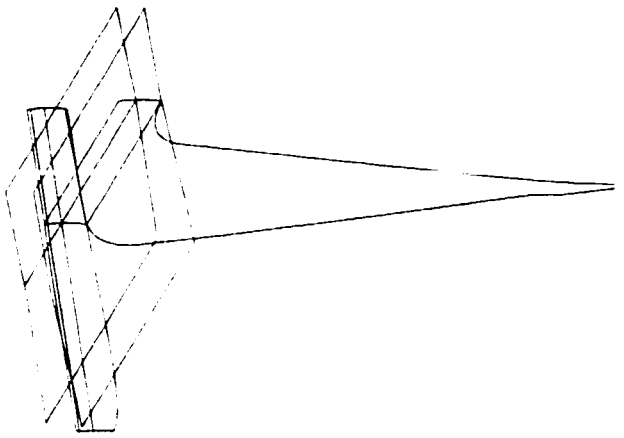


Figure 4.12: Critical planes needed for initial partitioning of the part

face along the tool axis, then 3 copies of the critical plane number 4 will be generated, and translated properly along the vector of the tool axis. For a case that the height of the workpiece is varying along the tool path, a proper *rotation* operation should be added to the above process to ensure a proportionate distribution of elements along the height of the workpiece. Figure 4.13 shows the cellularization structure generated for a case with 2 elements along the tool axis, 2 elements between the fixture level and tip of the tool, 15 elements along the tool path and only 1 element along the workpiece thickness. The high density area is associated with the contact face of the tool where a higher number of elements were desired for that zone. Union of the structure shown in Figure 4.13 followed by a Boolean subtraction from the workpiece geometry cellularize the whole workpiece. The outcome of attaching a cell topology to the cellularized workpiece at this stage will be the generation of a list of 3D cells which collectively exhausts the entire volume of the workpiece and at the same time meets the workpiece precisely at its boundary. If all the geometric and topological computations succeed, each cell should have 6 (or 5) faces and 8 (or 6) vertices and

should point to a neighboring cell in the workpiece. Inherent technical limitations of solid modeling lead to some unexpected situations that will be discussed next.

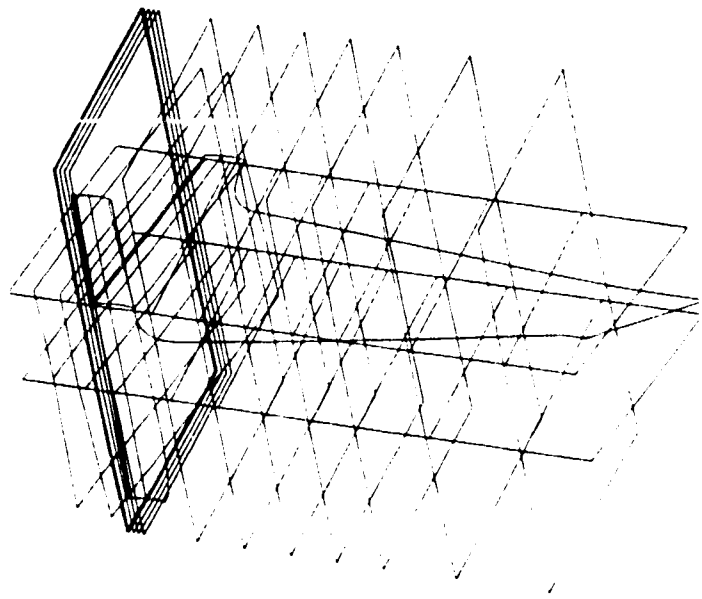


Figure 4.13: Cellularizing planes for a position along the tool path

4.7.5 Solid modeler data structure extension

The developed automatic mesh generation algorithm is linked to the solid modeler by introducing two new classes to its library. The added classes are the **FENODE** and **BRICK** classes, see Figure 4.14. An object of type **FENODE** is a *finite element node* and contains an integer value indicating its global number and a position indicating its location in the model space. An object of type **BRICK** is a *brick finite element* and consists of the following information:

1. An integer indicating the element number

2. An integer indicating the number of nodes used in the element

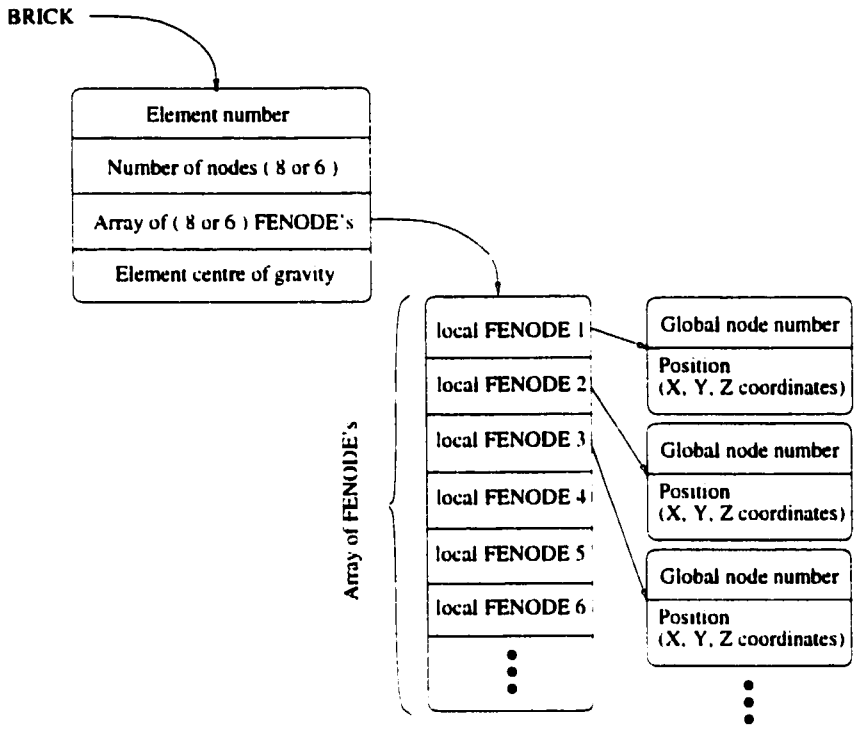


Figure 4.14: FENODE and BRICK classes added to ACIS library

3. An array of **FENODE**s containing all the nodes of the element

4. A position indicating the center of the element

Once the workpiece is cellularized, then for each vertex of a cell a **FENODE** object will be generated. Once all the **FENODE**s are generated for all the vertices, a **BRICK** object will be generated for each cell and initialized using **FENODE** objects generated earlier. As shown in Figure 4.14, the order of appearance of the **FENODE**s in the array of an element, corresponds with their *local* numbering.

4.7.6 Minimization of the bandwidth

By attaching cell data to a partitioned workpiece, first cfaces (c stands for cell) are added to the appropriate sides of all faces. Next, depending on whether faces are

both outside or not, cfaces are aggregated into 2D or 3D cells (cshells). The cshells are computed by finding the nearest radial cface from a seed cface around each edge and contained separation surfaces at each non-manifold vertex and repeating until no more cfaces are found [Inc95]. Since there is no direct control on the selection of seed cfaces for each cell, it happens that the order of positioning of the 3D cells in the *actual object* is different from that of the *list* of 3D cells. This implies that if such a 3D cell list is used to generate **BRICK** finite elements, then the bandwidth of the stiffness matrix generated by the finite element code could be inconveniently large. To overcome this problem, the generated mesh data should be processed for bandwidth minimization. The bandwidth minimization algorithm is basically a sorting routine. This algorithm uses the location of center of gravity of an element to find its location relative to the other elements and the workpiece. At the beginning, all **BRICK** elements are sorted based upon their position along the 'height' of the workpiece. For example, if the number of elements along the height of the workpiece is n, then the procedure would generate n layers of **BRICK** elements after the first sorting operation. During the second sorting operation, elements of each layer, one at a time, are sorted based upon their 'width' and 'thickness' positions from the boundary sides of the workpiece. Eventually, all the elements will be sorted from the smallest height-thickness-width to the largest ones, first in height then in thickness and width. At the end of this stage, finite elements in the 3D cell list would physically follow each other in the model space, i.e. the workpiece. Once the order of elements is corrected, the local numbering of **FENODE**s of each element should be corrected as shown in Figure 4.15. Here again, nodes of each element are divided into two groups. In group one, elements are below and in group two they are above the center of gravity of the element. Nodes of each group are then ordered counter clockwise relative to the projection of the center of gravity on the plane containing the group. Local node 1 (or 5), has always smaller thickness and width location relative to the

projection. Accordingly, the index of each node in the array of **FENODE**s will be its local number in the structure of that element, see Figure 4.14.

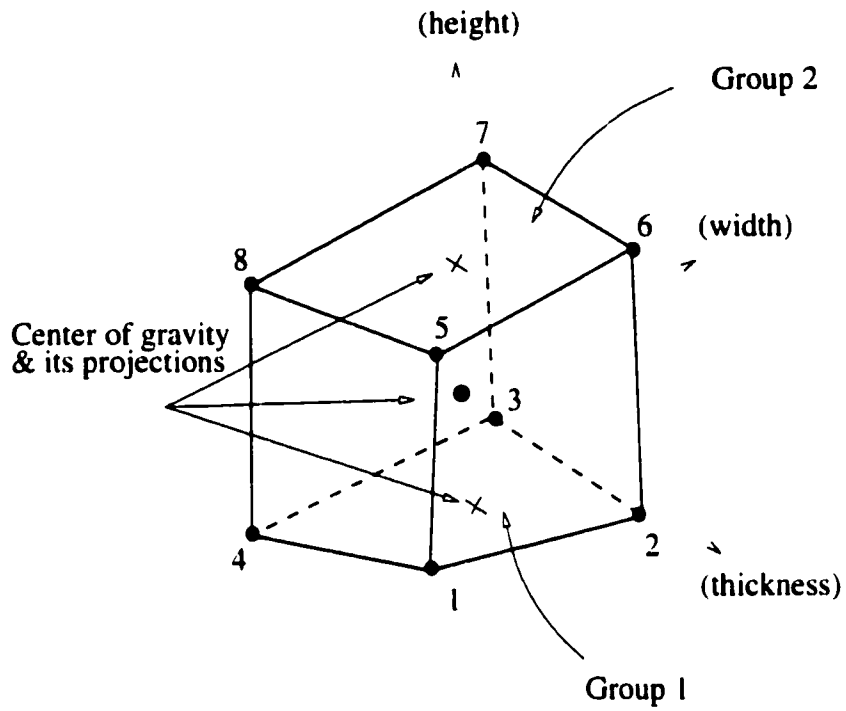


Figure 4.15: Local numbering of nodes of a brick element

4.7.7 Merging coincident global FE nodes

After bandwidth minimization operation, it is required to merge the coincident nodes at the adjacent finite elements. This routine starts from element 1 and sets its global node numbers to be equal to its local ones. It also generates an empty *vertex entity list* and loads it with the vertices of that element. However, for the rest of the **BRICK** elements in the list, the **FENODE**s are checked, one element at a time, versus *all* the vertices in the vertex entity list. Since the consequent elements in the list are also consequent in the workpiece, the vertex equality check should always

start from the *tail* of the list. If an equality happens, that **FENODE** is regarded as a coincident node and next node of the element is examined, otherwise the vertex of the **FENODE** is *added* to the vertex entity list and its global number is set to be equal to its index in the entity list. By the time all the elements in the list are exhausted, global nodes of all the **FENODE**s are set properly and coincident nodes are deleted from the system. Simultaneously the vertex entity list gets richer and richer, and eventually it contains all the distinct (not coincident) nodes generated by the algorithm. At this stage the information required to generate the finite element *input file* is stored in the list of the **BRICK** elements.

4.7.8 Geometry and topological integrity in solid modeling

Most solid modeling systems (including ACIS) are based on B-rep, which follow strict rules to maintain integrity. The key to this integrity is the hierarchy. For instance, edges and points are determined by intersecting faces. This in turn forces a manifoldness, i.e. each edge is the intersection of only two faces, each point is the intersection of only three faces or two edges. Non-manifold properties (which arises in mixed dimensional models) would be: a *cell* wall representing the junction of two internal regions within a solid, or a *scratch* edge on a surface that does not contribute to topology but is needed to specify where a load is to be applied, and for the same reason a *flap* with no internal thickness or volume protruding out from a solid, etc. Although solid modeling is becoming the basis of the majority of CAD software seats, it is still fraught with different kinds of lack of integrity. A few examples are listed here:

- neighboring edges, that are supposed to be joined by sharing a common point, but have separate, duplicate end points.
- an edge with end points that do not lie on its path (i.e edge/point sloppiness)

- an edge whose path intersects with itself
- a face with edges that do not form a close boundary loop
- a face with no finite area
- a face whose embedded geometry folds back on itself
- neighboring faces, that are supposed to be joined by sharing a common edge, but have separate, duplicate edges
- a face with edges that do not lie in its embedded geometry (i.e. face/edge sloppiness)
- spare entities that do not form part of another entity's hierarchy.
- some or all the topological relationships between entities defining a volume is missing

These lack of integrity fall roughly into two categories: geometrical and topological. While geometric defect is related to the issue of positions in the model space, eg. face/edge sloppiness, topological defects are related to the connections or relationships between entities, e.g. 'neighboring' faces not connected to the same edge (i.e. not having reversed coedges in common). To make a solid based automatic mesh generation algorithm *immune* to these problems, some sort of cleaning or filtering routines are to be devised along side of the meshing algorithm.

4.7.9 Filter against geometric and topological insanity

A validity check is first performed on the cell to ensure the object is a valid 3D cell. Extraction of the required cell vertices then starts by creating an empty vertex list for registering valid cell vertices and navigating through all the faces of the cell. In a

healthy topological situation, a valid cell has to have 5 or 6 faces. However, in some circumstances there can be some extra faces, usually with zero area. The first filtering net prevents those faces of entering into the vertex extraction procedure. Moreover, in a healthy geometrical situation, each face should have 3 or 4 edges. However, in some circumstances a valid face can have some extra edges, which will lead to extra invalid vertices for that face. The second filtering net, prevents registering a vertex not located at the corner of a face as a valid one. The algorithm for this filtering procedure is shown in Figure 4.16. First, all vertices of the troubled face are extracted and their center of gravity (**cg**) is found. Next a group of radial vectors (RV) from **cg** to vertices are generated. One of the RV's is picked as the reference and the others are ordered based on the ascending angle to the reference RV. Then, the end positions of three successive RV's are checked for alignment. If they are aligned then the *middle* vertex is *invalid*. If they are not aligned then the middle vertex is *valid*. In the case of detection of a valid vertex, it will be registered in the list of valid vertices only if it is *not* present in the list due to the processing of the other faces of the cell.

4.7.10 Tolerance in surface/surface intersection operation

One of the major issues in the automatic mesh generation based on the cellular topology is the issue of surface/surface intersection operation. Solid models are notorious in failing to perform this kind of operation if the surfaces are tangent or close to tangent to each other, or the intersection is at the boundary of one of the surfaces. In practice it is seen that the tolerance of 1e-6 should be used for the surface interpolation operation when partitioning sheets are generated. Later, the tolerance should be widened to the range of 1e-3 or 1e-2 for successful intersection operations.

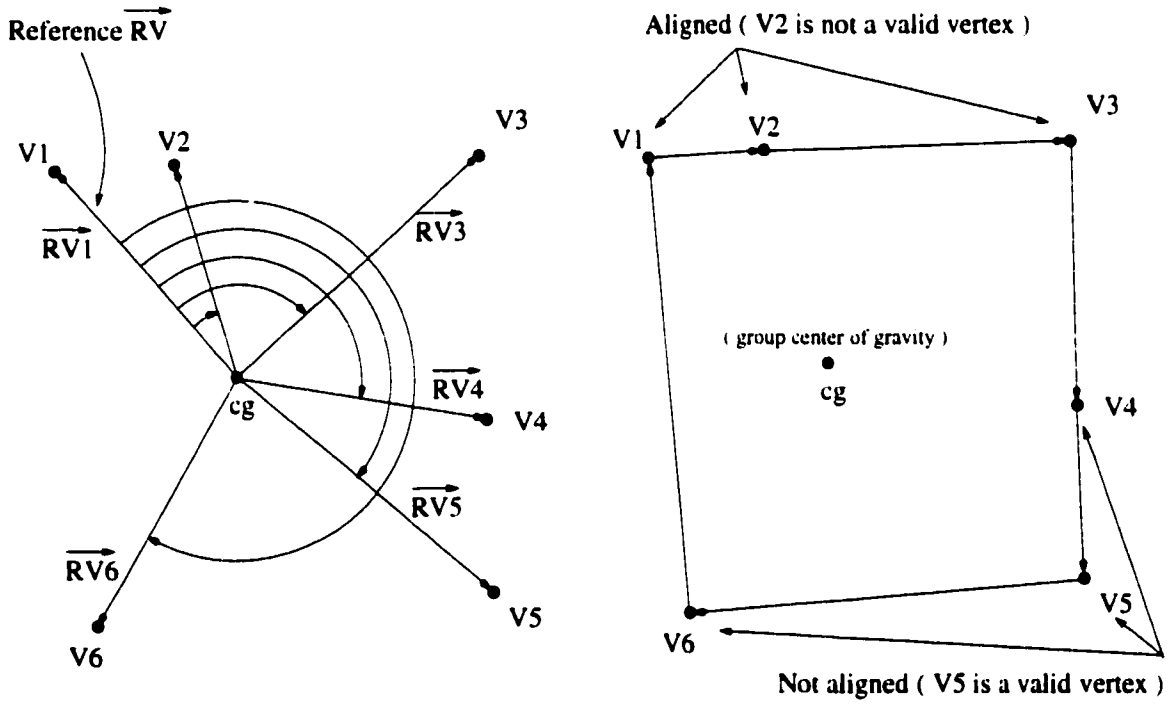


Figure 4.16: Detection of invalid vertices in a group

Chapter 5

Static multi-axis force model

Introduction

Application of five-axis CNC machining centres is increasing remarkably. When a complex shape workpiece is machined with conventional three-axis machining centres, special jigs must be used to face the cutting tool toward the workpiece at arbitrary angles, or the workpiece must be oriented on fixtures several times. A 5-axis machining center, however, makes it possible to machine such workpieces within one setup. Because of the two additional degrees of freedom, 5-axis milling has brought advantages as well as new problems. Among these new problems is modeling of the cutting process for five axis machining. While there have been a number of cutting force models for up to 3-axis milling, the force model for five axis milling has not been fully addressed in the open literature. In general, the mechanistic cutting force coefficients depend on the tool geometry, workpiece material and cutting conditions. Identification for each case requires many cutting tests, after which curve fitting is applied to the collected milling data. Also, an identification procedure based on the experiments alone does not give physical insight. That is, the relation of coefficients to the fundamental parameters, such as the rake and inclination angles, as well as

the friction and shear stress. On the other hand, basic information regarding shear and friction angle and the shear strength of the material provided by the orthogonal data base is general and independent of the type of the tool geometry involved. As a result, analysis of orthogonal cutting can be used to achieve a concrete force model applicable to flat, ball and tapered end milling. In this approach, the oblique cutting condition is modeled by its equivalent orthogonal cutting condition. Variation of the cutting edge geometry and the variation of the normal rake angle along the cutting edge are the two factors that vary from one tool to the other. The use of an orthogonal data base eliminates the need for the special milling tests for each cutter geometry and cutting condition. By modeling cutting forces using an orthogonal data base, however, the process of curve fitting does not happen directly to the measured forces but rather to the very basic parameters of the machining process. The experimentally determined parameters are then used to predict the cutting forces using Merchant's theory [Mer44]. The basic assumptions in using orthogonal data base are as follows:

- The chip flow direction η_c is equal to the inclination angle of the cutting edge i (Stabler's rule.)
- The direction of the friction force on the rake face is coincident with that of the chip flow.

While the first assumption can be modified by adopting a different direction for chip flow than the angle i, the root of the second assumption is in Merchant's analysis and can not be changed or modified without introducing a major change or modification to the whole theory.

5.1 Vectorial parameters in milling simulation

In this chapter a unified force model applicable to all types of end mill cutters

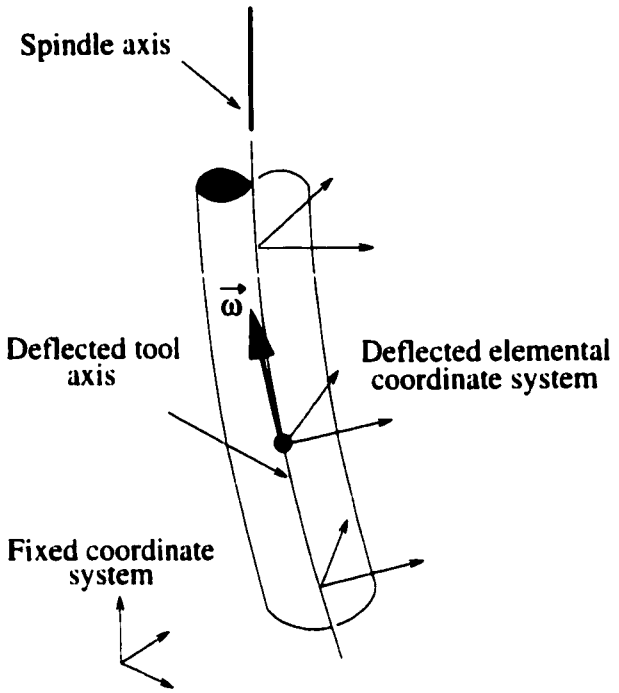


Figure 5.1: Vectorial definition of the spindle speed for a deflected tool.

is formulated. It is shown that in modeling of the multi-axis machining process the feed parameter should be considered as a *vectorial* quantity. In addition, for the case of simple machining processes the proposed formulation reduces to the conventional method used in the past for modeling of milling operation. Moreover, for milling simulation of rigid cutters, spindle speed is usually considered as a scalar quantity. For a statically deflected cutter, however, the spindle speed vector ($\vec{\omega}$) can be conveniently defined as

$$\vec{\omega} = ss \cdot \vec{TA} \tag{5.1}$$

where ss is the spindle speed in rpm and $\vec{TA}$ is the tangent unit vector to the tool deflection curve (pointing toward the tool shank). From the cantilever beam theory

[TM40], the tool axis deflection curve can be expressed by

$$\delta_z = \frac{Fz^3}{6EI} - \frac{Fl^2 z}{2EI} + \frac{Fl^3}{3EI} \tag{5.2}$$

Hence, the cutting force of a statically deflected tool can be computed in two steps. First, the the elemental cutting forces are computed in each deflected coordinate system, (see Figure 5.1.) Next, the elemental cutting forces are transfered to a fixed coordinate system and summed. In this study, only the effect of the feed variation in multi-axis milling is studied. The effect of the cutter static deflection on the milling forces is left for the future work.

5.2 Merchant's cutting force analysis

From Merchant's orthogonal cutting force analysis it is shown that the the power and thrust components can be written as follows [Mer44]:

$$\begin{aligned} F_P &= \frac{\tau \Delta A_c \cos(\beta - \alpha)}{\sin(\phi)\cos(\phi + \beta - \alpha)} \\ F_Q &= \frac{\tau \Delta A_c \sin(\beta - \alpha)}{\sin(\phi)\cos(\phi + \beta - \alpha)} \end{aligned} \tag{5.3}$$

where $\Delta A_c = bt_c$ is the undeformed chip load area. In order to accurately predict the cutting force components using equations 5.3, the parameters ϕ, β and τ must be determined experimentally for the given workpiece material. Figure 5.2 shows geometry of the turning process of a thin walled tube with the wall thickness of a, feed per revolution s and the tool side cutting edge angle of ψ_r. The friction force on the rake face makes the angle η_c with the normal to the edge. Note that the inclination angle i is not shown in this figure. The cutting process of Figure 5.2 can be approximated with the orthogonal cutting of the undeformed chip thickness $t_c = s\cos\psi_r / \cos\eta_c$ with

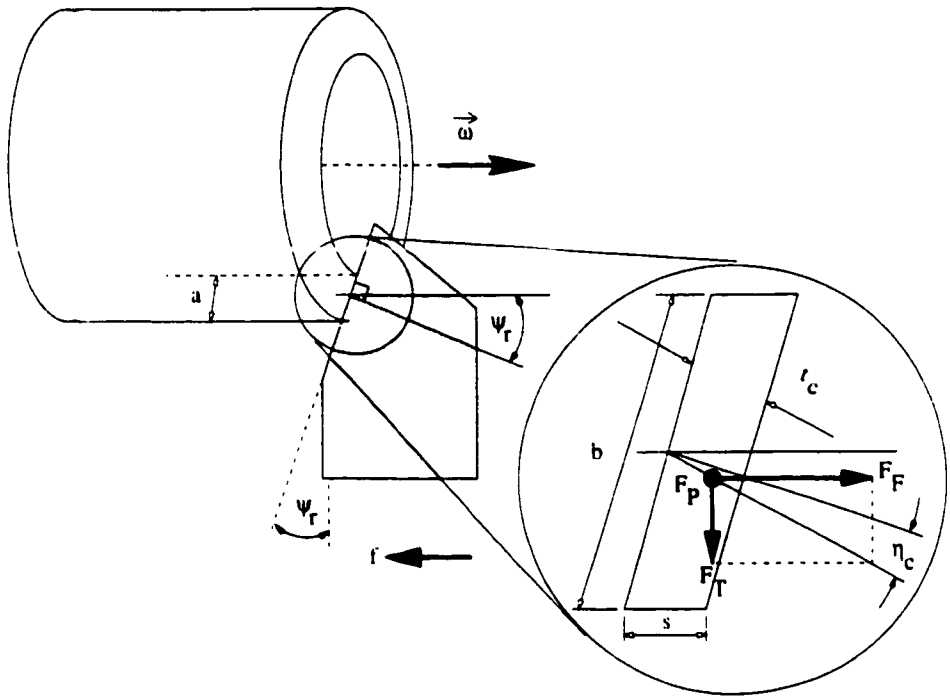


Figure 5.2: Oblique turning of a thin walled tube.

the depth of cut $b = a\cos\eta_c/\cos\psi_r$. The experimentally measurable components of the cutting force (power F_P, feed F_F and thrust F_T components) are related to the machining parameters as follows [NA76]:

$$\begin{aligned} F_P &= as\tau\left[\cot\phi + \tan(\phi+\beta-\alpha)\right] \\ F_F &= F_Q\cos(\psi_r+\eta_c) = as\tau\left[\cot\phi\tan(\phi+\beta-\alpha)-1\right]\cos(\psi_r+\eta_c) \\ F_T &= F_Q\sin(\psi_r+\eta_c) = as\tau\left[\cot\phi\tan(\phi+\beta-\alpha)-1\right]\sin(\psi_r+\eta_c) \end{aligned} \tag{5.4}$$

In equation 5.4 the unknown parameters are ϕ, β, τ and η_c. In the classical metal cutting subject it is shown that the shear angle has the following relation with the

chip thickness ratio and the rake angle:

$$\tan(\phi) = \frac{r_c \cos(\alpha_n)}{1 - r_c \sin(\alpha_n)} \tag{5.5}$$

where r_c is the chip thickness ratio. Having obtained the shear angle ϕ from equation 5.5, all the other unknowns can be determined using equation 5.4. Such a methodology used for steel 1045 leads to the following equations [YP91]

$$\begin{aligned} \phi &= 106.7(V.f)^{0.5} + 0.375\alpha + 13.64 \\ \tau &= 1.586(V.f)^{-0.25} + 67.703 \\ \beta &= 48.4(V.f)^{0.125} + 25.586 - \phi + \alpha. \end{aligned} \tag{5.6}$$

where V is the cutting speed and f is feed per tooth. The machining process of end mill cutters is not orthogonal due to the presence of helix angle i. In the next section the geometry of oblique machining will be discussed and consequently an orthogonal condition will be sought which produces the same amount of cutting force as oblique condition.

5.2.1 Geometry of oblique cutting condition

Shown in Figure 5.3 are all the *systems of coordinates*, *planes* and *angles* of importance in an oblique cutting operation. The geometric relation among the entities in this figure is important in further development of the model. The planes that one may identify in this figure are:

1. Plane containing the cutting velocity and the edge tangent, i.e. V_e
2. Plane containing the tool axis and normal to the cutting velocity, i.e. V_n
3. Plane containing the cutting velocity and normal to the tool axis, i.e. V_h

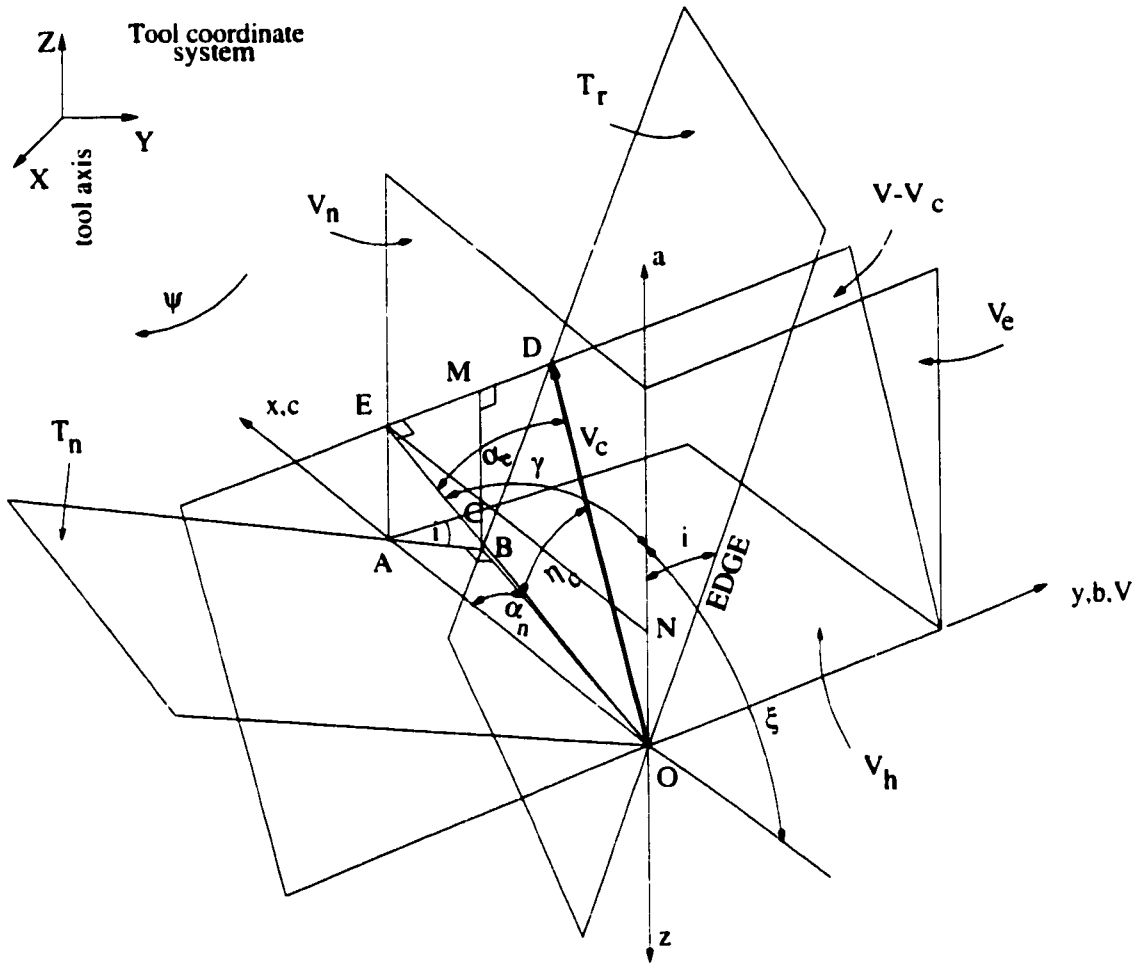


Figure 5.3: Geometric parameters in oblique cutting.

4. Plane containing the cutting velocity and chip velocity, i.e. V-V_c

5. Plane of the rake face, i.e. T_r

6. Plane of normal to the rake face and cutting edge, i.e. T_n

Obviously, the orientation of the item 6 is dependent on that of item 5, however, both are independent of the other planes. Henceforth, depending on the tool type (flat, ball or tapered end mill) and its design (depending the rake angle) items 5 and 6 will

be oriented freely with respect to the other planes. The system of coordinates defined in Figure 5.3 are :

1. The coordinate system *abc*. According to Merchant's theory the orthogonal cutting force developed in the plane V-V_c can be resolved into *power* and *thrust* components. The power component is always along the b axis, whereas depending on the angle γ the thrust component has projections on the a and c axes.

2. The right hand coordinate system XYZ, which is fixed in space, with its Z axis is aligned with the tool axis. For up to three axes machining, the axes X and Y are aligned with those of the machine tool.

3. The right handed coordinate system xyz which is attached to the cutting edge and rotates with the tool. It also rotates from point to point along the cutting edge, keeping its x axis always passing through the axis of the tool. This coordinate system does not have physical significance and is defined as an intermediate coordinate system to transfer the cutting forces from the system of coordinates *abc* to XYZ.

Finally, the angles of interest in Figure 5.3 are:

1. The inclination angle i measured on the plane V_e between the edge tangent and the normal to the cutting velocity.

2. The angle α_n, the normal rake angle measured on plane T_n between the normal to the cutting velocity and the rake face.

3. The angle α_e, the effective rake angle measured on plane V-V_c between the normal to the cutting velocity and the rake face.

4. The angle η_c measured on the plane T_r between V_c and the intersection of the planes T_r and T_n.

5. The angle γ measured on the plane V_n between the planes V-V_c and V_e.

6. The angle ξ measured on the plane V_n between the planes V_e and V_h.

7. The rotational position of reference tooth (angle ψ) measured on the plane V_h between axes x and X.

For the case of a conventional flat end mill the angle ξ is constant and equal to $\pi/2$. Also, the angles i, α_n, α_e as well as the cutting speed V are all constant. For a tapered end mill, however, the angle ξ is constant and equal to $\pi/2 - \Omega$ where Ω is half of the tool conic angle. The angles i, α_n and α_e are also constant for tapered end mills, but the cutting speed V is varying. For the case of ball end mills, all the parameters are varying along the cutting edge.

5.2.2 Orthogonal analysis of oblique cutting condition

As it is seen in Figure 5.3, for any oblique cutting operation there exists the plane V-V_c (containing the chip and cutting velocities) in which the cutting condition is orthogonal. This suggests that the cutting forces generated in any oblique operation can be predicted through orthogonal analysis, if the spatial orientation of the associated V-V_c plane relative to the oblique cutting edge is known [YP91]. Geometrically, to predict the cutting forces generated by a cutting edge with normal rake angle α_n and inclination angle i, the angles γ and ξ are to be determined. In Figure 5.3, in triangle OED : $\sin\alpha_e = ED/OD$. Also using triangles ABC and BMD : $ED = AC + MD = AB\cos i + BD\sin i$. Substitution of the latter equation in the former by noting that $AB = OB\sin\alpha_n = (OD\cos\eta_c).\sin\alpha_n$ and $BD = OD\sin\eta_c$,

yields:

$$\sin\alpha_e = \cos i \cos\eta_c \sin\alpha_n + \sin i \sin\eta_c \tag{5.7}$$

To determine the angle γ, from triangle OAE it is seen that: $\cos\gamma = AE/OE$. From the triangle OED, the relation for OE is: $OE = OD(\cos\alpha_e)$. Also the relation for AE is: $AE = BM - BC$, where $BM = BD\cos i = (OD\sin\eta_c)\cos i$ and $BC = AB\sin i = (OB\sin\alpha_n)\sin i = ((OD\cos\eta_c)\sin\alpha_n)\sin i$. Substituting for BM and BC in the former relation for AE leads to: $AE = OD(\sin\eta_c\cos i - \cos\eta_c\sin\alpha_n\sin i)$. Substitution of relations of AE and OD for $\cos\gamma$ yields:

$$\cos\gamma = \frac{\sin\eta_c\cos i - \cos\eta_c\sin\alpha_n\sin i}{\cos\alpha_e} \tag{5.8}$$

To determine the angle ξ, the cutting edge equation is to be known. Let us denote the functions $X(t)$,$Y(t)$ and $Z(t)$ to be the cutting edge components in the coordinate system of XYZ and t to be some convenient parameter. Then

$$\sin\xi = \frac{dZ}{ds\cos i} \tag{5.9}$$

where $ds = \sqrt{dX^2 + dY^2 + dZ^2}$.

According to the experimental trends in oblique cutting, the variation of parameters τ, β and ϕ due to the variation of angle i is *very small.* It follows that for the cutting edges with the $i < 40.0$ degrees, the orthogonal cutting data base could be used to predict the cutting forces in oblique machining. In this approach the concept is to use the angles denoted by γ and ξ in Figure 5.3 to write the effective chip load area. To illustrate this formulation, Figure 5.4 shows a point P on the cutting edge of a tapered end mill with three auxiliary views. The effective chip load area (defined for the orthogonal cutting condition happening in the plane V-V_c) can be defined

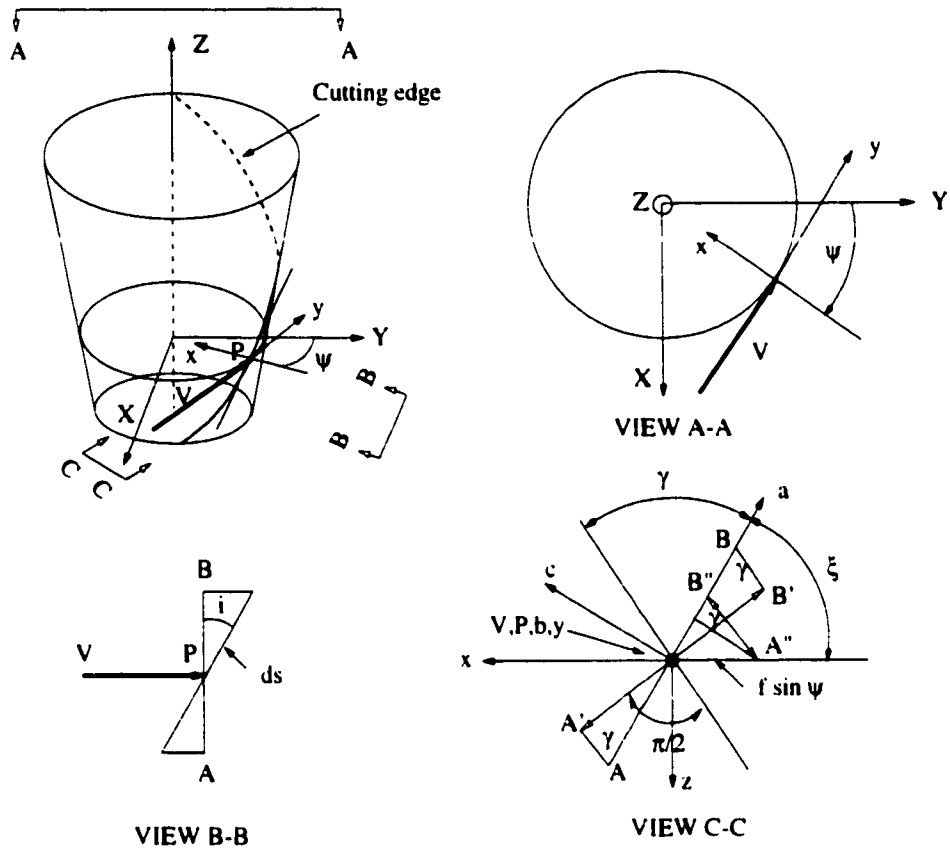


Figure 5.4: Equivalent orthogonal chip load area.

using the VIEW C-C of the Figure 5.4. Accordingly, $\Delta A_c = \overline{A''B''}.\overline{A'B'}$ where ΔA_c is the orthogonal chip load area, $\overline{A'B'}$ is the orthogonal depth of cut and $\overline{A''B''}$ is the orthogonal undeformed chip thickness. From the VIEW C-C, $\overline{A'B'} = ds \cos i \sin \gamma$ and $\overline{A''B''} = f \sin \psi \sin \xi / \sin \gamma$. Substituting for $\overline{A'B'}$ and $\overline{A''B''}$ in relation for ΔA_c we get:

$$\Delta A_c = ds f \cos i \sin \psi \sin \xi \tag{5.10}$$

Substituting equation 5.10 for ΔA_c in equations 5.3, the power and thrust components

can be predicted as follows [YP91]:

$$\begin{aligned} F_P &= \frac{\tau \Delta A_c \cos(\beta - \alpha_e)}{\sin(\phi)\cos(\phi + \beta - \alpha_e)} \\ F_Q &= \frac{\tau \Delta A_c \sin(\beta - \alpha_e)}{\sin(\phi)\cos(\phi + \beta - \alpha_e)} \end{aligned} \tag{5.11}$$

Note that since the relation is written for V-V_c plane, the rake angle α_n in equations 5.3 is to be changed into α_e. The power component is along the b axis, whereas the thrust component has two components along the a and c axes. Accordingly [YP91],

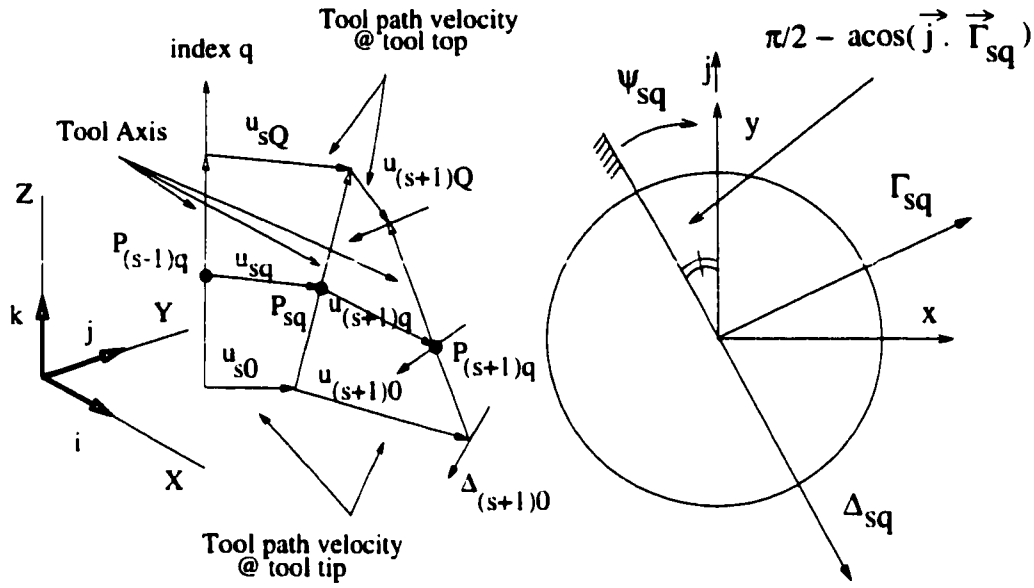
$$\begin{aligned} \Delta F_a &= F_Q \cos\gamma \\ \Delta F_b &= F_P \\ \Delta F_c &= F_Q \sin\gamma \end{aligned} \tag{5.12}$$

Referring to the definition of angle ξ in Figure 5.3, force components in xyz components will be [YP91]:

$$\begin{aligned} \Delta F_x &= \Delta F_a \cos\xi + \Delta F_c \sin\xi \\ \Delta F_y &= \Delta F_b \\ \Delta F_z &= -\Delta F_a \sin\xi - \Delta F_c \cos\xi \end{aligned} \tag{5.13}$$

And finally, the cutting force components in the fixed tool coordinate system can be written as follows:

$$\begin{aligned} \Delta F_X &= -\Delta F_x \cos\psi - \Delta F_y \sin\psi \\ \Delta F_Y &= -\Delta F_x \sin\psi + \Delta F_y \cos\psi \\ \Delta F_Z &= -\Delta F_z \end{aligned} \tag{5.14}$$



(a) Feed variation along tool axis (b) Shift angle in multi-axis milling

Figure 5.5: Variation of the feed rate and shift angle in multi-axis milling

5.3 Extension to a static multi-axis force model

For a machining process in which the axes of the workpiece coordinate system (WCS) remain in parallel with those of machine coordinate system (MCS) the solution for cutting forces is complete by equation 5.14. However, for the case of multi-axis machining, the orientation of WCS is changing relative to the MCS. Hence, the transformation given in equation 5.14 are the cutting force components represented in the tool coordinate system (TCS). Also, in four and five axis machining, the tool path velocity and accordingly the feed rate are varying along the tool axis. As a result, a shift angle is introduced to the ψ angle of the each tool element. Shown in Figure 5.5(a), is the variation of the tool path velocity along the tool axis at three sections

along some arbitrary five axis tool path. Each section can be considered as the spatial orientation of the tool after execution of an NC block. In Figure 5.5(b), the same variation of the feed vector is shown in the tool coordinate system (TCS). For the case of a rigid end mill the vector of the tool axis is always normalized by the *tool overhang length*, i.e. $\vec{TA}$ is a unit vector along tool axis. If the cutter is deflected then unit tangent vector to the curve of deflection should be used for $\vec{TA}$. In addition, for the case of four and five axis machining the feed parameter is usually specified by the *inverse* of the time period required from one tool orientation s to the next $s+1$, denoted by T_s. The machine tool controller moves the five axes from the orientation s such that by the end of the specified time period all the five axes are oriented according to the new orientation $s+1$. By knowing the tool orientation at two successive sections, i.e. s and $s+1$, and the time required for the execution of the associated NC block, it is possible to write for the tool path velocity of tool axial element q

$$\vec{u_{sq}} = (P_{(s+1)q} - P_{sq})/T_s \tag{5.15}$$

where $P_{(s+1)q}$ and P_{sq} are two corresponding positions of element q at sections s and $s+1$. By knowing the tool path velocity *vector*, i.e. its magnitude and direction, and the tool axis vector (either deflected or not) it is possible to find the local direction in which the five axis swept volume is bounded

$$\vec{\Delta}_{sq} = \vec{u}_{sq} \times \vec{TA} \tag{5.16}$$

If the local radius of the cutter is denoted by R, then by moving away normal to the tool axis along the $\pm R \cdot \vec{\Delta}_{sq}$ directions one reaches two points *on* the swept volume. These points are on the intersection of the side walls of the swept volume and its front face. For any direction other than $\pm R \cdot \vec{\Delta}_{sq}$ one of the points will be *in* the swept

volume. Furthermore, the local feed direction of element q at NC block s is given by

$$\vec{\Gamma}_{sq} = \vec{TA} \times \vec{\Delta}_{sq} \tag{5.17}$$

Magnitude of the feed along the feed direction $\vec{\Gamma}_{sq}$ can be determined using

$$v_{sq} = \vec{u}_{sq} \cdot \vec{\Gamma}_{sq} \tag{5.18}$$

which leads to the following relation for the feed per tooth along $\vec{\Gamma}_{sq}$ direction

$$f_{sq} = \frac{60 \cdot v_{sq}}{N_z \cdot ss} \tag{5.19}$$

where N_z is the number of teeth and ss is the spindle speed in rpm. Variation of the tool path velocity along the tool axis also produces a shift angle ψ^q_{sh} in the datum of the ψ angle (see Figure 5.5(b)). Depending on the motion of the forth and fifth axes the shift angle can be positive or negative

$$\psi^q_{sh} = \pi/2 - \cos^{-1}(\vec{j} \cdot \vec{\Gamma_{sq}}) \tag{5.20}$$

The rotational position of the tool axial element q at the NC block s can be determined using

$$\psi_{sq} = \psi_{BEF} - \delta\psi + \psi^q_{sh} \tag{5.21}$$

where ψ_{BEF} is the rotational position of the Bottom of the Engaged Flute (BEF) (measured from $+y$ direction in TCS), $\delta\psi$ is the variation of the rotational position of element q due to the helix angle β (for flat end mills $\delta\psi = q \cdot dZ \cdot \tan(\beta)/R$), and the third term is the shift angle in the datum of the ψ angle due to the variation of the tool path along the tool axis. Accordingly, the chip thickness can be determined

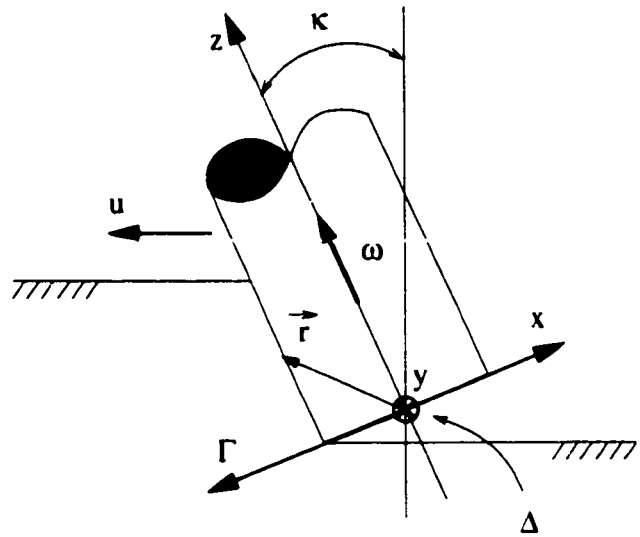


Figure 5.6: Milling with an inclined end mill

using $t_{sq} = f_{sq}\sin(\psi_{sq})$. Substituting the chip thickness t_{sq} in equation 5.10, then the equation 5.14 gives the cutting force associated with the element q in $\vec{\Delta}_{sq} - \vec{\Gamma}_{sq}$ coordinate system. The cutting force components seen in TCS can then be computed through

$$\begin{aligned} F_X^{sq} &= F_\Gamma^{sq}\cos(\psi_{sh}^q) - F_\Delta^{sq}\sin(\psi_{sh}^q) \\ F_Y^{sq} &= F_\Gamma^{sq}\sin(\psi_{sh}^q) + F_\Delta^{sq}\cos(\psi_{sh}^q) \\ F_Z^{sq} &= F_Z^{sq} \end{aligned} \tag{5.22}$$

which accounts for the shift angle for the axial element q. In equations 5.15 to 5.22 the index q varies from zero to Q where

$$Q = \frac{Tool\ Overhang\ Length}{dZ} - 1$$

where dZ is the thickness of axial elements along the tool axis. For the case of an undeflected tool or a simple tool path where feed parameter is constant along the tool axis, the vectorial process model reduces to a simple one (with only *scalar* parameters involved.) For example, in the special case of milling with an inclined mill, shown in Figure 5.6, $\vec{\Delta}$ reduces to $-\vec{j}$, $\vec{\Gamma}$ reduces to $-\vec{i}$, v reduces to $|\vec{u}|\cos(\kappa)$ and finally the relation for the rotational position of a point on the cutting edge reduces to $\psi = \psi_{BEF} - \delta\psi$ which is a function of only the tool rotational position and tool helix angle.

5.4 Chip flow direction

In the first detailed study of the oblique geometry by Stabler [Sta64], it was assumed that the chip flow direction η_c is equal to the angle of obliquity i. Stabler's rule assumes that the chip moves parallel to the cutting velocity vector, independent of the rake angle and friction. However, later experimental observations showed that there exists a dependency between the chip flow direction and other parameters such as friction and rake angle. Brown [RB66] suggested the relation $\tan\eta_c = \tan i \cos\alpha_n$ to account for the rake angle. Zorev [Zor66] considered the effect of the cutting speed on the η_c and suggested the relation $\eta_c = i/V^{0.08}$ where V is in m/min. Among the analyses of chip flow angle is that of Whitfield [Whi86] in which the shear force and shear velocity directions are assumed to be the same. The formulation for η_c in this approach is as follows:

$$A\sin\eta_c - B\cos\eta_c - C\sin\eta_c\cos\eta_c + D\cos^2\eta_c = E \tag{5.23}$$

where

$$
\begin{aligned}
A &= r\cos\alpha_n + \cos i \tan\beta \\
B &= \tan\beta \sin\alpha_n \sin i \\
C &= r\sin\alpha_n \tan\beta \\
D &= r\tan\beta \tan i \\
E &= \sin i \cos\alpha_n
\end{aligned}
$$

Finally, for an oblique cutting condition with a continuous chip formation, Rubenstein [Rub90] suggested $\tan(\eta_c) = f_2/f_1$ where f_1 is the cutting force component parallel to the rake face in the plane normal to the cut surface and cutting edge (plane T_n in Figure 5.3), and f_2 is the cutting force component parallel to the cutting edge direction. In this study, however, Stabler's rule for the chip flow direction is be used, and the effect of other approaches is left for the future research.

Chapter 6

Dynamic multi-axis force model and surface profile

The prediction of cutting forces in milling is a key factor in determining dimensional surface accuracy. Increasing emphasis on the quality of the finished surface in five axis machining, requires accurate models to account for the interaction between tool and workpiece systems. In particular, for a side milling operation on ruled surfaces such as turbine or compressor blades, there is a pressing need for computer-based simulation tools to predict the three-dimensional texture of the machined surface. A computer simulation capable of predicting cutting forces and generated machined surfaces, is a valuable tool for process planners to increase the productivity. The offline simulation of the cutting process can replace the trial and error methods or conservative cutting conditions.

Depending on the workpiece geometry and cutting condition used, unwanted system vibrations may occur which result in poor surface quality. In order to predict the cutter vibrations, either forced or self-excited, a dynamic force model is developed in this chapter. The model accounts for both the dynamics of the cutter and the workpiece structures. Moreover, by monitoring the effective deflections of

the tool/workpiece system at the Surface Generation Point (SGP), the surface profile/error are predicted. It is seen that, in case of thin-walled workpieces, the surface generation is mainly affected by the workpiece deflections and to a lesser extent by the cutter deflection. Modeling of the workpiece dynamics is discussed at the end of this chapter.

6.1 Time Domain simulation of dynamic end milling

In the previous chapter a unified multi-axis rigid force model was introduced in which depending on the motion of 4th and 5th axes, the chip thickness could be varying along the tool axis. In this chapter, the method of chip load regeneration introduced in [TI81] is extended for the multi-axis machining. The regenerative force model argues that the instantaneous cutting force on any given tooth in cut not only depends on the nominal feed per tooth (static component), but also on the current deflections in the tool/workpiece system (dynamic component) as well as the surface waviness left from cutting operation of the previous teeth (undulation component). Similar to many previous research works, the dynamic deflection of the thin elemental slice of the cutter is modeled using a mass-damper-spring system shown in Figure 2.3. The following system of two second order uncoupled differential equations describes the motion of the tip of the tool in x and y directions .

$$M_2\ddot{u}_2 + C_2\dot{u}_2 + K_2u_2 + K_1(u_2 - u_1) + C_1(\dot{u}_2 - \dot{u}_1) = 0 \tag{6.1}$$

$$M_1\ddot{u}_1 + C_1(\dot{u}_1 - \dot{u}_2) + K_1(u_1 - u_2) = F \tag{6.2}$$

Where u stands for both x and y directions, F is the component of the cutting force and M, C and K's are the mechanical parameters along the u direction. The mechanical parameters shown in table 6.1 are determined experimentally for the tip of

MODE	f_n (Hz)	Damping Ratio	Stiffness (N/mm)
mode 1 x	331	0.051	7.52e6
mode 2 x	437	0.031	7.92e6
mode 1 y	456	0.018	9.5e8
mode 2 y	553	0.039	1.12e7

Table 6.1: Mechanical parameters of the tool dynamic model.

the tool [AES98]. The dynamic model implies that the effect of inertia and damping properties of tool and spindle are considered in the modeling of system deflections.

6.1.1 Regenerative force model

If the cutting force components computed in equations 5.22 are applied to the system shown in Figure 2.3, due to the presence of a relative vibration between the tool and workpiece a wavy surface will be left on the machined surface. As the next tooth cuts into the undulated surface of the workpiece, the static component of the chip load will be modulated by the current deflections and waviness of the surface. If the variations in the modulated chip thickness result in a low amplitude vibration then the cutting condition is considered stable. On the other hand, if the chip thickness variation is left undisturbed for some period of time, then it can excite one of the modes of the system and generate a high amplitude vibration known as chatter. Due to the nonlinear characteristic of chatter, the chatter vibrations also stabilize at some high amplitude. In the case of non-closed form models where there is no direct solution for the integration of the chip-load area along the cutting edge, the cutter is always divided into incremental disk like sections. The angular position of the cutter itself is defined by the angular position at the tip of some reference tooth, denoted by ψ_{0j0}. For the case of static simulation, in one full rotation of the cutter, the index j varies between zero and some maximum number J_{max}. This maximum number is selected

such that a monotonic variation in the simulated force pattern is achieved. However, in the case of dynamic simulation, J_{max} should be large enough to ensure that the high frequency-low amplitude components of the force signal are correctly simulated.

Relative to the reference tooth, the angular position of the elemental slice i, on the tooth number k when the cutter is at the jth rotational position, is determined by

$$\psi_{ijk} = \psi_{0j0} + 2k\pi/N_z - \frac{i+1}{2} dZ \tan(\beta)/R \tag{6.3}$$

where dZ is the tool element thickness, β is the cutter helix angle, N_z is the number of teeth and R is the cutter radius, (R_i for a tapered end mill).

From the equation 5.10, it is shown that the chip load area for a rigid force model is $\Delta A_c = dsf \cos i \sin w \sin \xi$ in which the quantity $f \sin(w)$ is the projection of the local feed per tooth in the direction of chip load. To account for the instantaneous dynamics in the system and the surface undulations, the chip thickness can be modified as following [TI81]

$$t_{c(n)} = f \sin(\psi_{ijk}) + x_{(n-1)} \cos(\psi_{ijk}) + y_{(n-1)} \sin(\psi_{ijk}) + MAX(u_1, u_2, u_3) \tag{6.4}$$

where $t_{c(n)}$ is the effective dynamic chip thickness, f is the nominal feed per tooth, $x_{(n-1)}$ and $y_{(n-1)}$ are the instantaneous deflections in the *system* at time step $n-1$ and u_m, $(m = 1, 2, 3)$ are the possible undulations that could be left behind from the previous teeth, (see Figure 6.1). The first term in equation 6.4 is the static component of the chip load, the second and third terms are the effect of the current deflections on the chip thickness and the last term accounts for the undulations left on the work

surface. The relevant relation for u_m is as follows

$$u_m = m \times f \sin(\psi_{ijk}) + x_{(n-mN)} \sin(\psi_{ijk}) + y_{(n-mN)} \cos(\psi_{ijk}) \tag{6.5}$$

where index $m = 1, 2, 3$ and $N = J_{max}/N_z$ is the number of rotational positions for one *tooth* period. As it is shown in Chapter 5, any rotational motion of axes a and b in multi-axis machining, introduces a shift angle to the coordinates system associated with each cutter element. Accordingly, ψ_{ijk} in equations 6.4 and 6.5 should be modified to account for the effect of rotations in axes a and b as follows

$$\psi_{ijk} = \psi_{0j0} + 2k\pi/N_z - \frac{i+1}{2} dZ \tan(\beta)/R + \psi^t_{sh} \tag{6.6}$$

where ψ^t_{sh} is given in equation 5.20. The basic non-linearity can be implemented by the following condition

IF $t_{c(n)} < 0.0$ THEN $t_{c(n)} = 0.0$

The non-linearity condition accounts for the momentary loss of contact between the cutter and workpiece, due to the high amplitude vibrations in the system. Having the chip load computed from 6.4 it is possible to write for the tangential and radial force components using equation ??. The cutting force components computed from equation 5.22 can be directly used for the dynamic response of the cutter (see Appendix C). However, forces computed in the tool coordinate system should be negated first and then transfered back into the workpiece coordinate system (WCS) for the dynamic response of the workpiece structure. Finally, in addition to cutting forces, inertial forces are also applied to the workpiece. The inertial forces are generated due to the accelerated motion of the workpiece coordinate system. In general, these accelerations are applied to the workpiece through the supporting fixture and can be

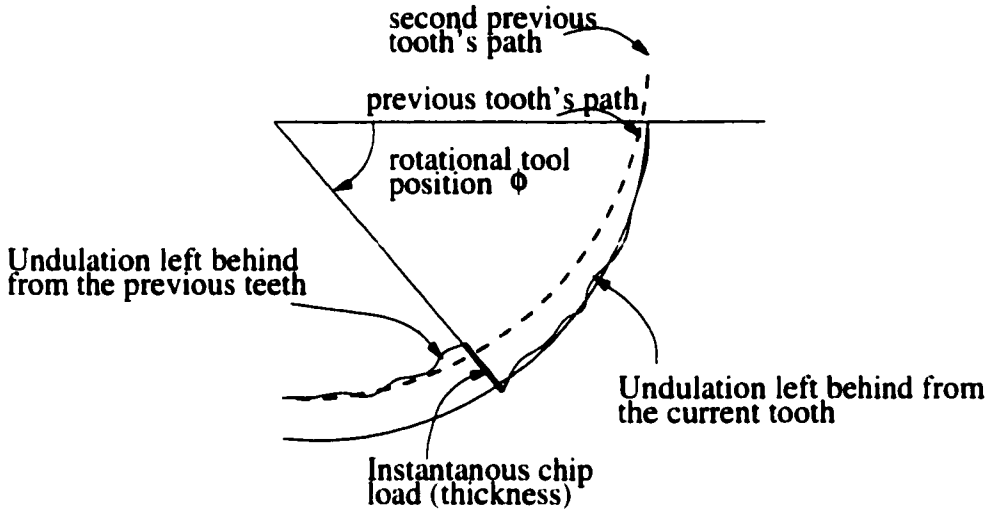


Figure 6.1: Graphic representation of the regenerative chip thickness.

modeled, accordingly, by applying ground acceleration to the clamped nodes in the finite element mesh. However, in this study because the part is assumed to be thin and flexible the effect of inertial forces is neglected.

6.1.2 Rake angles in dynamic milling model

The normal and effective rake angles α_n and α_e are defined for rather a static condition. For the case that the tool and workpiece are vibrating then the direction of the approaching material relative to the tool (specially when the cutting speed is low) can be altered. This change in the direction of the cutting speed can change the effective rake angle, the chip thickness and the process damping. In this study, however, the static definitions of the rake angles are used, and modification of the force model for the dynamic rake angles is left for future research work.

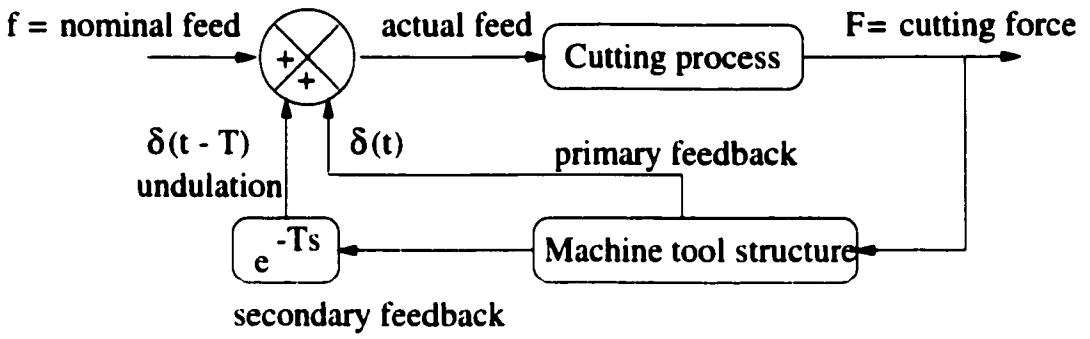


Figure 6.2: Block diagram of the regenerative cutting force model.

6.2 Stability limit in milling

6.2.1 Machining stability

Due to the inherent feedback existing between the cutting forces and the structural deflections (see Figure 6.2), there are conditions under which the milling system becomes unstable. When this occurs, a condition of self-excited *chatter* exists. Chatter affects adversely both the surface finish and the dimensional accuracy of the workpiece. Furthermore, it promotes tool wear and may cause tool breakage and damage to the machine tool itself. In order to avoid chatter, conservative metal removal rates are usually used, which limits the productivity of the machine tools. In the theory of self-excited chatter for single point tools wherein the tool is continuously in contact with the workpiece, the dynamic system can be described by a time invariant characteristic equation. This case has a rather fully developed formulation and solutions. Specifically, Tlusty [KT67] derived a fundamental stability theory in orthogonal cutting where chatter safe axial depth of cut is shown to be inversely proportional to the cutting coefficients and real part of the transfer function between the tool and the workpiece. Tobias [Tob65] also introduced a similar stability law and added process stiffness and damping terms produced by the dynamic motion of the tool during cutting. The analytical prediction of stability lobes by Tlusty and Tobias are mostly

applicable to turning like operations where the direction of the cutting forces and chip thickness generation are time invariant. Knowing the difficulty in the realistic application of the basic stability theory to the milling process, Tlusty et al [TI81] presented a method of generating stability lobes using time domain simulations of the chatter vibrations in milling. In their work, the velocity dependent process damping and the basic nonlinearity in chatter which arises when the tool jumps out the cut are introduced. Although time domain simulations are quite powerful in predicting chatter, an accurate analytical solution is still desirable for rapid and easy prediction of chatter free cutting conditions. For the case of ball end milling, the varying cutting characteristics along the cutting edge, as well as the interdependency of axial and radial depths of cut result in the stability analysis being even more complicated . The concept of stability lobes formulated originally for flat end mills by Budak et.al. [BA95] is extended for ball end mills by Abrari et.al. [AES98]. Due to interdependency of axial and radial depths of cut in ball end mills and due to different cutting characteristics of the cutting edge at different heights of the ball nose, the stability lobes are extended to stability surfaces in [AES98].

6.2.2 Analytical stability analysis

In 1964, Merritt [Mer65] recognized the machine tool-cutting process as a feedback loop which enabled him to apply classical control techniques in the formulation of the problem. Merritt's formulation eventually led to a characteristic equation in the form of a differential-difference equation with *constant* coefficients. Accordingly, it was possible to apply Nyquist stability criterion in determining the stability of the system. However, the assumption of time invariant coefficients for the system was a rather severe restriction on the dynamics of the problem, because it would imply that the direction of the cutting forces will be stationary relative to the structure of the machine tool. This condition is suitable for processes such as turning, broaching etc.

Recognizing this shortcoming, a comprehensive theoretical analysis of milling dynamics has been performed in 1968 by Sridhar et al. [SHL68]. To develop the system equations, the milling process is subdivided into three elements: cutting process, structure dynamics, and feedback loop. The mathematical equation resulting from this approach is an ordinary linear differential - difference equation with *periodic* coefficients:

$$\Omega \frac{d}{d\theta}\mathbf{r}(\theta) = \mathbf{A_0}\mathbf{r}(\theta) + \mathbf{A_1}(\theta)[-\mathbf{r}(\theta) + \mu\mathbf{r}(\theta - \Delta)] \tag{6.7}$$

where Δ is the pitch angle. μ is the overlap factor which is usually 1 for milling, and θ is the tool rotational position with respect to a fixed coordinate system. The vector $\mathbf{r}(\theta)$ is finite dimensional and its dimension depends on the number of significant modes of vibration of the structure. The quantity Ω is the spindle speed, A_0 is a constant matrix and matrix $A_1(\theta)$ is periodic with period Δ. Since the coefficients $A_1(\theta)$ in equation 6.7 is non-stationary, then the frequency analysis techniques are not applicable to this type of differential equations. Accordingly, Sridhar et al. [SHL68] studied the stability of the system by examining the eigenvalues of the system's stable transition matrix at the completion of one period of revolution of the cutter. Mathematically, it was proven that the system will be asymptotically stable if all the eigenvalues computed lie inside the unit circle in the complex plane. This can be recognized as the first detailed mathematical model of the dynamic milling process. However, the formulation was rather complicated and did not provide a clear relationship for the (critical) depth of cut at the onset of chatter.

Minis and Yanushevsky [MY93] presented yet another comprehensive analytical method for solving the dynamic milling model presented by Sridhar et al [SHL68]. The proposed stability method is based on Fourier analysis and the concept of parametric transfer functions [Zad50], which is recapitulated briefly in the following. Consider

Λ be a *linear operator* such that: $y(t) = \Lambda[x(t)]$, where $y(t)$ and $x(t)$ are input and output signals. By definition, the parametric transfer function associated with operator Λ is $W(\lambda, t)$, such that: $W(\lambda, t) = e^{-\lambda t}\Lambda[e^{\lambda t}]$ in which λ is a complex number. If the input signal is $x(t) = e^{\lambda t}$ then one will have:

$$y(t) = \Lambda[e^{\lambda t}] = e^{\lambda t}W(\lambda, t) = W(\lambda, t)x(t).$$

The dynamics of the milling process, equation 6.7, can be alternatively written as following:

$$\mathbf{F}^{(m)} + b\Lambda(\mathbf{F}^{(m)}) = 0. \tag{6.8}$$

where $\mathbf{F}^{(m)}$ is the resultant cutting force and b is the axial depth of cut. Λ is the product of operators $A(D, t)G_M(D)$ where operator $A(D, t)$ is a 2×2 matrix containing periodic coefficient functions of cutting forces and $G_M(t)$ is the flexibility matrix of the structure. By Fourier representation of the parametric transfer function $W(\lambda, t)$ the stability problem is shown to be reduced to the analysis of a finite order characteristic equation with constant coefficients, as follows:

$$\begin{aligned} \mathbf{F}^{(m)}_{(\mu)} &= -b\sum_{k=\infty}^{\infty} W(\lambda + ik\omega)\mathbf{F}^{(m)}_{(k)}. \\ \mu, k &= 0, \pm 1, \pm 2, \ldots \end{aligned} \tag{6.9}$$

where $\mathbf{F}^{(m)}_{(\mu)}$,$\mathbf{F}^{(m)}_{(k)}$ represent μ^{th} and k^{th} Fourier series coefficients of $\mathbf{F}^{(m)}_{p}(t)$ respectively, and $W_{\mu-k}(\lambda + ik\omega)$ is the $(\mu - k)^{th}$ Fourier series coefficient of the parametric transfer function $W(\lambda + ik\omega, t)$. The algebraic linear system given by equation 6.10 has nontrivial solution if its determinant is zero; i.e.,

$$det[\delta_{\mu k}I + bw_{\mu-k}(\lambda + ik\omega)] = 0. \tag{6.10}$$

$$\omega = 2\pi/T, \mu, k = 0, \pm 1, \pm 2, ...$$

where $\delta_{\mu k}$ is the Kronecker delta and I is the 2×2 identity matrix. Equation 6.10 is the characteristic equation of the closed loop milling system. For the system to be stable all the roots (eigenvalues) of the characteristic equation must be located in the negative half of the complex plane. In this formulation the eigenvalues of the system depend directly on the depth of cut h, which has been traditionally used as the measure of the stability in most of chatter studies. Location of the approximate roots of the characteristic equation can be obtained by using a truncated version of Fourier representation of the parametric transfer function and Nyquist criteria of stability. The solution algorithm is as follows: For a given spindle speed, a value for the axial depth of cut will be selected. Next, the chatter frequency is swept from 0 toward $+\infty$ to generate the Nyquist plot of the approximate characteristic equation. The system is stable if the Nyquist plot does not contain the origin of the complex plane. If that is the case then the axial depth of cut is gradually increased to a value that puts the system in an unstable condition. For a selected range of spindle speeds, the same iterative procedure is continued. Both the spindle speed and the axial depth of cut are obtained, stored at each iteration, and finally used to generate stability lobe plots. Due to the iterative procedure, generating the stability lobes for a wide range of spindle speeds could be quite time consuming specially if the higher order of the Fourier coefficients are used to approximate the characteristic equation of the system. The stability lobes generated in [MY93] is shown to be practically the same as Sridhar et al [SHL68].

Budak and Altintas [BA95] presented another alternative approach for the stability analysis of the milling operation. In their approach the time varying dynamic cutting force coefficients are approximated by their Fourier series components, as in Minis and Yanushevsky [MY93]; however, the chatter free axial depth of cuts and

spindle speeds are calculated directly from a set of a proposed linear analytical expressions. The analytical expressions are shown to be able to provide practically the same results obtained by the numerical solution of Sridhar et al. [SHL68], the iterative analysis of Minis et al [MY93], and the time domain simulations of Smith and Tlusty [ST93]. Since the formulations proposed in [BA95] are simpler than those of the other alternative approaches, and more closely resemble the basic chatter stability theory proposed by Tobias [Tob65] and Tlusty [KT67], it was selected for use in this thesis.

6.2.3 Ball end milling stability analysis

To apply the available analytical techniques of stability lobe prediction to ball end mills, it is essential to model the mechanics of ball end mills using the following type of formulations:

$$\begin{aligned} F_t &= K_T(Z) b h_{av}(\psi_j), \\ F_r &= K_R(Z) F_t. \end{aligned} \tag{6.11}$$

where F_t and F_r are the total tangential and radial forces applied to the tooth number j at rotational position ψ_j, $K_t(Z)$ and $K_r(Z)$ are the specific cutting constants, b is the depth of cut and $h_{av}(\psi_j)$ is the average chip thickness defined as $h_{av}(\psi_j) = (h_t + h_b)/2.0$ in which h_t and h_b are the chip thickness at the highest and lowest point of engagement along the cutting edge, respectively. To find the unknown functions of $K_t(Z)$ and $K_r(Z)$, the radial and tangential components of the cutting force is determined using static force predictions of the method of Orthogonal Cutting Conditions proposed by Yang and Park [YP91]. For any given values of b and d, see Figure 6.4(b), Z is

defined as following:

$$Z = -(d + b/2) \tag{6.12}$$

and the associated $K_t(Z)$ and $K_r(Z)$ are found such that the tangential and radial forces computed from equation 6.11 match with those of Yang and Park [YP91]. A series of K_t and K_r values is determined for different Z values and then using least squares method the best possible third degree polynomial is fitted to the data. The computed $K_t(Z)$ and $K_r(Z)$ are as follows:

$$\begin{aligned} K_t(Z) &= 3636.5 - 9.2Z - 1.1Z^2 - 0.2Z^3 \\ K_r(Z) &= 1734.6 + 42.2Z + 0.3Z^2 + 0.4Z^3. \end{aligned} \tag{6.13}$$

Expression 6.13 is generated for up-milling process of a 25.4 mm ball end mill with spindle speed of 600 rpm and feed per tooth of 0.1 mm. Figure 6.3, compares the above equations with the cutting constants determined from Yang and Park's model [YP91].

Stability surfaces in ball end milling

As it was shown earlier, the effective chip thickness of an engaged tooth can be expressed as follows [TI81]:

$$\begin{aligned} h_{inst} &= g \cdot (f \sin(\psi) + x \cos(\psi) + y \sin(\psi) + MAX(u_1, u_2, u_3)) \\ &= g \cdot (f \sin(\psi) + \bar{h}_{av}) \end{aligned} \tag{6.14}$$

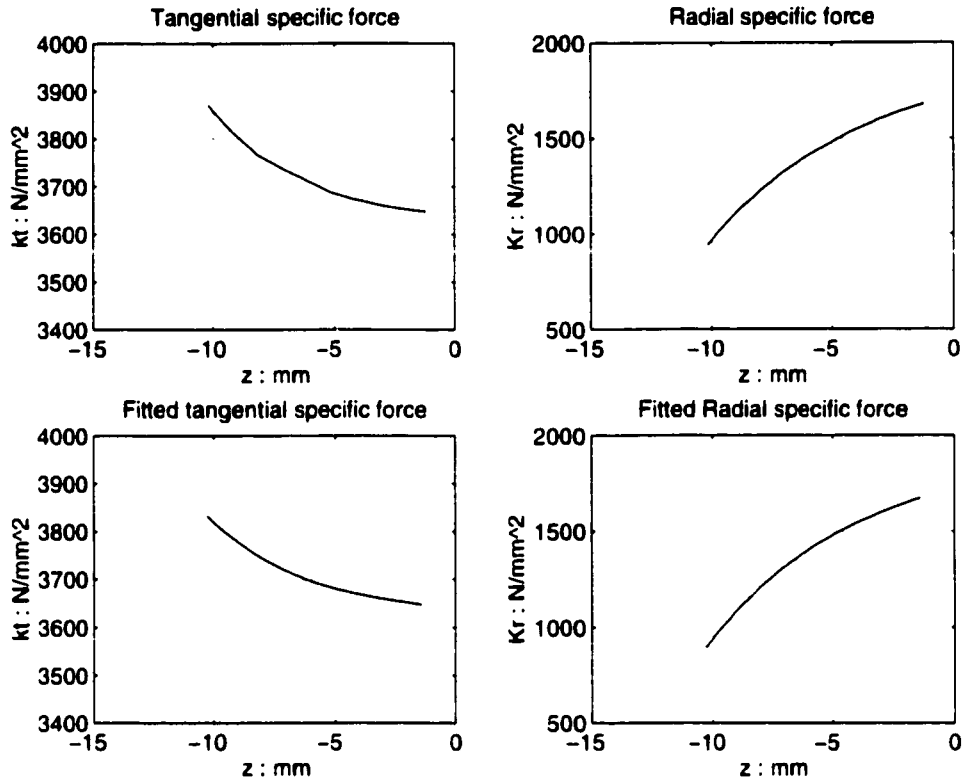
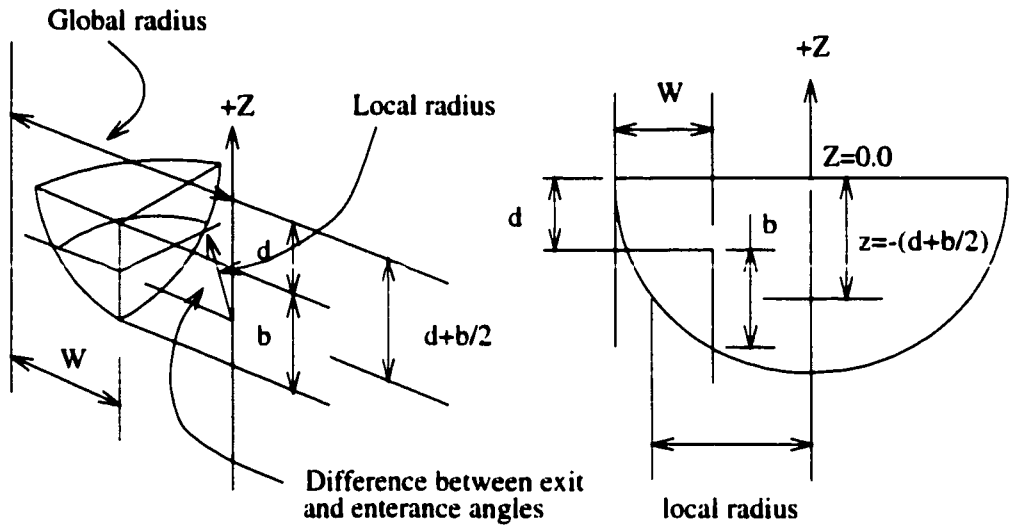


Figure 6.3: Comparison of Yang and Park's model with equation 6.13.

Where g is the unit step function which determines whether the tooth is in or out of cut [BA95]:

$$g(\psi_j) = \begin{cases} 1 & \text{if } \psi_{st} \leq \psi_j \leq \psi_{ex} \\ 0 & \text{otherwise} \end{cases}$$

where ψ_{st} and ψ_{ex} are the start and exit angles of the cutter as shown in Figure 6.4(b). Since the static component of the instantaneous chip load $f\sin(\psi)$, does not contribute to the dynamic chip load regeneration, it can be removed from equation 6.14 and the remaining components denoted by $\bar{h}_{av}$, will be used to write the tangential and radial components of the cutting force acting on the tooth j at the tool



(a) Removed shoulder (b) Front view of removed shoulder

Figure 6.4: Geometric parameters of the tool swept volume

position ψ_j as follows:

$$\begin{aligned} F_{tj} &= K_t(Z) b \bar{h}_{av}(\psi_j) \\ F_{rj} &= K_r(Z) F_{tj} \end{aligned} \tag{6.15}$$

where $\bar{h}_{av}(\psi_j)$ is the average of the dynamic component of the chip load at the highest and lowest point of engagement. Equation 6.15 obtained for the tangential and radial components of the cutting force for ball end mills is essentially the same as that of flat end mills. The only difference is that in the former the cutting coefficients are seen to be functions of position along the cutting edge as opposed to constant values for the case of flat end mills. Accordingly, by using the equation 6.15 the analytical stability lobe predictions developed for flat end mills can also be applied for ball end mills. The remainder of this section, which is based on the work presented in [BA95] and [MY93], briefly summarizes the analytical stability lobe prediction for the flat end

mills. The algorithm required to adapt the flat end milling stability lobe predictions for the ball end milling is introduced in section 6.3. Resolving F_{tj} and F_{rj} determined from equation 6.15 into cutting forces in x and y direction and summing for all the teeth engaged will result in [BA95]:

$$\begin{aligned} F_x &= \sum_{j=0}^{N-1} (-F_{tj}\cos(\psi_j) - F_{rj}\sin(\psi_j)) \\ F_y &= \sum_{j=0}^{N-1} (F_{tj}\sin(\psi_j) - F_{rj}\cos(\psi_j)) \end{aligned}$$

where F_x and F_y are the varying components of the cutting force. Rearranging the above expression in matrix form will yield [BA95]:

$$\begin{Bmatrix} F_x \\ F_y \end{Bmatrix} = \frac{1}{2}\, b\, K_t \begin{bmatrix} a_{xx} & a_{xy} \\ a_{yx} & a_{yy} \end{bmatrix} \begin{Bmatrix} \Delta x \\ \Delta y \end{Bmatrix} = \frac{1}{2}\, b\, K_t\, [A(t)]\{\Delta(t)\} \tag{6.16}$$

where Δx and Δy represent time varying component of the chip thickness. The time varying *directional dynamic milling force coefficients* $A(t)$ are periodic functions with the tooth passing frequency $\omega = N\Omega$ and period of $T = 2\pi/\omega$, where N is the number of the teeth and Ω is the spindle speed. $A(t)$ can be represented by Fourier series as follows [BA95]:

$$\begin{aligned} [A(t)] &= \sum_{r=-\infty}^{\infty} [A_r] e^{ir\omega t}, \\ [A_r] &= \frac{1}{T}\int_0^T [A(t)] e^{-ir\omega t} dt. \end{aligned} \tag{6.17}$$

For the most simplistic approximation, the average component of the Fourier series expansion can be considered, i.e. $r = 0$. Then the time invariant but immersion

dependent matrix of directional cutting coefficients will be determined as follows:

$$[A(0)] = \frac{1}{\psi_p}\int_{\psi_{st}}^{\psi_{ex}} [A(\psi)]\, d\psi = \frac{N}{2\pi}\begin{bmatrix} \alpha_{xx} & \alpha_{xy} \\ \alpha_{yx} & \alpha_{yy} \end{bmatrix} \tag{6.18}$$

where ψ_p is the pitch angle $2\pi/N$. Substituting equation 6.18 in 6.16, the dynamic milling expression for ball end milling will be reduced to the following:

$$F(t) = \frac{1}{2}\, b\, K_t\, [A_0]\Delta(t) \tag{6.19}$$

where K_t and K_r are determined using equation 6.13.
If the transfer function matrix at the cutter contact zone is denoted by $[G(iw)]$ as follows:

$$[G(i\omega)] = \begin{bmatrix} G_{xx}(i\omega) & G_{xy}(i\omega) \\ G_{yx}(i\omega) & G_{yy}(i\omega) \end{bmatrix} \tag{6.20}$$

then equation 6.19 alternatively can be written as following:

$$\{F\}e^{i\omega_c t} = \frac{1}{2}\, b\, K_t[1 - e^{-i\omega_c T}][A_0][G(i\omega_c)]\{F\}e^{i\omega_c t}$$

which has nontrivial solutions if its determinant is zero [BA95]:

$$det(\ [I]\ +\ \Lambda[G_0(i\omega_c)]\ )\ =\ 0; \tag{6.21}$$

where $G_0[(i\omega_c)]$ is the *oriented transfer function matrix* determined as following:

$$[G_0(i\omega_c)] = \begin{bmatrix} \alpha_{xx}G_{xx}(i\omega_c) + \alpha_{xy}G_{yx}(i\omega_c) & \alpha_{xx}G_{xy}(i\omega_c) + \alpha_{xy}G_{yy}(i\omega_c) \\ \alpha_{yx}G_{xx}(i\omega_c) + \alpha_{yy}G_{yx}(i\omega_c) & \alpha_{yx}G_{xy}(i\omega_c) + \alpha_{yy}G_{yy}(i\omega_c) \end{bmatrix} \tag{6.22}$$

and Λ is the eigenvalue of the characteristic equation 6.21. The relationship between the chatter free axial depth of cut and the real part of Λ is shown in [BA95] to be:

$$b_{lim} = -\frac{2\pi\Lambda_R}{NK_t}(1+\kappa^2). \tag{6.23}$$

where κ is defined as following:

$$\kappa = \frac{\Lambda_I}{\Lambda_R} = \frac{\sin(\omega_c T)}{1-\cos(\omega_c T)}. \tag{6.24}$$

The relationship between the spindle speed n and the chatter frequency ω_c is determined as following [KT67]:

$$n_k = \frac{60\ \omega_c}{N(\epsilon + 2k\pi)} \tag{6.25}$$

Hence, for a selected chatter frequency and for any integer number of complete undulations imprinted on the surface of the work, i.e. $k = 0, 1, 2, \ldots$, an associated spindle speed n_k will be determined. For the case that the cross transfer functions of the system are neglected, i.e. $G_{xy} = G_{yx} = 0$, then characteristic equation 6.21 will be reduced to a quadratic equation [BA95]

$$a_0\Lambda^2 + a_1\Lambda + 1 = 0$$

where

$$\begin{aligned} a_0 &= G_{xx}(i\omega_c)G_{yy}(i\omega_c)(\alpha_{xx}\alpha_{yy} - \alpha_{xy}\alpha_{yx}) \\ a_1 &= \alpha_{xx}G_{xx}(i\omega_c) + \alpha_{yy}G_{yy}(i\omega_c) \end{aligned}$$

and the eigenvalues Λ will be obtained as:

$$\Lambda = -\frac{1}{2a_0}(a_1 \pm \sqrt{a_1^2 - 4a_0}) \tag{6.26}$$

6.3 Stability in shoulder removal of die cavities

Shoulder removal is one of the major applications of ball nose cutters in die cavity machining, see Figure 6.5. Since reducing the time of machining by selecting the most

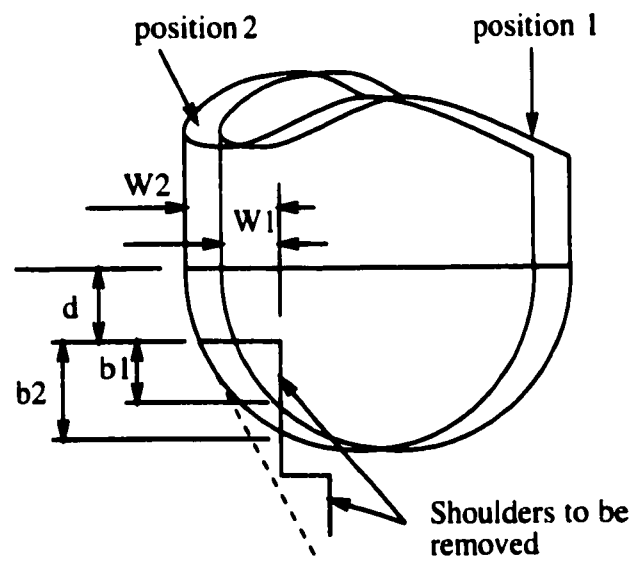


Figure 6.5: Shoulder removal process using ball end mill.

stable cutting conditions will increase the productivity and reduce the cost drastically, the stability of ball end mills in machining die cavity shoulders are studied using both analytical and time domain simulations. For the shoulder removal process, the algorithm for stability surface calculation is defined in the following :

1. Input geometrical information; the tool radius and d value, see Figure 6.4(b). Set the initial width of cut equal to the radius.

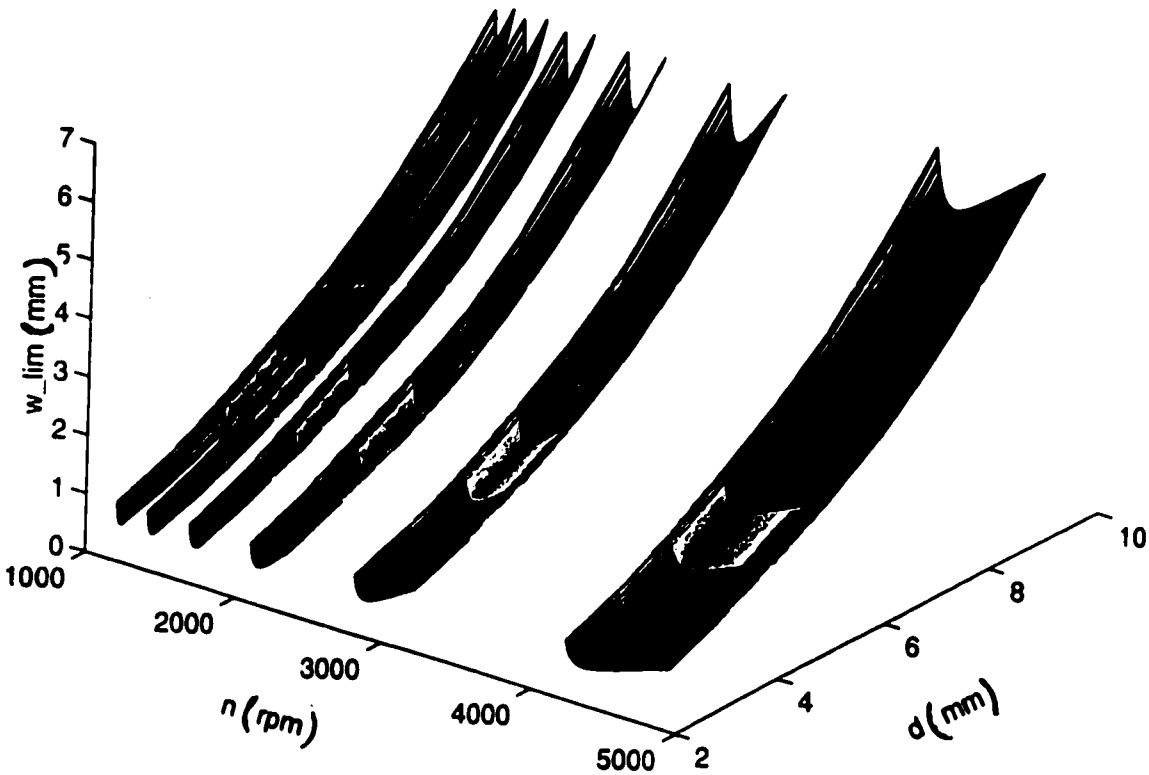


Figure 6.6: Stability surfaces in semi-finishing of die cavities.

2. Select a chatter frequency around a dominant mode,

3. Compute the transfer function matrix at chatter frequency, equation 6.20,

4. For the initial geometric values, evaluate K_t and K_r, equation 6.13,

5. Compute the oriented transfer function of the system, equation 6.22,

6. Solve the eigenvalue equation 6.26 for $\Lambda's$,

7. Compute the critical depth of cut from equation 6.23,

8. Compare the computed depth of cut with the initial value for convergence,

9. If converged go to step 13,

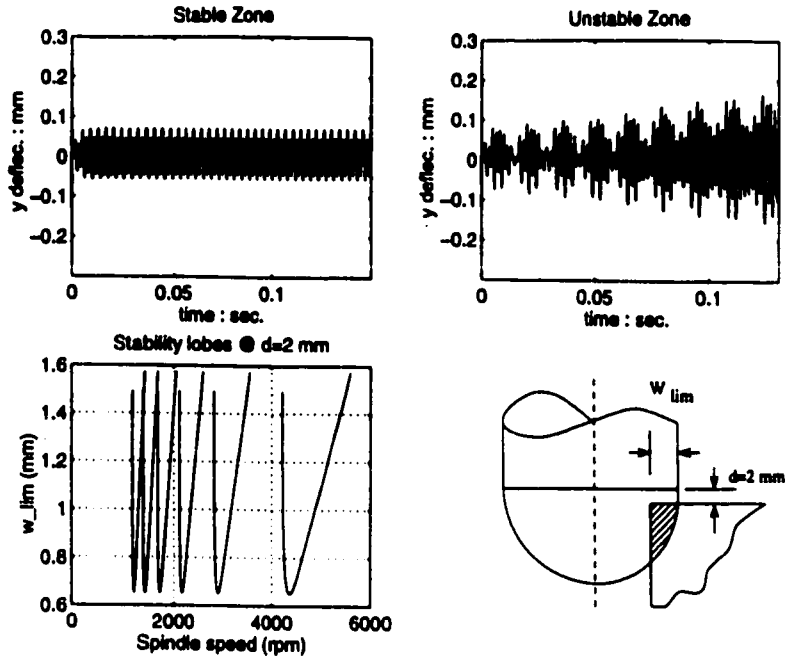


Figure 6.7: Comparison of time domain and analytical predictions : d=2 mm.

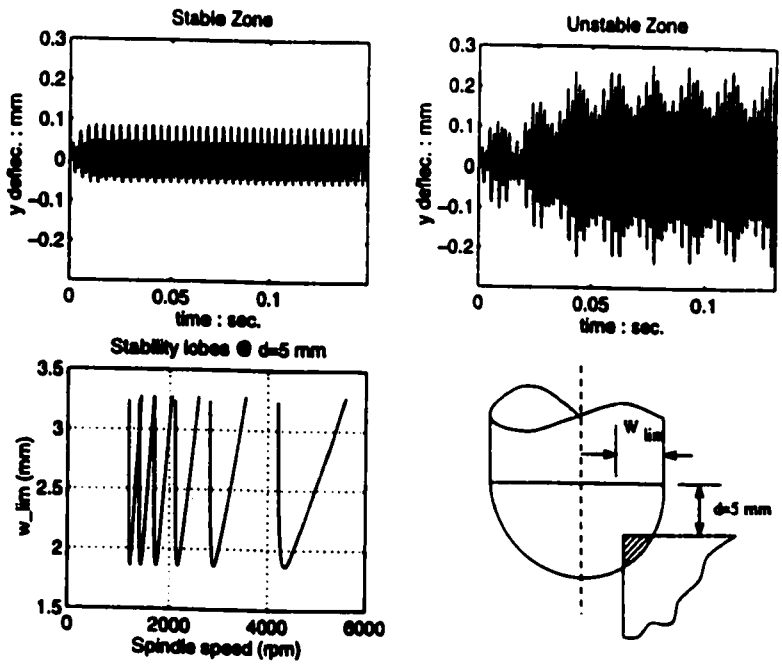


Figure 6.8: Comparison of time domain and analytical predictions : d=5 mm.

10. If not converged update Z of cut using equation 6.12,

11. Update ψ_{ex},K_t and K_r based on updated Z value,

12. Go to step 5,

13. Using equation 6.25 compute required spindle speed for any selected integer value of full imprinted oscillations on cut,

14. Save spindle speeds, critical widths of cut and d values for the stability surface plots.

15. Change d value and go to step 2, or stop.

Figure 6.6 shows the stability surfaces in the shoulder removal process using a 25.4 mm, four flute varying helix angle ball end mill, in up milling operation with 0.1 mm/tooth feed. In Figure 6.6, W_{lim} is the limit of radial depth of cut for the given d value that results in an unstable condition. Three cross sections of Figure 6.6 are shown in Figures 6.7, 6.8 and 6.9. Time domain simulation of two cases of stable and unstable conditions selected from the stability lobes are also included in Figures 6.7, 6.8 and 6.9. Comparison of the vibrations generated in the system shows spindle speed can reduce the level of vibrations drastically. Only the geometrical effect of the spindle speed on the stability lobes is studied in this thesis, i.e. equation 6.25. For the case that the effect of spindle speed on the mechanics of cutting is also included, curves $K_t(Z)$ and $K_r(Z)$ are surfaces of the form $K_t(Z,n)$ and $K_r(Z,n)$. These surfaces can be obtained by interpolating a series of $K_t(Z)$ and $K_r(Z)$ curves obtained for different spindle speeds using Yang and Park's model [YP91]. The iterative strategy above, has to be modified also to update K's as the spindle speed vary in generating stability surfaces.

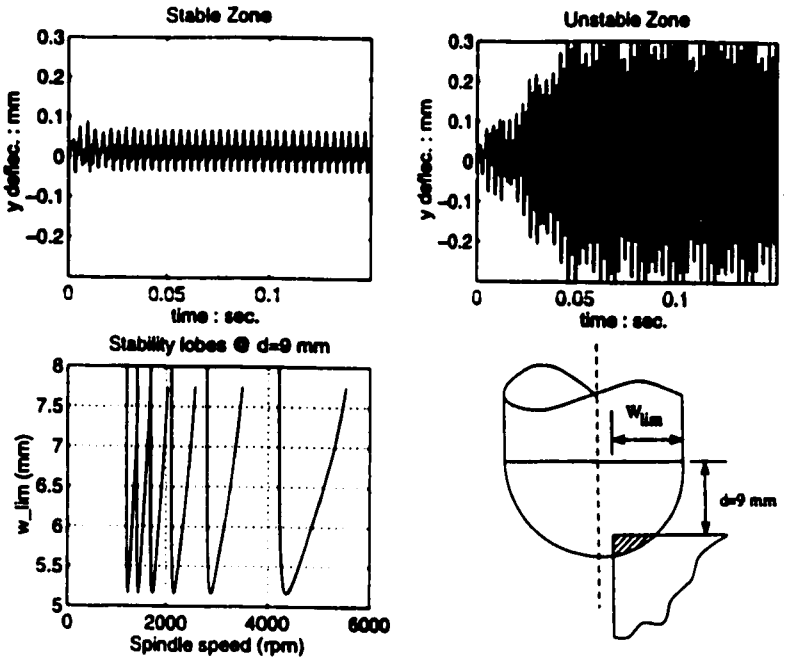


Figure 6.9: Comparison of time domain and analytical predictions : d=9 mm.

6.4 Dynamic response of the end mill cutter

For a simple dynamic analysis, where only tool dynamics is involved (such as the case of shoulder removal of die cavities) the response of the cutter as a whole can be approximated by its tip response. However, for an accurate surface prediction in flexible part machining, response of the each cutter element is to be determined along the tool axis. Since the cutter is numerically divided into thin disk like elements, the modal parameters obtained experimentally at the tip of the tool (table 6.1) have to be extrapolated in order to obtain the stiffness and the mass of each tool element [SE90]. The stiffness along the cutter axis is calculated using the cantilever beam theory. The following equations apply for both modes of x and y directions

$$K_{ii} = K^{*}\left[\frac{(CF)^3 - L^3 + 3L^2(L - CF)}{< CF - L_i >^3 - (L - L_i)^3 + 3(L - L_i)^2(L - CF)}\right] \tag{6.27}$$

where K_{ii} is the stiffness of the ith elemental disk, K^* is the modal stiffness at the tip of the tool and L is the overhang length of the cutter. Also, $L_i = \frac{3}{2}i\ dZ$ is the distance of the element i from the tip of the tool, and $CF = L - (\sum F_i(L - L_i)/\sum F_i)$ is the distance of the center of cutting force from tip of the tool. The derivation of equation 6.27 is given in Appendix B. Furthermore, the modal mass and damping of the each cutter element is given by

$$M_{ii} = M^*(1 - L_i/L) \tag{6.28}$$

$$C_{ii} = 2\zeta^*\sqrt{K_{ii}M_{ii}} \tag{6.29}$$

where ζ^* is the modal damping factor and M^* is the modal mass at the tip of the tool. Accordingly, the system of equations for the cutter elements is given by

$$\begin{aligned} M_{2ii}\ddot{u}_{2ii} + C_{2ii}\dot{u}_{2ii} + K_{2ii}u_{2ii} + K_{1ii}(u_{2ii} - u_{1ii}) + C_{1ii}(\dot{u}_{2ii} - \dot{u}_{1ii}) &= 0 \\ M_{1ii}\ddot{u}_{1ii} + C_{1ii}(\dot{u}_{1ii} - \dot{u}_{2ii}) + K_{1ii}(u_{1ii} - u_{2ii}) &= F_{ii} \end{aligned}$$

The dynamic response of the tool elements, i.e. $\delta^T(z) = \delta x^T(z) + \delta y^T(z)$, is then computed using a fourth order Runge-Kutta method (see Appendix C).

6.5 Workpiece dynamic response analysis

Dynamic response of the workpiece structure to the instantaneous cutting forces is computed as follows:

- All the finite elements nodes at the vicinity of the cutting edge are extracted.
- The instantaneous components of the cutting force are distributed on the neighboring finite element nodes.

- For the time period of one tool rotational step, a force history file is generated such that the cutting force components of the previous step are applied at the start and the current cutting forces are applied at the end of the period, see Figure 6.10.
- Finite element code is called for a response analysis.

For the time period that none of the cutting edges are in contact with the workpiece, a free vibration analysis is done on the workpiece.

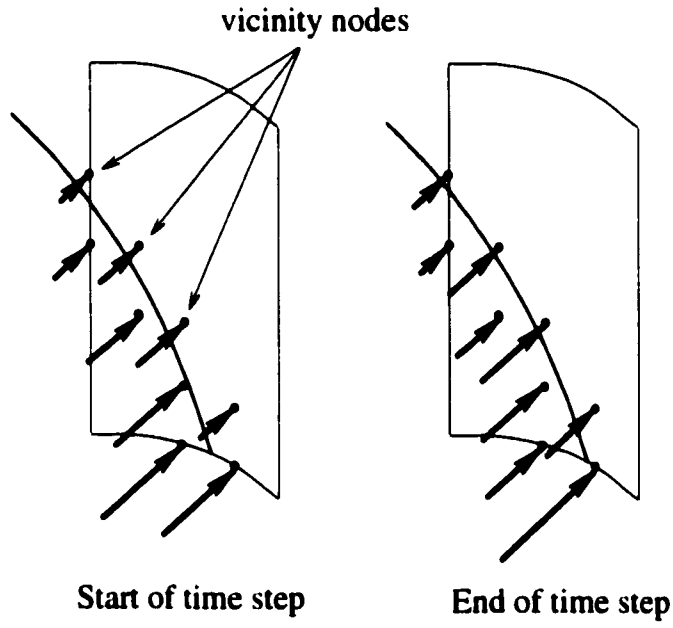


Figure 6.10: Nonlinear distribution of cutting forces.

6.6 Workpiece deflection along the cutting edge

For the case that the dynamics of both tool and the workpiece are considered in the simulation, the Effective System Deflection (ESD) is defined as the algebraic

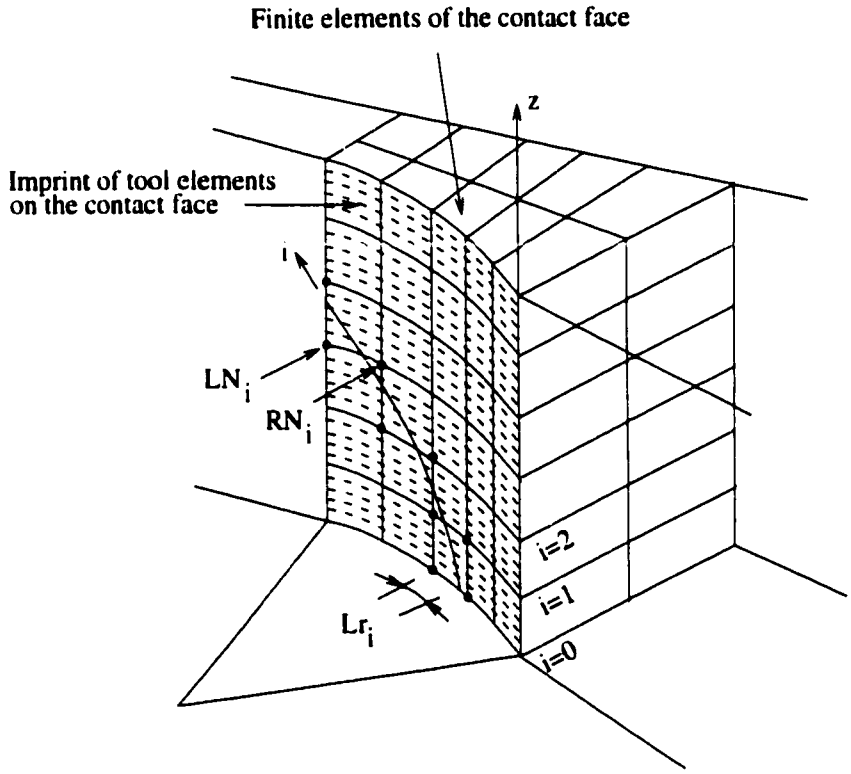


Figure 6.11: Workpiece deflection along instantaneous cutting edge position.

summation of tool and the workpiece deflections. The ESD is then fed back into the cutting process to account for the effect of the regenerativity. Shown in Figure 6.11 are the imprint of the tool elements (dashed lines) and the workpiece finite elements (solid lines) on the tool/workpiece contact face. The parameter Lr_i is the *normalized* distance between the left node of layer i and the cutting edge. By computing the dynamic response of the workpiece at the neighboring left and right nodes of each layer, i.e. LN_i and RN_i, then the following equations can be used to write for the

response of the workpiece *along* the cutting edge

$$\begin{aligned} \delta u_i &= \delta u_i^{LN} + Lr_i(\delta u_i^{RN} - \delta u_i^{LN}) \\ \delta u^W(z) &= \delta u_i + \frac{z - z_i}{z_{i+1} - z_i}(\delta u_{i+1} - \delta u_i) \end{aligned} \tag{6.30}$$

where δ_u stands for the deflections in both x and y directions. The dynamic response of the workpiece along the instantaneous rotational position of the cutting edge, can be determined by $\delta^W(z) = \delta x^W(z) + \delta y^W(z)$. The effective instantaneous deflection in system is then determined through the summation of projections of the tool and workpiece deflections along the direction normal to the work surface, (see Figure 6.12)

$$\overrightarrow{ESD}(z) = \delta^W(z) \cdot \Delta + \delta^T(z) \cdot \Delta \tag{6.31}$$

The equation 6.31 is then used as the dynamic term of the chip load in equation 6.4.

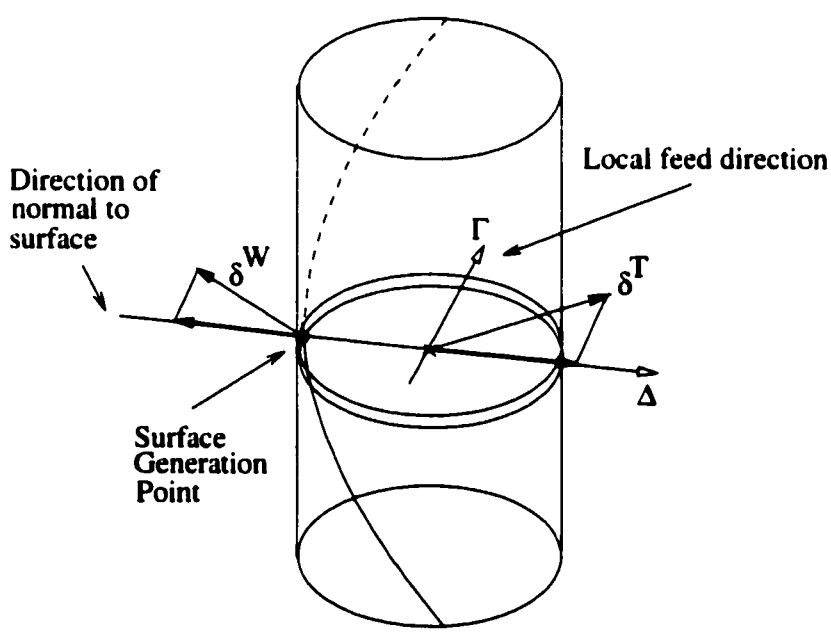


Figure 6.12: Effective deflection in tool/workpiece system.

6.7 Simulation of surface error in dynamic milling

For the case of expensive parts where the added value of the machined component is high, it is desirable to predict the machined surface texture and errors before the actual machining. Machined surface can be predicted by defining the Surface Generation Point (SGP) as the point in which an engaged tooth leaves a mark on the machined surface [KDS82]. This point is always located on the *entrance* curve of the contact face. For the case that the machining system is perfectly rigid, SGP_r is simply the intersection of the cutting edge with the entrance curve of the contact face. However, for a flexible part the actual position of SGP_f is determined by translating the position of SGP_r by its ESD vector

$$SGP_f = SGP_r \cdot \overrightarrow{ESD} \tag{6.32}$$

Finally, since the vector of effective deflections in the tool/workpiece system is defined to be normal to the local design surface, it can be considered as the local surface error. The length of ESD vectors (surface errors), are functions of height along the tool axis, and distance along the tool path. A 3D-graph of simulated surface errors is shown in Figure 6.13.

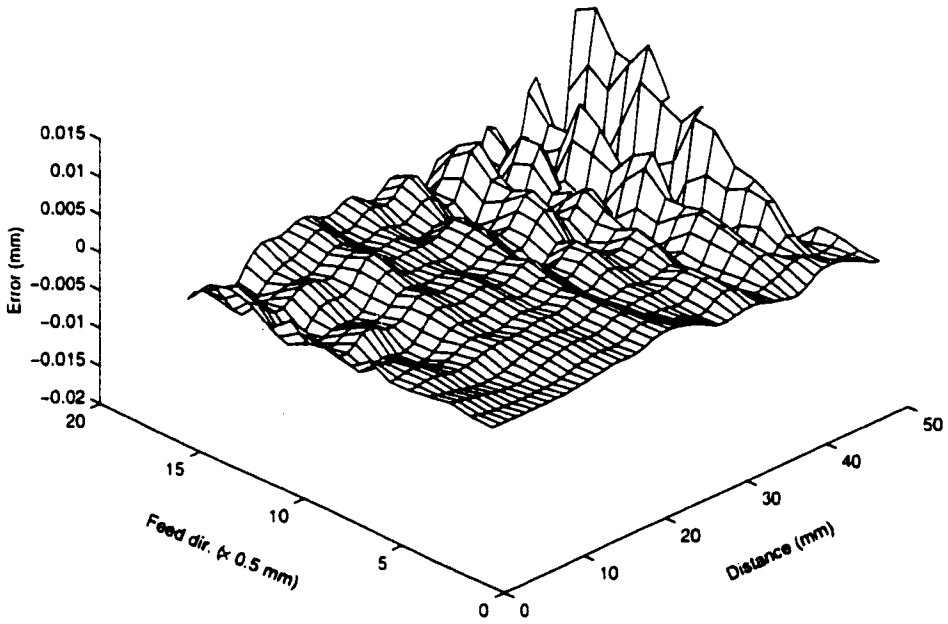


Figure 6.13: Distribution of surface errors in 4-axis milling of a flexible ruled surface.

Chapter 7

Experimental verification and results

Figure 7.1, shows a CFFF thin walled workpiece with two ruled surfaces at its front and back faces that was machined for the experimental verification of the developed software. The rail curves for the workpiece are two straight lines that are angled 12.5° from each other along the rule. The structural twist and the slenderness of the workpiece represent two basic characteristics of many flexible components in the aircraft and power generation industries. The initial geometry of the workpiece was machined out of a block of steel 1045 using the Fadal 5-axis machining center. Post-processing is referred to the process of generating actual machining NC codes from the computed CL data. Since post-processing is directly related to the number of axes involved in the machining, definition of various multi-axis milling operations comes first, then the case of five axis side milling will be discussed.

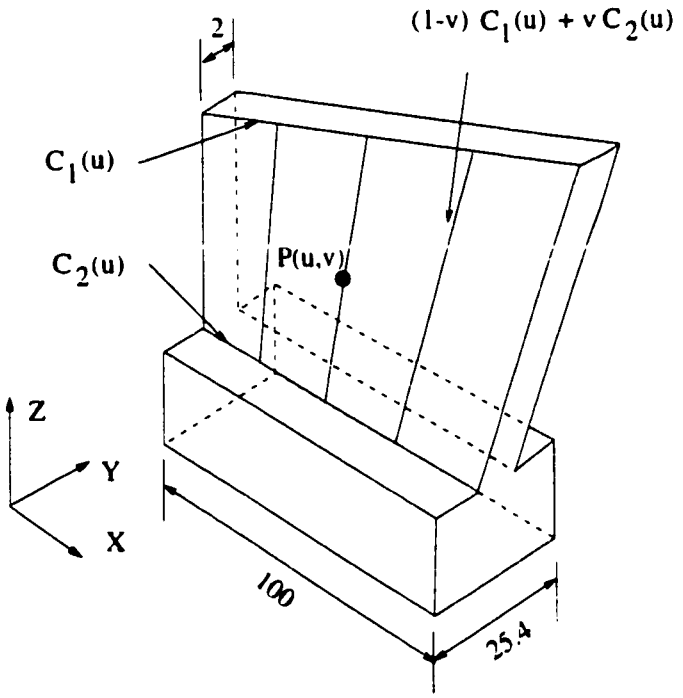


Figure 7.1: Ruled surface between two straight rail curves.

7.1 Various multi-axis milling operations

When milling with three to five axes the tool axis direction in its relative position towards the machine coordinate system MCS must be computed for each point on the path. The definition of the direction of the tool depends on the pre-specified kind of the processing. If for example the vector of direction of the tool axis T is identical with the Z axis of the MCS and the direction is not varying during the machining, then the milling can be categorized as 3 axis. If the vector of the tool axis is identical with the local normal of the surface, the milling can be 5 axes. The tool direction usually is represented as a unit vector in a clockwise rotating system of coordinates for the tool axis direction (K_r) with the unit vectors i, j and k. In the neutral position T is identical with k. In order to define the cutting process of

the tool in the coordinate system K_r, the tool axis vector T is first rotated about the j axis (dive angle) and then rotate about the k axis (pivoting angle). The definite tool direction for the computation of the rotational angles must be in the coordinate system MCS. The computation of the orientation of the tool coordinate system, i.e. K_r, defines the kind of milling process:

- **3 axis milling:** The orientation of the tool axis direction $K - r$ is equal to the machine coordinate system MCS. Dive and pivoting angles are both zero.

- **3 1/2 axis milling:** Similar to the 3 axis milling case but one or both of the dive or pivoting angles are non-zero but constant.

- **4 axis milling:** The vector i of the four axis milling coordinate system is pre-specified. If n is the surface normal, then we will have:

$$\begin{aligned} k &= n \times i \\ n &= i \times k. \end{aligned}$$

  Depending on the direction of normal n, the coordinate system rotates about the i axis to coincide its n axis with that of surface. If the i axis is parallel to the surface normal, there is not a possible solution for this case. If i is parallel to one of the two rotational axes, the 4 axes of the milling machine move synchronously.

- **4 1/2 axis milling:** Similar to the previous case, however, one of the rotating axes has a constant non-zero value while machining.

- **5 axis milling:** The vector i of the Cutter Contact (CC) coordinate system is parallel to the longitudinal tangent of the cutting path, the vector k to the surface normal and vector j is $j = k \times i$, i.e. cross tangent to the path. Accordingly,

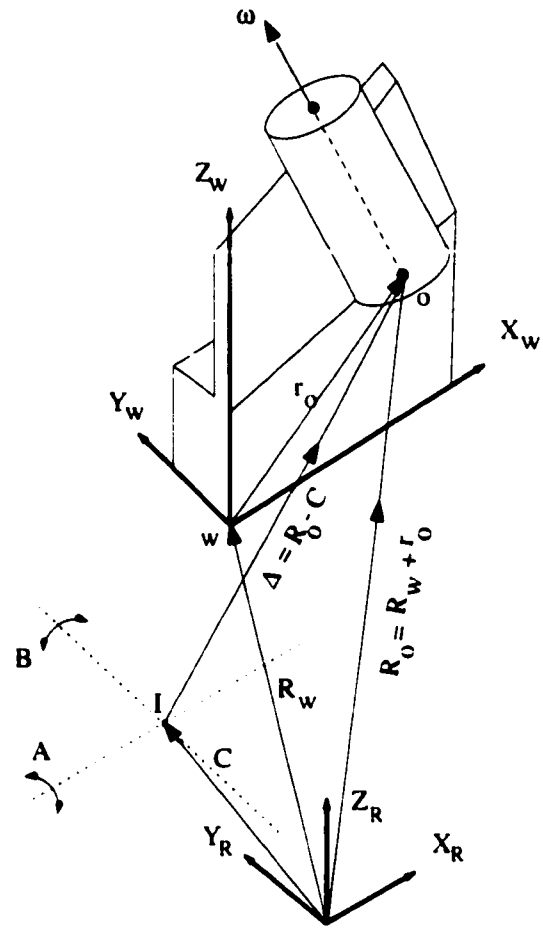


Figure 7.2: Post-processing the CL data for NC machining.

any dive or pivoting angle can be selected.

7.1.1 Post-processing of a five axis tool path

Since the orientation of the workpiece coordinate system (WCS) is optional, so it is supposed that the direction of x, y and z axes of WCS correspond to those of machine coordinate system (MCS) and that the direction of z axis is the same as the machine spindle (see Figure 7.2). Since the 5 axis CNC machine tool available (Fadal 4020) is

of type *rotational table on a tilting one*, here the post-processing of only this type of machine tools will be considered. Figure 7.2, represents all the positions, vectors and coordinate systems of interest involved in post-processing of a 5 axis tool path. The origin of coordinate system **w**, in which the workpiece geometry is defined, is denoted by the vector $\vec{R_w}$. Also, the intersection point of axes a and b is denoted I. For the Fadal 4020 machining center the position of I is seen to be at

$$\vec{C} = 9.3295\vec{i} - 0.1725\vec{j} - 345.297\vec{k}$$

where constants are in mm's. From the CL data file, the vectors of tool axis $\vec{\omega}$ and tool tip position $\vec{r_o}$ are both known in **w** coordinate system

$$\begin{aligned} \vec{\omega} &= \omega_{x_w}\vec{i} + \omega_{y_w}\vec{j} + \omega_{z_w}\vec{k} \\ \vec{r_o} &= r_{x_w}\vec{i} + r_{y_w}\vec{j} + r_{z_w}\vec{k} \end{aligned} \tag{7.1}$$

Since the corresponding unit axis vectors of the coordinates systems **R** and **w** are parallel to each other, components of $\vec{\omega}$ can be used directly to solve for the rotation of the a and b axes

$$\begin{aligned} a &= \tan^{-1}\left(\frac{-\omega_{y_w}}{\omega_{z_w}}\right) \\ b &= \sin^{-1}(\omega_{x_w}) \end{aligned} \tag{7.2}$$

For the given a and b values, x, y and z coordinates of the tool tip are determined as follows [FS97]

$$\begin{aligned} x &= \Delta_x \cos(b) - \Delta_z \sin(b) + C_x \\ y &= \Delta_x \sin(a)\cos(b) + \Delta_y \cos(a) + \Delta_z \sin(a)\cos(b) + C_y \\ z &= \Delta_x \cos(a)\sin(b) - \Delta_y \sin(a) + \Delta_z \cos(a)\cos(b) + C_z \end{aligned}$$

where $\vec{\Delta} = \vec{R}_o - \vec{C}$, (see Figure 7.2.) Contrary to the conventional milling, where the feed rate is defined by the tool advancement in a unit of time, feed rate in five axis machining is usually defined by the inverse time required for the controller to move from one tool position/orientation to the next one. Accordingly, if the desired feed rate from one *tool tip* position to another, along the tool path, is denoted by f_0, then in the associated NC code, the following feed rate value should be used in the feed rate register

$$\frac{1}{T} = \frac{f_0}{\|\vec{R}_{(i+1)} - \vec{R}_i\|}$$

7.2 Experimental verifications and discussion

In this section the predicted static and dynamic cutting forces are compared with the experimental data. Effect of runout was filtered using a high pass filter set at the frequency of spindle rotation. The good agreement seen between the experimental and simulation results approves validity of the proposed formulation for the modeling of cutting forces in multi-axis side milling. Also, the texture left on the machined surface of a very flexible four-axis workpiece and its errors is compared with the simulation results. Again, the good agreement seen between the surface of the actual part and simulation, validates the models for the tool and workpiece dynamics, and more importantly for their interaction with the cutting process. The following is a comparison between the predicted static forces and the experimental data. Figure 7.4 is related to the up-milling operation of a one-axis straight tool path and 7.5 is related to that of a four-axis tool path. The axial and radial depths of cut and the tool diameter for both cases were 20, 0.65 and 25.4 mm. The spindle speed were 1200 and 1500 rpm and the tool tip feed rates were 125 and 100 mm/min; respectively. Because of machining with a long axial depth of cut, during the period of each tooth engagement, there exists a time period in which the length of in-cut segment remains

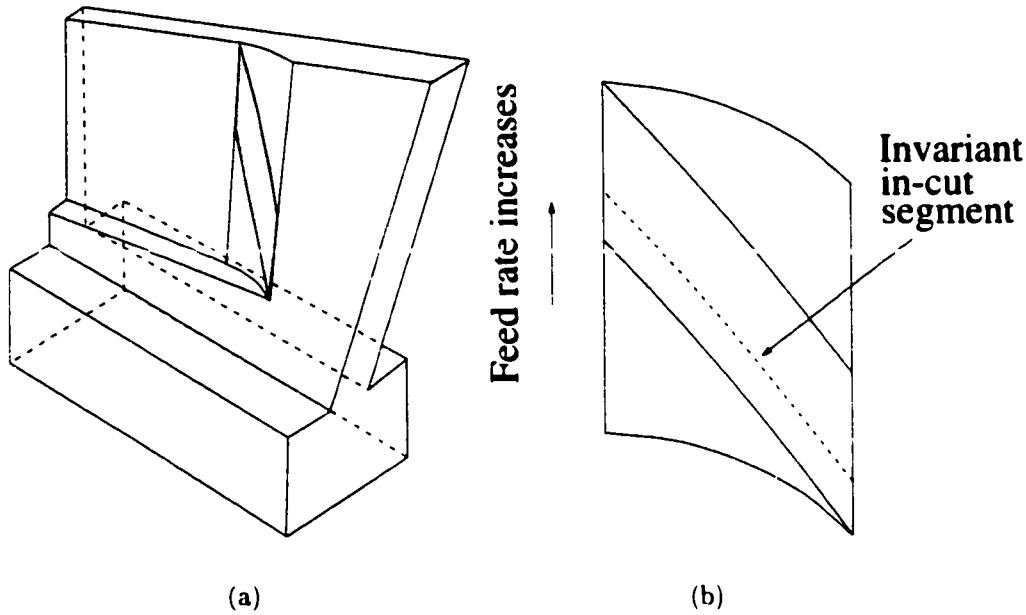


Figure 7.3: Constant immresion zone and feed variation in multi-axis milling.

invariant (see Figure 7.3(b).) As expected, the cutting force during this period is constant for the case of a linear tool path (see Figure 7.4.) On the other hand, as it was shown earlier, due to the rotation of the rotary axes in a multi-axis tool path, different heights of the cutter experience different feed rates. For the case of the four-axis experiment, the higher levels of the cutter have *higher* feed rates. Hence, with the rotation of the cutter the invariant in-cut segment moves *upward* and cuts with higher feed rates. As a result, the period of constant engagement in the multi-axis machining, does not necessarily generate a period of constant force, as shown in Figure 7.5.

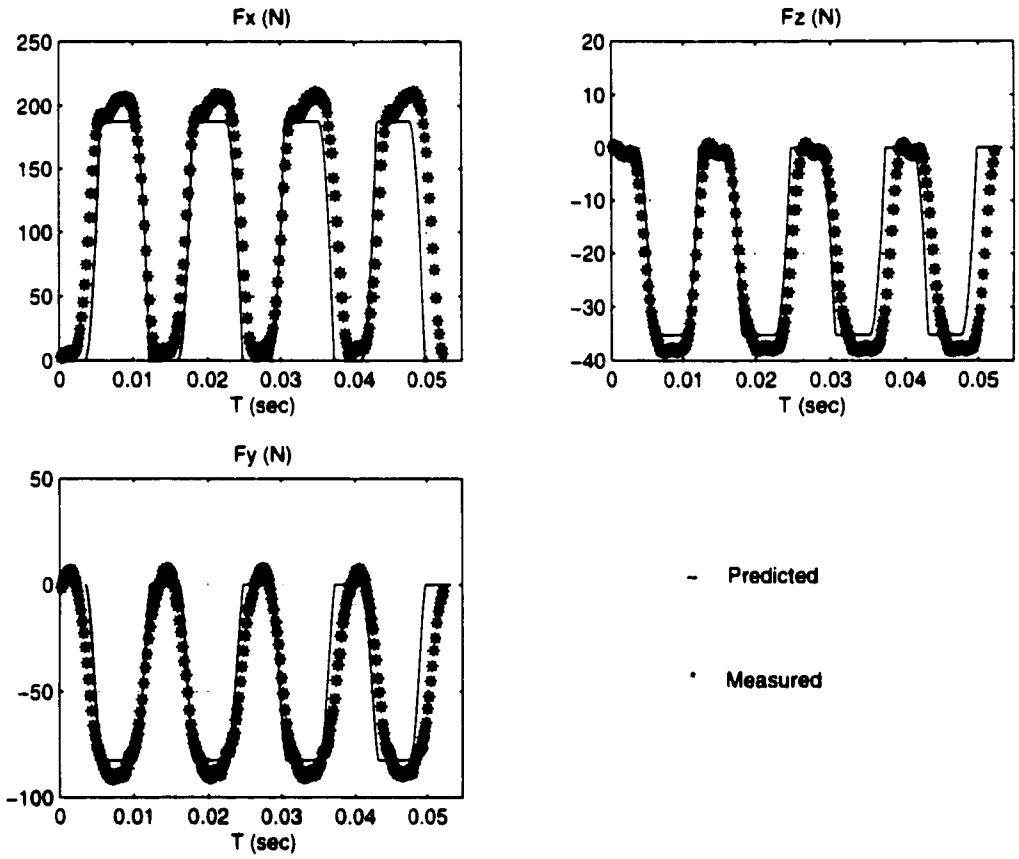


Figure 7.4: Simulated and experimental static cutting force in one axis milling.

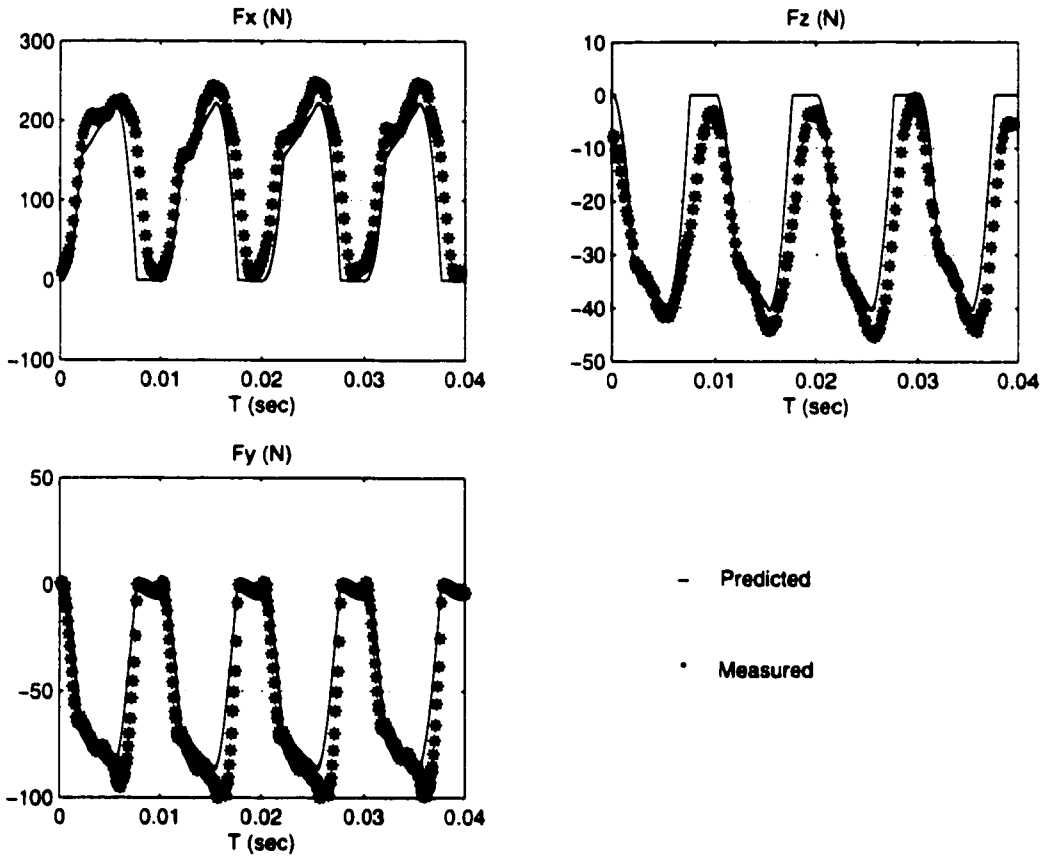


Figure 7.5: Simulated and experimental static cutting force in four axis milling.

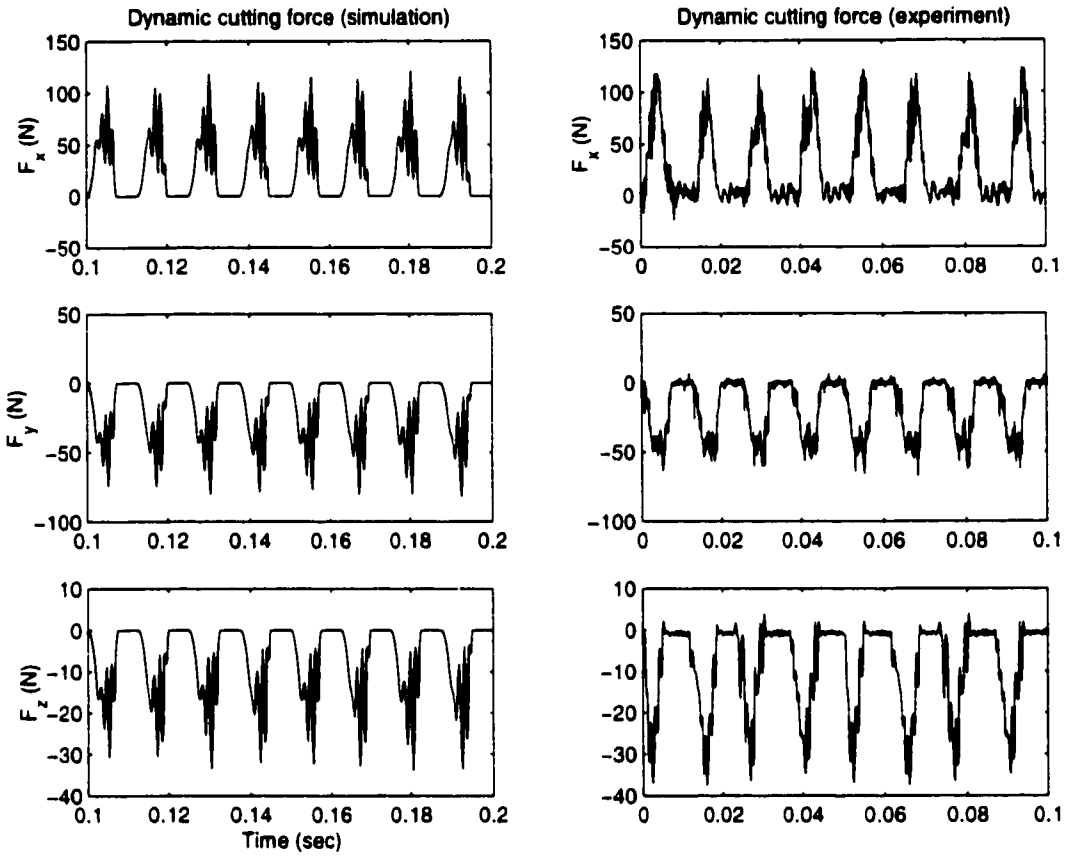


Figure 7.6: Simulated and experimental dynamic cutting force in four axis milling.

Comparison of the dynamic simulation and experimental data for the case that the workpiece thickness was reduced to 2.0 (mm) is shown in Figure 7.6. Again a reasonable agreement is seen between the predicted and experimental data. The power spectrum of the cutting forces reveals the frequency components of the measured or simulated signal. Figure 7.7 compares the power spectrum of the simulated cutting forces for the following three cases. For a rigid system, only the tooth frequency is observed (first row of Figure 7.7.) However, for the case of dynamic simulation consisting of the tool dynamic system only, both the tooth and the tool mode frequencies are seen in the power spectrum, i.e. second row in Figure 7.7. Finally, for the case

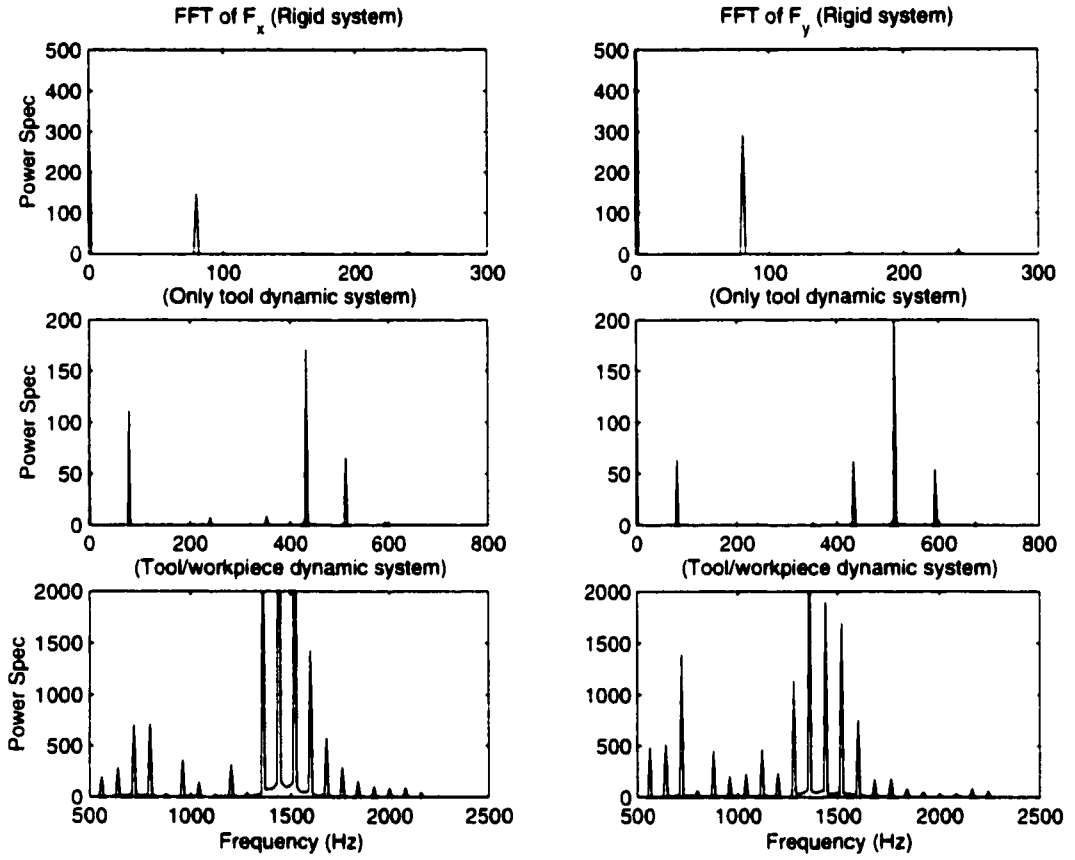


Figure 7.7: Power spectrum comparison of rigid and flexible systems.

where the dynamics of both tool and workpiece structures are included, the dominant modes are those of the workpiece.

An important question in the milling simulation of the flexible parts is: how often should the finite element mesh be updated? Figure 7.8 represents the power spectrum of the cutting forces sampled at the begining, middle and the end of the tool path for the workpiece with original thickness of 2.5 mm. Axial and radial depths of cut for these data are 20 and 0.5 mm, respectively. As it is expected, due to the considerable reduction of the workpiece stiffness during the machining, the dynamics of the workpiece changes noticeably. Accordingly, as shown in Figure 7.8 the dominant

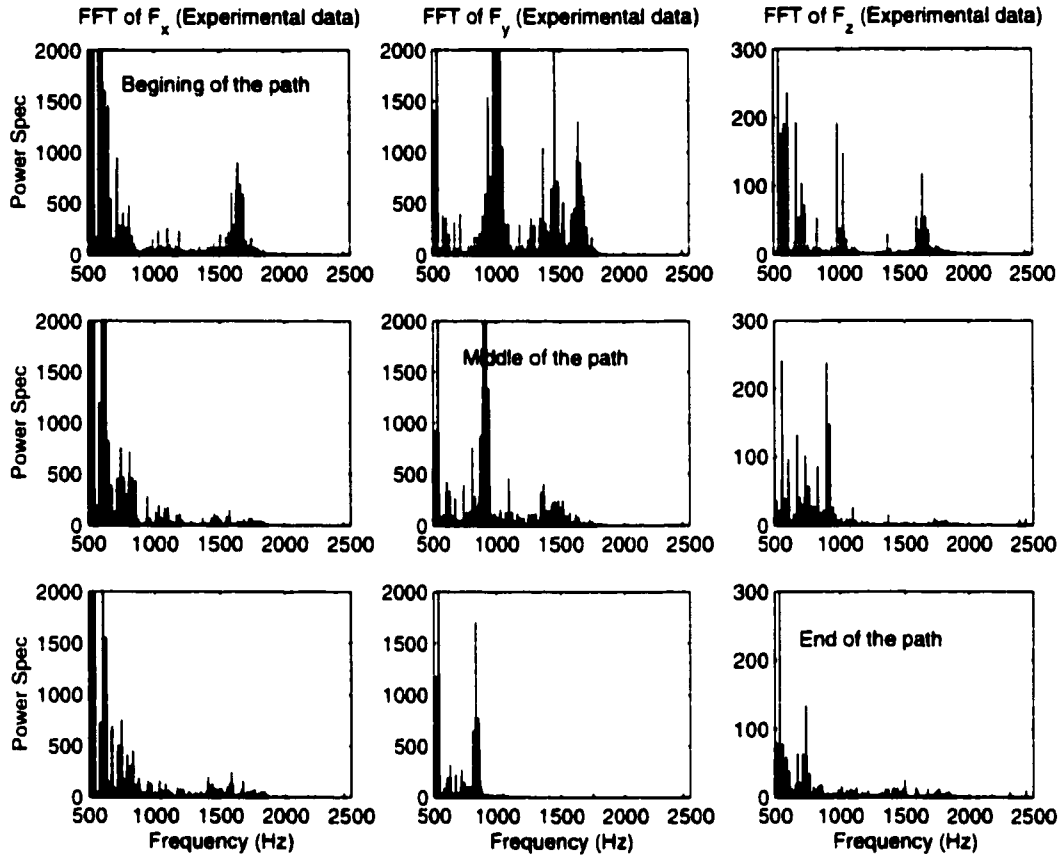


Figure 7.8: Experimental power spectrums at the positions along the tool path.

workpiece frequencies shift toward the smaller values. Similarly, Figure 7.9 is related to the power spectrum of the simulated cutting forces at the begining, middle and the end of the tool path. Again, as the process of metal removal causes the workpiece flexibility to increase, the dominant workpiece frequencies reduces. Considering the simplistic boundary condition, i.e. fixed nodes, used in the finite element analysis for the clamping of the workpiece, an acceptable agreement is seen between Figures 7.8 and 7.9. Also the validity of the modeling techniques developed for the surface generation and geometric error prediction in dynamic milling, is assessed by comparing the profiles of the simulated surface and its error with the actual surface of the workpiece.

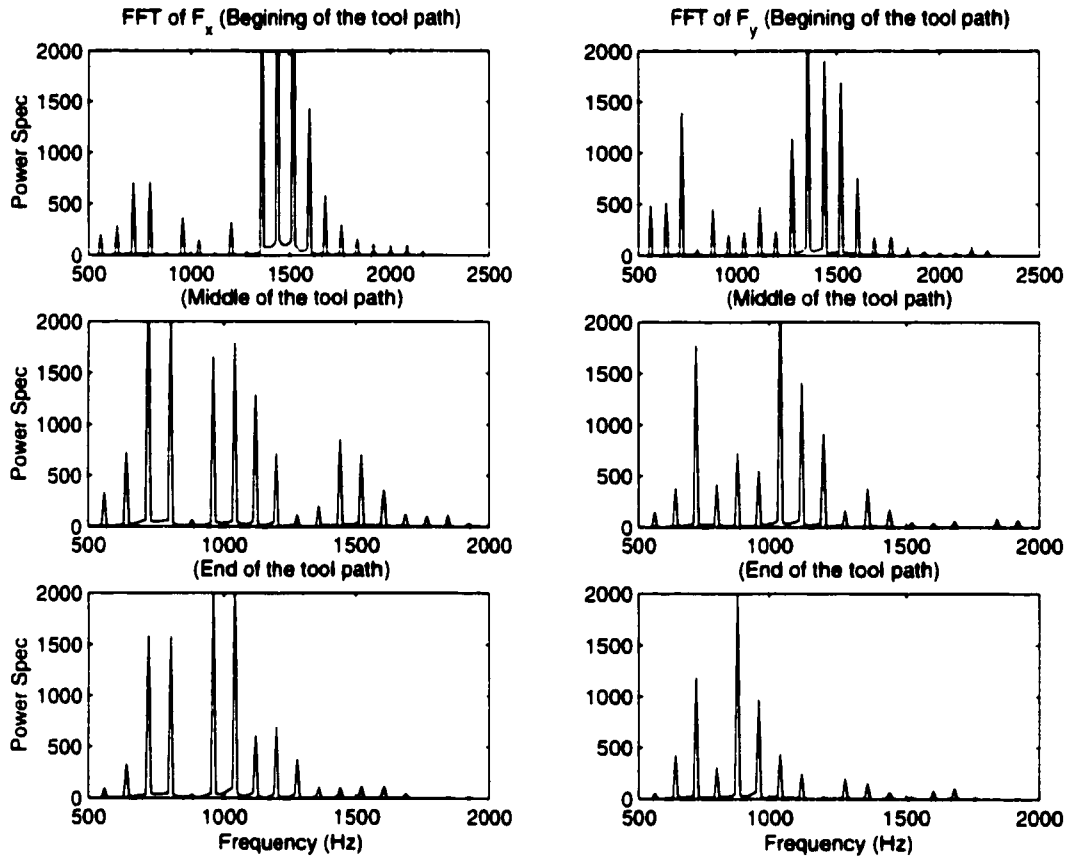


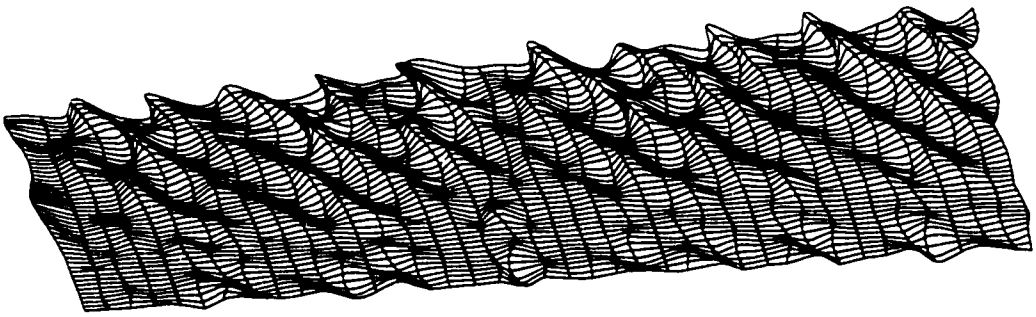
Figure 7.9: Simulated power spectrums at the positions along the tool path.

Figure 7.10 is a comparison between the predicted profile and the actual surface close to the end of the tool path. The area is 3 cm^2, and simulated deformations are magnified by a factor of two. Also, Figure 7.11(a) shows an area of 1 cm^2 of the simulated surface profile with a magnification factor of five. In both cases top of the workpiece is towards the top of the page. The corresponding area on the workpiece is shown in Figure 7.11(b). A good agreement is seen between the simulated profiles and the actual workpiece surface. The geometric inaccuracy seen on the work surface is the direct result of the workpiece flexibility. As the approaching tooth engages with the

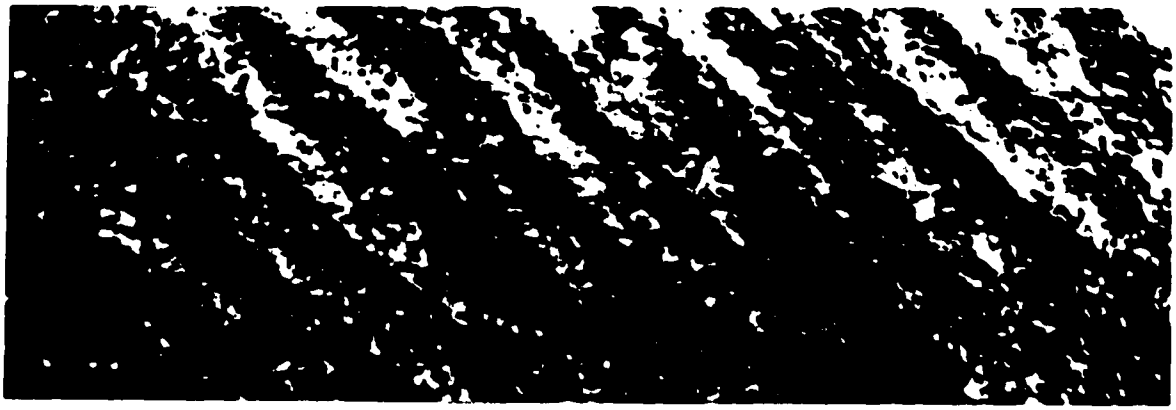
bottom of the contact face, due to the vibration of the workpiece, it encounters a deflected workpiece. As cutting proceeds, the Surface Generation Point moves towards the free edge of the workpiece where the vibration amplitude is highest. Accordingly, the surface error increases towards the free edge of the flexible workpiece (see Figure 7.12(a).) Also, by the time that the SGP reaches to the top of the contact face, the workpiece completes one or more of vibrational cycles; and as a result, a wavy surface is left behind. It can be shown that the wave length of these marks is dependent on the natural frequency of the in-process workpiece, spindle speed and the tool radius

$$\lambda = \frac{2\pi\ R\ \cdot ss}{60\ f\ \tan(\beta)} \tag{7.3}$$

where λ is the wave length and f is the frequency of the in-process workpiece vibration and ss is the spindle speed in rpm. For instance, since the length of the waves imprinted half way along the tool path on the surface of the workpiece (see Figure 7.11(b)) is around 4 mm, then for the spindle speed of 1500 rpm and helix angle of 30° degree, the estimated frequency of workpiece vibration is around 860 Hz, which matches with the expected frequency and the simulation results. Also, as it is shown in Figure 7.12, comparison between the machining error across the simulated surface profile with that of the actual surface (generated using TalySurf) reveals a close agreement between them. Due to the high flexibility of the workpiece (thickness of only 2.0 mm), the geometric errors are both positive and negative relative to the design surface, i.e. zero error. However, negative errors are higher in magnitude compared to the positive errors. This is known as the undercut effect of the up-milling operation, i.e. workpiece is under-size. Moreover, closer to the workpiece free edge the higher the errors, i.e. the workpiece vibrations. As suggested in [TSW96], for the case of long end mills this phenomenon can be avoided by using a relieved shank tool, which limits the tool workpiece contact to the intended cutting zone (see

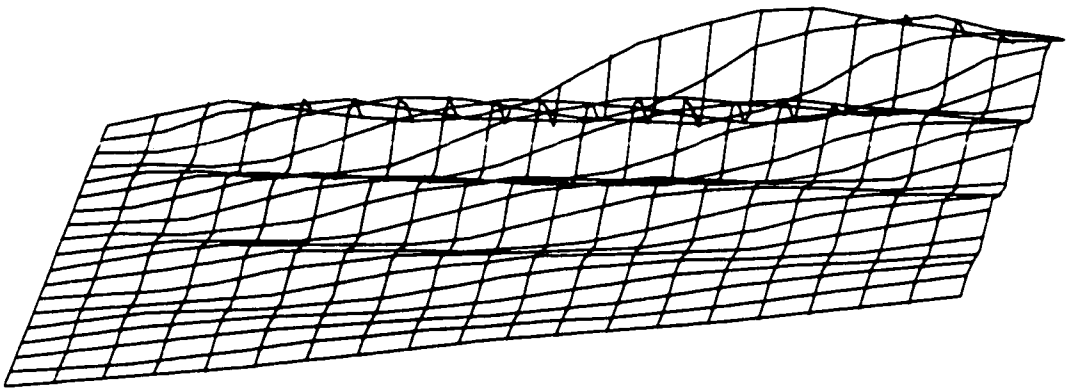


(a)



(b)

Figure 7.10: Comparison between the actual and simulated surfaces.



(a)

(b)

Figure 7.11: Simulation and actual surface profile in flexible part machining.

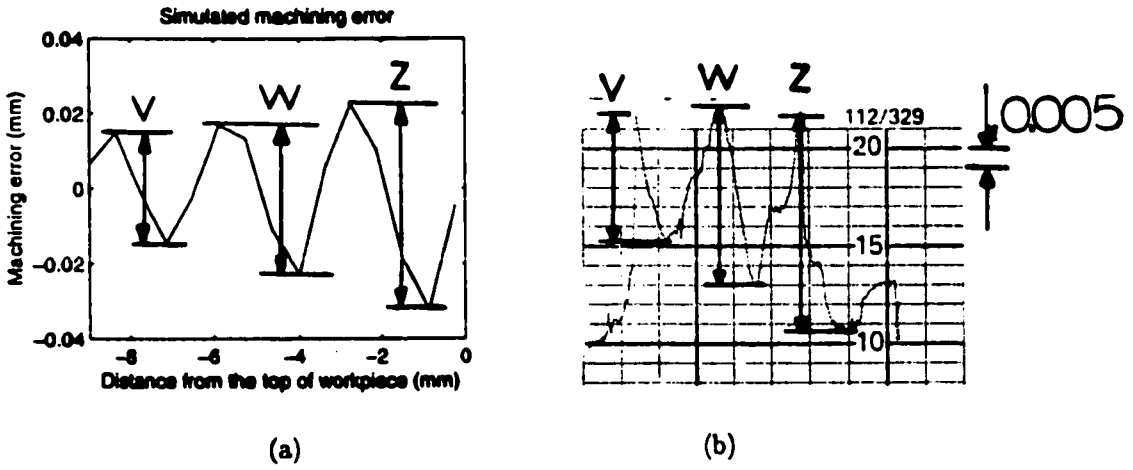


Figure 7.12: Comparison between the actual and simulation error distribution.

Figure 7.13.)

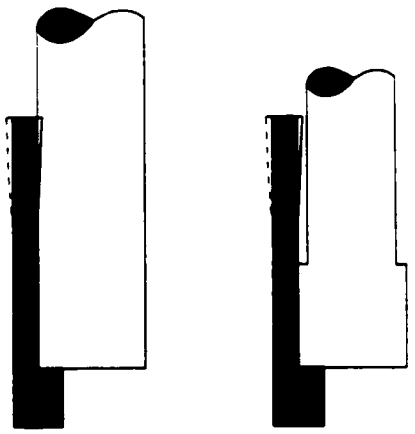


Figure 7.13: Relieved shank for thin workpiece milling.

Chapter 8

Summary and future work

8.1 Summary

In all the previous research work dealing with the problem of the interaction among the tool, workpiece and the cutting process, geometry of the workpiece is restricted to the rectangular plates and accordingly the tool path to the straight lines [SE90, AMB92]. Two main reasons can be mentioned for these shortcomings

1. Lack of an accurate and comprehensive geometric simulator.
2. Lack of a dynamic multi-axis side milling force model.

To address the first shortcoming, state of the art B-rep solid modeling technique is used in this thesis for all the geometric aspects of the milling simulation. Following is the summary of all the geometric computations involved

- To represent the cutter flute, a list of positions is generated using the end mill cutting edge equation and interpolated using B-spline curves. To generate a complete model of the cutter, copies are generated and rotated (depending on the number of the teeth).

- Kinematics of the multi-axis milling operation consists of rotation of the cutter and translation/rotation of the machine tables. This is simulated by the three transformations applied to the B-spline curve, i.e. rotation about the tool axis, translation along the tool path and rotations about the rotary axes.
- Solid based model of the design workpiece is generated (imported).
- Using the tool and workpiece geometric information, as well as the user defined machining tolerance, the multi-axis side milling CL data is generated.
- Using the information regarding tool position/orientation in the CL data file, sections of the tool swept solid are generated using the wire skinning technique.
- Advanced skinning formulation is then used to skin the swept solid sections and generate the multi-axis tool swept solid. A multi-axis tool library is generated for the three types of end mill cutters.
- In-process workpiece geometry is computed by the Boolean subtraction of the tool swept solid from that of the workpiece.
- Instantaneous flute/workpiece engagement are computed by intersecting the B-spline representation of the cutting edges with the boundary curves of the tool/workpiece contact face. The data is passed to the physical simulator.
- An automatic finite element mesh generation is developed for the meshing of the in-process workpiece geometry.

Also, to address the second shortcoming, a unified multi-axis force model is introduced in this thesis. The static force model is extended to a dynamic one in which the dynamics of both tool and workpiece are considered. The main features of the force model can be summarized as follows

- The concept of the equivalent orthogonal cutting condition (originally developed for the ball end milling) is extended to account for the effect of rotary axes.
- Using the orthogonal data base, the tapered and ball end milling operations can also be modeled using the formulation introduced.
- In the multi-axis milling, it is seen that the rotary axes introduce a variation to the feed parameter along the tool axis. To account for this observation, a vectorial milling force model is introduced. Depending on the angular speed of the rotary axes, the magnitude and direction of the local feed vector are changed along the tool axis.
- Also, in the multi-axis milling, the rotary axes are seen to introduce a shift angle to the axes of the coordinate system in which the cutter rotational position is measured.
- The proposed multi-axis side milling formulation reduces to the conventional one if a simple tool path is simulated.
- The deflection of the workpiece computed at the nodes of the tool/workpiece contact face, are interpolated to compute the workpiece deflections along the cutter flute.
- The Effective System Deflection, defined as the algebraic summation of the tool and workpiece deflections, is used to write for the dynamic term in the equation of the chip load in dynamic milling simulation.
- The Effective System Deflection at the Surface Generation Point is monitored to predict the machined surface profile.
- Predicted surface profile is compared with the design surface to predict the surface error in flexible part machining.

A twisted ruled surface was machined using a four-axis up milling operation and the static and dynamic forces were measured. To operatively see the effect of the workpiece flexibility on the machined surface profile and cutting forces, the thickness of the twisted blade was reduced to 2.0 mm. To demonstrate the effect of the multi-axis machining on the cutting forces, a long axial depth of cut was used. It was shown that in the multi-axis machining, variation of the feed rate along the tool axis had a profound effect on the pattern and the magnitude of the cutting forces. For a single axis tool path, the cutting force remained constant during the period of invariant in-cut segment. However, the same invariant in-cut segment, produced a varying force level in the multi-axis side milling operation. Finally, the surface profile left on the workpiece surface was compared with the simulation result. The good agreement seen between the experimental data and simulation results verified the validity of the following developed models and techniques.

1. The static and dynamic multi-axis cutting force models.
2. The interaction among the cutter, workpiece and the cutting process.
3. The prediction of surface profile/error in flexible part machining.

8.2 Future work

Possible continued research would include the following:

- In practice it is seen that the up/down milling operations tend to generate over/under-cuts. These form errors can be minimized by accounting for the system static deflections in the CL data file.
- The effect of the fixture on the surface profile can be included.

- Tool/workpiece engagement can be more accurately computed by statically deflected tool swept volumes. This is specially important for milling simulation of slender tools with long axial depths of cut.
- Using the formulation shown in section 5.1, the physical simulation can include the static deflection of the cutter.
- Effect of the chip flow angle on the force predictions can be investigated using equations of section 5.4.
- Traditionally, definition of the rake angle is for a static tool. For the case of a vibrating system, effect of the dynamic rake angle on the milling force can be investigated.

Appendix A

The cutter runout model

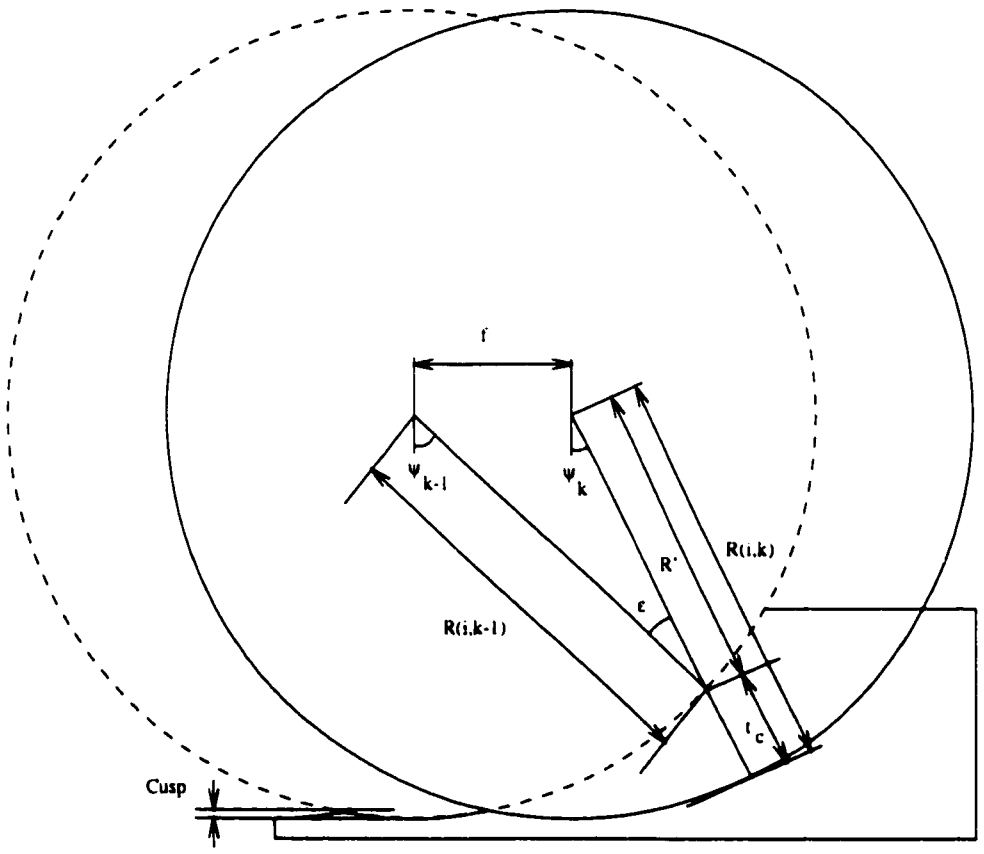


Figure A.1: Modification of chip thickness for runout

The effect of cutter runout on the chip thickness is studied and improved chip thickness equation is derived in [KD83]. As shown in Figure A.1, tooth number k at angle ψ_k is removing the material left by tooth $k-1$ at angle ψ_{k-1}. The actual chip thickness is then given by

$$t_c = R(i,k) - R' \tag{A.1}$$

where $R(i,k)$ is the radius of the kth flute at the ith axial disk. Using the law of sines for the triangle ABC,

$$\frac{R(i,k-1)}{\sin(\pi/2 + \psi_k)} = \frac{R'}{\sin(\pi/2 + \psi_{k-1})} \tag{A.2}$$

which leads to

$$R' = R(i,k-1) \times \cos(\psi_k)/cos(\psi_{k-1}) \tag{A.3}$$

Since $\psi_{k-1} = \epsilon + \psi_k$, equation A.3 can be written as

$$R' = R(i,k-1)[\cos(\epsilon) - \frac{\sin(\beta) \times \sin(\epsilon)}{\cos(\beta)}] \tag{A.4}$$

Also from the law of sines,

$$\sin(\epsilon) = f \times \cos(\beta)/R(i,k-1) \tag{A.5}$$

Using the identity $\cos^2 + \sin^2 = 1$ and equation A.5 we get

$$\cos(\epsilon) = \frac{\sqrt{R^2(i,k-1) - f^2 \times \cos^2(\psi_k)}}{R(i,k-1)} \tag{A.6}$$

Substitution of equation A.6 in A.4 leads to

$$R' = \sqrt{R^2(i,k-1) - f^2 \times \cos^2(\psi_k)} - f \times \sin(\psi_k) \tag{A.7}$$

Substituting equation A.7 into A.1

$$t_c(i,k) = R(i,k) - \sqrt{R^2(i,k-1) - f^2 \times \cos^2(\psi_k)} - f \times \sin(\psi_k) \tag{A.8}$$

Finally, by approximating

$$R^2(i,k-1) \cong f^2 \times \cos^2(\psi_k) \tag{A.9}$$

then

$$t_c(i,k) = [R(i,k) - R(i,k-1)] + f \times \sin(\psi_k) \tag{A.10}$$

Appendix B

Mechanical parameters of the cutter elements

The elemental stiffness on the cutter is derived based on the cantilever beam theory.

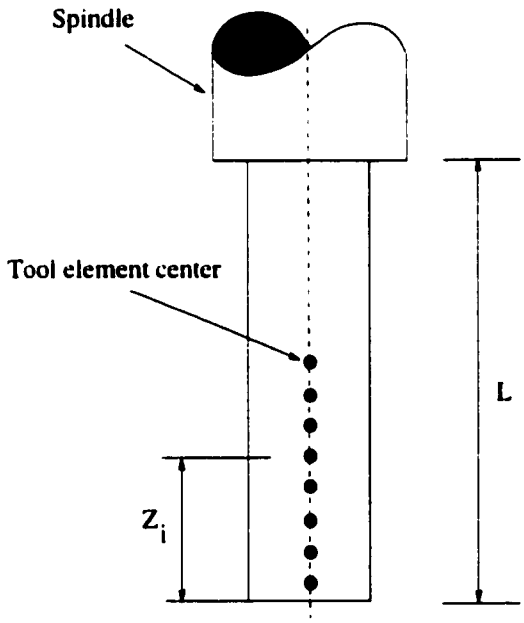


Figure B.1: Finite element representation of an end mill cutter.

The static deflection model was presented by Kline et. al. [KDS82] which assumed a rigidly clamped cutter deflecting due to the x and y cutting forces applied to the cutter at the x and y force centers. The deflection model is given by the following equation (see Figure B.1)

$$\delta^u = f^u / K^u \tag{B.1}$$

and

$$K^u = \frac{CF}{\left[< CF^u - Z >^3 - (L - Z)^3 + 3(L - Z)^2 (L - CF^u)\right]} \tag{B.2}$$

where u is any of directions x and y, L is the overhang length, δ^u, F^u, K^u and CF^u are the deflection, force, stiffness and force center in u direction, Z is the height of tool element from tip of the tool and C is the material and geometry constant. Knowing the experimental stiffness at the tip of the tool for given u direction, i.e. K^* at $Z = 0.0$, equation B.2 can be solved for constant C as follows

$$C = K^* \left[(CF^u)^3 - L^3 + 3L^2 (L - CF^u)\right] \tag{B.3}$$

Therefore, for any other element at height Z along the cutter axis, the stiffness can be determined using

$$k_{ii} = K^* \left[\frac{(CF^u)^3 - L^3 + 3L^2 (L - CF^u)}{< CF^u - Z >^3 - (L - Z)^3 + 3(L - Z)^2 (L - CF^u)}\right] \tag{B.4}$$

Appendix C

Fourth order Runge-Kutta scheme

In simulation of the cutter dynamic response to the cutting forces, the fourth order Runge-Kutta scheme is used. The system of equations for each tool element is given by equation 6.2

$$\begin{aligned} M_2\ddot{u}_2 + C_2\dot{u}_2 + K_2u_2 + K_1(u_2 - u_1) + C_1(\dot{u}_2 - \dot{u}_1) &= 0 \\ M_1\ddot{u}_1 + C_1(\dot{u}_1 - \dot{u}_2) + K_1(u_1 - u_2) &= F \end{aligned}$$

For such a system in each direction, say x, one may denote as follows

$$\begin{aligned} \eta_1 &= x_1 \\ \eta_2 &= \dot{x}_1 \\ \eta_3 &= x_2 \\ \eta_4 &= \dot{x}_2 \end{aligned}$$

Expressing the vibratory system in the state space form results in

$$
\begin{aligned}
\dot{\eta}_1 &= \eta_2 \\
\dot{\eta}_2 &= \frac{1}{M_1}[f - C_1(\eta_2 - \eta_4) - K_1(\eta_1 - \eta_3)] \\
\dot{\eta}_3 &= \eta_4 \\
\dot{\eta}_4 &= \frac{1}{M_2}[(C_2 + C_1)\eta_4 + (K_2 + K_1)\eta_3 - C_1\eta_2 - K_1\eta_1]
\end{aligned}
$$

The system of equations above can be expressed in the following vector format

$$\dot{\eta} = g(\eta, t) \tag{C.1}$$

Using equation C.1, one can apply fourth order Runge-Kutta method as follows

$$
\begin{aligned}
k_1 &= g(x_i, t_i) \\
k_2 &= g(x_i + k_1(h/2), t_i + h/2) \\
k_3 &= g(x_i + k_2(h/2), t_i + h/2) \\
k_4 &= g(x_i + k_2 h, t_i + h)
\end{aligned}
\tag{C.2}
$$

Having computed k's from the system of equations above, the state vector at time step $i + 1$ can be computed based on the state vector at previous time step i, as follows

$$\eta_{i+1} = \eta_i + (k_1 + 2k_2 + 2k_3 + k_4)h/2 \tag{C.3}$$

Note that first element in *vector* η_{i+1} in equation C.3 is the displacement of the cutter finite element and h is the integration time step.

Bibliography

[AAK87] M. Anjanappa, D.K. Anand, and J.A. Kirk. Identification and optimal control of thin rib machining. *ASME Winter Annual Meeting*, 6:272–278, 1987.

[Abr93] F. Abrari. *A regenerative dynamic force model for ball end milling.* M.Eng thesis, McMaster University, 1993.

[AE96] F. Abrari and M.A. Elbestawi. Closed form solution of ball and flat end milling. *Int. J. of Mach. Tool Des. and Res.*, 37(1):17–27, 1996.

[AES98] F. Abrari, M.A. Elbestawi, and A.D. Spence. On the dynamics of ball end milling, modeling of cutting forces and stability surfaces. *The International J. of Machine Tool and Manufacture*, 38(3):215–237, 1998.

[AMB92] Y. Altintas, D. Montgomery, and E. Budak. Dynamic peripheral milling of flexible structures. *Trans. ASME Journal of Eng. for Industry*, 114(2), 1992.

[Amr86] J. Amresh. Modern methods for automatic FEA mesh generation. *Proceedings of American Society of civil Engineers*, pages 19–28, 1986.

[AS91] Y. Altintas and A.D. Spence. End milling force algorithms for CAD systems. *CIRP Annals*, 40(1):31–34, 1991.

[BA95] E. Budak and Y. Altintas. Analytical prediction of stability lobes in milling. *Annals of the CIRP*, 44(1):357-362, 1995.

[BB73] W.R. Buell and B.A. Bush. Mesh generation a survey. *Trans. ASME the J. Eng. Ind.*, pages 332-338, Feb.1973.

[BH82] P.R. Brown and D.R. Hayhurst. The use of the Schwatz-Christoffle transformation in mesh generation for the solution of 2d problems. *ASME. Computers in Engineering*, pages 1-7, 1982.

[BS85] L.J. Buibas and J. Stolfi. Ruler,compass and computer: the design and analysis of geometric algorithms. *Theoretical foundations of computer graphics and CAD , in NATO ASI*, F 40:111-165, 1985.

[Cas82] M.V. Casey. A computational geometry for the blades and internal flow of centrifugal compressors. *ASME Paper 82-GT-155*, 1982.

[CFF85] J.C. Cavendish, D.A. Field, and W.H. Frey. An approach to automatic three dimensional finite element mesh generation. *Int. J. Numer. Meth. Eng.*, 21:329-347, 1985.

[Cha86] P.F. Chavez. Automatic mesh generation and optimization from the solid model database. *Proceedings of American Society of Civil Engineers*, pages 29-42, Oct. 1986.

[DK89] M. Dewaele and G.L. Kinzel. Speed improvements for NC program verification using solid modeling. *Proc. ASME Int'l computers in engineering conference and exposition, Anaheim C.A.*, 1(Anaheim, CA.):601-607, 1989.

[EMSE96] H. El Mounayri, A.D. Spence, and M.A. Elbestawi. Enhanced CAD/CAM for simulation and optimization of 3-5 axis milling of dies and molds. *CSME Forume, Hamilton, Ont.,Canada*, pages 394–401, 1996.

[Fau64] J. Faupel. *Engineering design.* Wiley, New York, 1964.

[FE92] B.K. Fussell and C. Ersoy. Computer generated CNC machining feedrates. *Proceedings 92 Japan USA symposium on flexible automation*, New York,NY,USA:377–384, 1992.

[FS97] R.V. Fleisig and A.D. Spence. Integrated digitizing, path planning, and five-axis machining for the refurbishment of compressor and turbine blades. *Manufacturing Science and Engineering*, pages 31–38, 1997.

[Gor83] W.J. Gordon. An operator calculus for surface and volume modeling. *IEEE Compt. Graph. and Appl.*, pages 18–22, Oct. 1983.

[Her76] L.R. Hermann. Laplacian isoparametric grid generation scheme. *J. of Eng. Mech. Div. proc. of Amr. Soc. civil Eng.*, 102(EM 5), 1976.

[Hig83] E.A. Highway. A mesh generator for automatically subdividing irregular polygons into quadrilaterals. *IEEE Trans. on. Mag.*, MAG-19(6):2535–2538, 1983.

[HL88] K. Ho-Le. Finite element mesh generation methods: a review and classification. *Computer Aided Design*, 20:27–38, 1988.

[HSA+81] R. Haber, M.S. Shepphard, J.F. Abel, R.H. Gallagher, and D.P. Greenberg. A general two-dimensional graphical finite element preprocessor utilizing discrete transfinite mapping. *Int. J. Numer. Meth. Eng.*, 17:1015–1044, 1981.

[Hum90] J.L. Humar. *Dynamics of structures.* Englewood Cliffs, N.J. : Prentice Hall, 1990.

[IE98] B.M. Imani and M.A. Elbestawi. Simulation methodology for die cavity machining. *Proceedings of CSME forume, Ryerson University, Toronto, Canada*, pages 194–201, May 19-22, 1998.

[Inc95] Spatial Technology Inc. *ACIS Geometric Modeler - Application Guide.* Spatial Technology Inc., 1995.

[JY89] S. Jain and K.C.H. Yang. A systematic force analysis of the milling operation. *Proc. ASME winter annual meeting, San Franscisco*, pages 55–63, 1989.

[KD83] W.A. Kline and R.E. DeVor. The effect of run-out on the cutting geometry and forces in end milling. *Int. J. Mach. Tool Des. Res.*, 23(2/3):123–140, 1983.

[KDS82] W.A. Kline, R.E. DeVor, and I.A. Shareef. The prediction of surface accuracy in milling. *Journal of Engineering for industry*, 104:272, 1982.

[KS66] F. Koenigsberger and A.J.P. Sabberwal. An investigation into the cutting force pulsation during milling operation. *Int. J. Mach. Tool Des. Res.*, 1:233–240, 1966.

[KT67] F. Koenigsberger and J. Tlusty. *Machine tool structures - Vol I: Stability against chatter.* Pergamon Press, 1967.

[KV90] L. Kops and D.T. Vo. Determination of the equivalent diameter of an end mill based on its compliance. *Annals of CIRP*, 39(1):93–96, 1990.

[LaC94] D.E. LaCourse. *Handbook of solid modeling.* McGraw-Hill, Inc., 1994.

[Liu95] X. Liu. Five-axis NC cylindrical milling of sculptured surfaces. *Computer Aided Design*, 27(12):887–894, 1995.

[Mar41] M.E. Martellotti. An analysis of milling process. *Trans. of ASME*, 36:667–700, 1941.

[Mar45] M.E. Martellotti. An analysis of milling process : Part II, down milling. *Trans. of ASME*, 74:233–240, 1945.

[Mer44] M.E. Merchant. Basic mechanics of metal cutting process. *Trans. ASME Journal of applied mechanics*, 66:A168–175, 1944.

[Mer65] H.E. Merritt. Theory of self excited machine tool chatter. *Trans ASME J. Eng. Ind.*, 87:447–454, 1965.

[Mon90] D. Montgomery. *Milling of flexible structures.* M.Eng thesis, University of British Columbia, 1990.

[MY93] I. Minis and R. Yanushevsky. A new theoretical approach for the prediction of machine tool chatter in milling. *Trans. ASME J. Eng. Ind.*, 115:1–8, 1993.

[NA76] K. Nakayama and M. Arai. On the storage of data on metal cutting forces. *Annals of the CIRP*, 25(1):13–19, 1976.

[Noo84] H. Nooshin. *Formex configuration processing in structural engineering.* Elsevier applied science publishers ltd., 1984.

[O'R94] J. O'Rourke. *Computational Geometry in C.* Cambridge University Press, 1994.

[PT97] L. Piegl and W. Tiller. *The NURBS book.* Springer, 2 edition, 1997.

[RB66] J.K. Russel and R.H. Brown. The measurement of the chip flow direction. *International Journal of Machine Tool Design and Research*, 6:129–138, 1966.

[REW89a] M. Rautenberg, A. Engeda, and W. Wittekindt. Mathematical formulation of blade surfaces in turbomachinery, part 1: Theoretical surface formulation. *ASME*. (89-GT-160):1–7, 1989.

[REW89b] M. Rautenberg, A. Engeda, and W. Wittekindt. Mathematical formulation of blade surfaces in turbomachinary, part 2: Practical examples of determined surfaces. *ASME*. (89-GT-161):1–9, 1989.

[Rub90] C. Rubenstein. The edge force components in oblique cutting. *Int. Journal of Machine tools Manufact.*, 30(1):141–149, 1990.

[SD86] J.W. Sutherland and R.E. DeVor. An improved method for cutting force and surface error prediction in flexible end milling system. *Journal of Engineering for industry*, 108(Nov.):269, 1986.

[SE90] R. Sagherian and M.A. Elbestawi. A simulation system for improving machining accuracy in milling. *Computers in industry*, 14(1):293–305, 1990.

[SHL68] R. Sridhar, R.E. Hohn, and G.W. Long. A stability algorithm for the general milling process. *Trans. ASME J. Eng. Ind.*, 90:330–334, 1968.

[SM73] D.J.L. Smith and H. Merryweather. The use of analytical surfaces for the design of centrifugal impellers. *Int. Journal for Numerical Methods in Engineering*, 7, 1973.

[ST93] S. Smith and J. Tlusty. Efficient simulation programs for chatter in milling. *Annals of the CIRP*, 42(1):463–466, 1993.

[Sta64] G.V. Stabler. The chip flow law and its consequence. *Advances in Machine Tool Design and Research*, pages 243–251, 1964.

[Tha80] W.C. Thacker. A brief review of techniques for generating irregular computational grid. *Int. J. Numer. Meth. Eng.*, 15:1335–1341, 1980.

[TI81] J. Tlusty and F. Ismail. The basic non-linearity in machining chatter. *Annals of the CIRP*, 30(1):299, 1981.

[TM40] S. Timoshenko and G.H. MacCullough. *Elements of Strength of Materials.* D. Van Nostrand Inc., 2 edition, 1940.

[Tob65] S.A. Tobias. *Machine tool vibration.* Blakie and Sons Ltd., 1965.

[TSW96] J. Tlusty, S. Smith, and W.R. Winfough. Techniques for the use of long slender end mills in high-speed milling. *Annals of the CIRP*, 1996.

[TTS89] S. Takata, M.D. Tsai, and T. Sata. A cutting simulation system for machinability evaluation using a workpiece model. *The annals of the CIRP*, 38(1):417–420, 1989.

[VP82] Van-Phai. Automatic mesh generation with tetrahedron elements. *International Journal for Numerical Methods in Engineering*, 18:273–289, 1982.

[Wan88] W.P. Wang. Solid modeling for optimizing metal removal of three dimensional end milling. *Journal of manufacturing systems*, 7(1):57–65, 1988.

[Whi86] R.C. Whitfield. *A mechanics of cutting approach for the prediction of forces and power in some commercial machining operations.* Ph.D. thesis, University of Melbourne, 1986.

[Yel85] I. Yellowley. Observations on the mean values of forces, torque and specific power in the peripheral milling process. *Int. J. Mach. Tool Des. Res.*, 25(4):337, 1985.

[YKSS91] K. Yamazaki, N. Kojima, C. Sakamoto, and T. Saito. Real time model reference adaptive control for 3d sculptured surface machining. *The annals of the CIRP*, 40(1):479–482, 1991.

[YP91] M. Yang and H. Park. The prediction of cutting force in ball end milling. *Int. J. Mach. Tools Manufact.*, 31(1):45–54, 1991.

[YS83] M.A. Yerry and M.S. Shephard. A modified quadtree approach to finite element mesh generation. *IEEE Comput. Graph and Appl.*, pages 39–46, 1983.

[YS84] M.A. Yerry and M.S. Shephard. Automatic three-dimensional mesh generation by the modified-octree technique. *International Journal for Numerical Methods in Engineering*, 20:1965–1990, 1984.

[YW83] Y.B. Yildir and A. Wexler. MANDAP-A FEM/BEM data preparation package. *IEEE Trans. Mag.*, MAG-19(6):2562–2565, Nov. 1983.

[Zad50] L.A. Zadeh. Frequency analysis of variable networks. *Proceedings of IRE*, 38:291–299, 1950.

[Zei91] I. Zeid. *CAD/CAM theory and practice.* McGraw Hill, Inc., 1991.

[Zor66] N.N Zorev. *Metal cutting mechanics.* Pergamon Press, 1966.

[ZP71] O.C. Zienkiewicz and D.V. Phillips. An automatic mesh generation scheme for plane and curved surfaces by isoparametric coordinates. *Int. J. Numer. Meth. Eng.*, 3:519–528, 1971.