GENERIC SOLID MODELLING BASED MACHINING PROCESS SIMULATION

By

HAZIM A. EL-MOUNAYRI, B.Sc. (Mechanical Eng.), M.Sc. (Materials Eng.)

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Doctorate of Philosophy

McMaster University

Generic Solid Modelling Based Machining Process Simulation

DOCTORATE OF PHILOSOPHY (1997) McMaster University

(Mechanical Engineering) Hamilton, Ontario

TITLE: Generic Solid Modelling Based Machining Process Simulation

AUTHOR: Hazim A. El-Mounayri

B.Sc. (Mechanical Eng.), M.Sc. (Materials Eng.)

SUPERVISORS: Dr. M. A. Elbestawi, Professor

Department of Mechanical Engineering

McMaster University

Dr. A. D. Spence, Assistant Professor

Department of Mechanical Engineering

McMaster University

NUMBER OF PAGES: xvi, 188

Abstract

In machining, the ability to automatically generate an optimum process plan is an essential step toward achieving automation, higher productivity, and better accuracy. These requirements are particularly emphasized in die and mold manufacturing, where complex tool and workpiece geometries involved make generation of the process plan a difficult task. High die production costs, narrow tolerance requirements, and the continuous demand for new components make process planning and NC code generation very complex and error-prone tasks. The current research need is to develop a system that is based on a simulation of the actual machining process, rather than simple geometric verification. Such a machining process simulator is needed to respond to the current need to enhance CAD/CAM technology with a machining process simulation that accounts for process mechanics and dynamics. A major impediment to implementing these systems is the lack of a general and accurate method for extracting the required geometric information.

In this thesis, a novel approach to perform this task is presented. It uses general and accurate representations of the part shape, removed material, cutting edge and rake face. Solid models are used to represent the part and removed volume, B-spline curves are used for the cutting edge representation and B-spline surfaces for the rake

face. Next, a generic solid modeler based milling process simulation system for 3-axis machining of complex parts using flat and ball end mills is implemented. It consists of geometric and physical simulators. In the geometric simulation, the tool swept volume is generated for every completed tool path (one NC block) and intersected with the part, yielding the corresponding removed material volume. The tool cutting edges are then intersected with that volume to produce the tool-part immersion geometry in the form of in-cut segments. These and an expression for the chip thickness are used to determine the chip load distribution along the cutting edge. The updated part can be used instead of the removed volume in the intersection step to generate the in-cut segments. The physical simulator models the mechanics and dynamics of the cutting process and uses the chip load to compute instantaneous cutting forces and predict other process parameters.

The milling process simulation is demonstrated and verified experimentally for 2 1/2- and 3-axis ball end milling. In addition, it is shown that geometric modelling of 4- and 5-axis milling using different tool shapes as well as other machining processes such as turning and drilling can be performed using the same approach.

Acknowledgements

I would like to express my deepest gratitude and thanks to my supervisor Dr. M. A. Elbestawi for his continuous guidance, moral and financial support throughout the course of this research. I would like also to thank my supervisory committee, especially my co-supervisor Dr. A. D. Spence without whom this work could not have been completed. I would like to extend my sincere gratitude and appreciation to my colleagues and the staff at the IMMRC, Mechanical Engineering Department and McMaster University for their encouragement and help at various occasions. These include Louise Perry, Betty Bedell-Ryc, Jane Mah, Jim McLaren, Joe Verhaeghe, Ron Lodewyks, Dave Schick and Allyson Wenzowski, just to mention a few. Special thanks are due to Behnam Imani, Derek Jenkins and Farid Abrari for their invaluable technical assistance at many different occasions. My best friends Dr. Tahany El-Wardany, Nasser Aly, Marwan Hassan, Nael Barakat, Hossam Kishawy and Mohsen Ghaemian deserve special thanks for their various supportive roles. Last but not least, it is the trust, love, expectations and prayers of my family (my mother and three brothers Mohamed, Khaled and Omar) that actually kept me going in this long journey, and supplied me with hope and strength to achieve this objective.

I dedicate this work to them and to the memory of my father who taught me the

merit and value of education. God bless his soul and reward him.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Conventional machining practice

Conventional machining practice suffers from a number of drawbacks and limitations. First, manual tool path generation is very error prone. Next, manually selecting the cutting conditions from machining handbooks is based on experience or trial and error. It also results in conservative values being selected. Finally, the machining of complex surfaces with conventional machining is both uneconomical and inaccurate.

1.2 Enhanced machining technology

1.2.1 Enhanced machine tools (NC and CNC machine tools)

Significant improvements in machining were realized with the introduction of NC (Numerically Controlled) and CNC (Computer Numerically Controlled) machine tools that continue to replace conventional machines. Their main advantages consist of an

increased flexibility, a better accuracy of the machined part and a more efficient operation, thus a shorter production cycle. In addition, CNC made possible the cutting of complex parts as well as 4- and 5- axis machining.

1.2.2 CAD/CAM

The complexity of some of the cutting operations, (e.g. 3-5 axis ball end milling of dies and molds with sculptured surfaces) as well as the need for higher accuracy and reliability, have dictated the use of the CAM technology in planning and generating the tool path. For similar and other reasons, the CAD technology was introduced in the design phase, and thus the introduction of CAD/CAM systems. The latter have established themselves in the manufacturing industry and have become an essential tool in the machining sector, especially where higher productivity, improved reliability, and better accuracy are sought. CAD/CAM systems also contribute to the integration of design and manufacturing and thus are key to automation. In particular, solid model based systems are very suitable in an untended or automated environment as the solid model representation can produce, in principle, geometric infohining process

The geometric simulation of the machining process can serve two purposes:

1. Part updating, a basic operation for tool path verification.

2. Chip load extraction, a basic information for predicting process parameters.

The first objective can be achieved via two geometric procedures: 1) Regularized set-union operator and 2) Regularized set-difference operator. The geometric entities

1.3 Current limitations

1.3.1 Existing CAD/CAM systems

The role of CAD/CAM systems in achieving a reliable, accurate and efficient machining operation is still, however, limited. First, they lack step by step and/or complete NC geometric verification. Second, they rely on manual scheduling of the cutting conditions (i.e. user input from machining data handbooks). Because cutting conditions (e.g. feed rate) are not related to chip load, the maximum metal removal rate that satisfies the physical constraints is not used. Next, they lack geometric procedures to extract the critical geometric information needed for machining process simulation. In fact, though solid modeling has established itself as the geometric representation in many of today's commercial CAD/CAM systems, it still does not support directly some of the manufacturing applications, including machining process simulation. Finally, existing CAD/CAM systems do not support force models for different machining operations required to predict the physical parameters (e.g. cutting forces) and automatically schedule reliable or optimum cutting conditions.

1.3.2 Geometric simulation of machining process

The geometric simulation of the machining process can serve two purposes:

1. Part updating, a basic operation for tool path verification.

2. Chip load extraction, a basic information for predicting process parameters.

The first objective can be achieved via two geometric procedures: 1) Regularized set-union operator and 2) Regularized set-difference operator. The geometric entities

involved in this case are: 1) The part and 2) The tool swept volume. The second objective is more involved and the extraction of the chip load distribution along the cutting edge requires geometric representations of: 1) The updated part and 2) The cutting edge. The latter can be modeled precisely as a 3D solid. However, it is usually sufficient to model it using a combination of surfaces and curves. For example, a sharp edge can be represented in terms of two surfaces (rake and clearance faces) and one edge (where the faces meet). The geometric algorithms required to extract the chip load distribution depends on the type of representation used and the level of accuracy sought. Procedures for extracting this information for the general 3-5 axis machining of free-form parts do not exist yet. The current work addresses this issue and presents a methodology and an implementation that deal with that limitation.

1.3.3 Physical simulation of machining process

The objective of this simulation is to allow for a control of the machining process by predicting the different process parameters which are then checked against the constraints. In its most general form, the modeling would include:

1. Chip load
2. Cutting process mechanics (force modelling)
3. Workpiece dynamics
4. Machine tool dynamics

The complexity and importance of each of the above aspects depend on the particular machining operation. Also, the accuracy and scope of application of the simulation

will depend on the completeness of the modeling. For example, if chatter is to be predicted, then the most flexible modes of the machine tool being used have to be considered. In the case of end milling, most of the simulations available in the open literature are oversimplified and either cannot compute instantaneous cutting forces or unable to model the dynamic characteristics of the process. This is particularly true for the case of ball end milling.

The current work implements accurate force models to predict instantaneous cutting forces from which other parameters can be computed.

1.3.4 On-line adaptive control

A current practice is to use CAM softwares to generate the tool path and then rely on on-line adaptive control of the machining process during actual production. However, this approach suffers from a number of difficulties. As observed by Tlusty and MacNeil [58], the transfer function between the table velocity f (feedrate) and the cutting force F is a part of the AC (Adaptive control) feedback loop. In particular, it is the dynamics of the transfer function F/f which affects the behavior of the system. There are three aspects of this dynamics: 1) The variation of the cutting force in a steady state of constant depth of cut; 2) The variation of the cutting forces in the transients; and 3) The time delay included in the transfer function. In end milling (which is the focus of the current work), the force responds to a change in feedrate with a time delay that may cause instability in the adaptive control loop. More generally, on-line adaptive controllers fail to deal with process transients characterized by sudden jumps in force magnitude. In addition, they rely heavily on laboratory sensing devices that are impractical for industrial application.

1.4 Proposed solution

1.4.1 Process control through machining simulation

Process modeling and simulation presents itself as a more reliable alternative for process control (compared to on-line adaptive control). This simulation is twofold. The geometric part of the modeling must accurately compute the required geometric information (i.e. cutter-part intersection geometry). The physical simulation is basically a force model that relies on the geometric data to determine the instantaneous cutting forces, torque and deflections and to predict other process parameters such as chatter.

1.4.2 Implementing the machining process simulation using CAD/CAM technology as the platform

Being a very powerful tool in design and manufacturing that is widely used in industry, CAD/CAM systems stand as a very suitable platform for developing new applications. Here, the particular application of interest is machining. On the other hand, such a development can be seen as a current demand to fill the existing gap between CAD/CAM technology and the machining process simulation. A closer look at CAD and CAM tools is needed to determine the requirements of such a development.

CAD tools can be seen as the intersection of three sets [67]: Geometric modeling; 2) Computer graphics; and 3) Design tools. This means that any CAD application involves procedures from those three "packages". If any of the required procedures is unavailable, the application is also unavailable. Similarly, CAM tools can be defined

as the intersection of three sets: 1) CAD tools; 2) Networking concepts; and 3) Manufacturing tools. Again any CAM application will rely on procedures from all three sets. The above suggests that CAD/CAM tools can be defined as shown in Figure 1.1. Based on this definition, one can now identify the algorithms required to be developed for machining modeling and simulation to fall in two categories (or sets): 1) Geometric modeling and 2) Manufacturing. In the current thesis, the first part of the development handles the geometric aspect of the machining operation while the second deals with the physical aspect of the operation.

1.4.3 Using the machining process simulation

An important use of a machining process simulator is to verify NC codes, both geometrically and physically, before actual machining. During that verification phase, the cutting conditions can be rescheduled to maximize the metal removal rate subject to physical constraints (machine, tool, and part constraints). This approach is more efficient and safer than trial and error on (expensive) NC (or CNC) machines. Also, the resulting NC code is more reliable and efficient, and thus contributes towards higher productivity and sustained untended machining operation. Finally, this simulation can be part of an automated environment. If there is already a CAD/CAM system running in such environment, then it can be used as the platform for integrating the process simulator.

1.5 Thesis overview

The ultimate objective of the current work is to enhance existing CAD/CAM technology with machining process simulation. A CAD/CAM system enhanced with a machining process simulator is refered to as: Enhanced CAD/CAM system or Process-based CAD/CAD system. The objective is achieved in three steps. First, a novel and generic approach for simulating the geometric aspect of the machining operation is introduced (Chapter 3). Next, it is implemented using GWB (Geometric Workbench), a polyhedral solid modeller, as the platform (Chapter 4). The implementation is verified for a typical 2 1/2 axis die semi-finishing using ball end mills. Finally, the Process-based CAD/CAM system is presented and the components required to enhance a conventional CAD/CAM system are developed (Chapter 5), namely a geometric simulator and a physical simulator. These two components form what is called here a Geometric based machining process simulator. The resulting system currently supports full 3-axis end milling using flat and ball end tools.

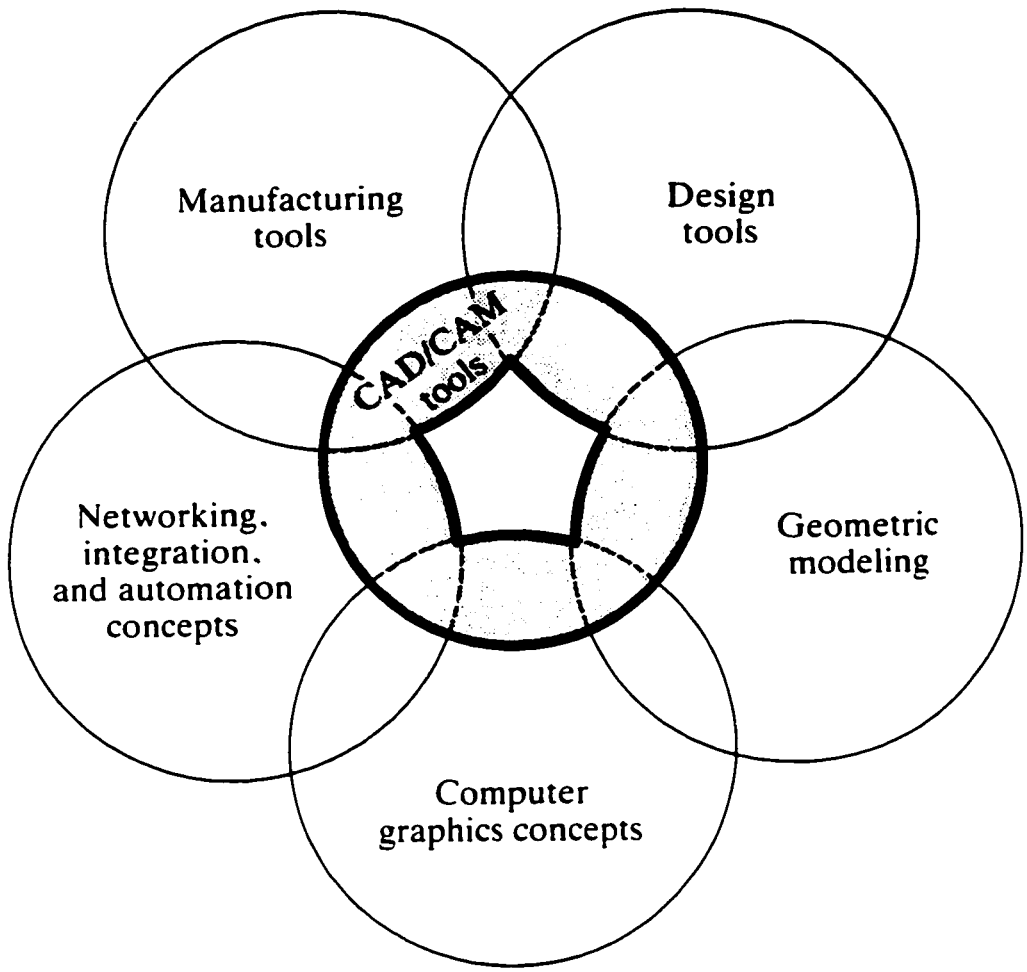


Figure 1.1: Definition of CAD/CAM tools based on their constituents (Figure 1-10 [67])

Chapter 2

Literature Review

2.1 Machining process simulation

This simulation is twofold. It consists of geometric and physical aspects. This distinction flows naturally from the need for different tools to handle separately the modelling requirements and complexities encountered with each aspect. The first aspect includes all the geometric calulations involved in determining the geometric information required by the physical model. The second aspect refers to the mechanics of the machining process. It consists of force models that uses the geometric data to compute cutting forces and other process parameters, and automatically reschedule the feed to maximize the metal removal rate.

The underlying metal cutting theory for force prediction and the existing modeling approaches are first covered. Next, the state of the art of geometric modeling is presented. This is needed for the following section to identify the extent to which such a (geometric) tool has been exploited in the context of machining process simulation

and in particular milling. The last section concludes the chapter by summarizing the current limitations and consequently the objective of this work.

2.2 Physical modelling of the cutting process

2.2.1 Theory of metal cutting

In general, cutting operations (such as milling) involve two or more cutting edges inclined at various angles to the direction of cut. However, the simplest case consists of one edge perpendicular to the cutting velocity (between tool and workpiece). This type of cutting configuration is referred to as *orthogonal.* On the other hand, if the cutting edge is inclined with respect to the normal to the cutting speed, then the cutting condition is called *oblique.* The latter is geometrically more complex than orthogonal cutting.

The metal cutting process was found to be basically a plastic-flow process. However, the nature of the deformation zone is not unique. For this reason, some researchers modelled it as a thin-zone while others adopted a thick-zone model. It was reported that the available experimental evidence indicates that the thick-zone model may describe the cutting process at very low speed, but at higher speeds, most evidences indicate that a thin shear plane is approached.

One of the most recognized analysis is by Merchant [41] who adopted the thin-shear-plane model and derived analytical equations for cutting force computation. The model is based on the following assumptions:

- Tool tip is sharp

- The deformation is two-dimensional (i.e. there is no side spread)
- The stress on the shear plane is uniformily distributed
- The resultant force R on the chip applied at the shear plane is equal, opposite and colinear to the resultant force applied to the chip at the tool-chip interface.
- No built-up-edge

R is then decomposed into two components (Figure 2.1). The first (F_S) is in the direction of shear plane and thus can be expressed as a function of the shear strength which is a material property and the shear area (the product of chip thickness and chip width). The second (N_S) is along the normal to the shear plane. The key issue is then to determine or estimate the direction of the shear plane (i.e. the shear angle ϕ), the shear strength, and the direction of the resultant force with respect to the rake face β (also referred to as the friction angle between the tool and chip material).

Experiments carried out by Merchant and others have shown that the above model qualitatively describes the cutting process satisfactorily. On the other hand, quantitatively, the shear-angle relationship has been found inaccurate. Also, the shear strength and coefficient of friction values are higher than those established by conventional tensile and friction tests.

There are other analytical models but none can precisely predict conditions in practical cutting situations. For this reason, researchers of the machining process had to resort to other approaches for force prediction.

2.2.2 Other approaches to physical modelling

There are basically two other approaches used to model cutting forces in metal cutting applications. These are:

- Semi-empirical
- Mechanistic

The semi-empirical approach consists of approximating oblique cutting with orthogonal cutting and deriving empirical equations for τ_s, ϕ and β or equivalent angles. This approach was used by a number of researchers where a fairly good agreement between the predicted cutting forces and test results was reported (e.g. Yang and Park [65]). Because empirical expressions for the fundamental parameters (i.e. τ_s, ϕ, and β) are derived, semi-empirical models can be applied to solve various cutting processes.

The mechanistic approach is the one mostly used in the literature. The basic idea is to ignore completely the analytical equations and to write the force components in terms of two quantities. The first quantity embeds all the mechanics of the process (Q_1) while the second one is purely geometric (Q_2):

$$F_i = Q_1 \cdot Q_2 \tag{2.1}$$

where F_i is a cutting force component. The physical quantity is determined through experimental calibration while the geometric quantity requires a geometric tool for its extraction. The calibration is valid for a certain cutting condition and a certain tool/workpiece combination. Empirical models supply information for specific cutting operations and usually consider the influence of practical rather than fundamental

variables. For this reason, it is difficult to apply data from one cutting process to another. Generally, force prediction using mechanistic models is the most accurate.

Scope and accuracy of mechanistic force models

Smith and Tlusty [50] grouped the mechanistic force models found in the literature in five categories based on the level of accuracy and scope of application they allow for. The five types can be described in terms of the process parameters they can predict as follows:

- Rigid average forces and static tool deflections.
- Rigid instantaneous cutting forces.
- Rigid instantaneous cutting forces and static tool deflections.
- Flexible instantaneous cutting forces and static tool deflections.
- Dynamic instantaneous cutting forces, dynamic deflections, accurate surface finish, and chatter.

The first type uses the Metal Removal Rate (MRR) and specific power (P_{sp}) to compute the average power:

$$P = P_{sp} \cdot MRR \tag{2.2}$$

The average torque, average cutting force and corresponding cutter deflection can then be estimated, as follows:

$$\begin{aligned} T &= \frac{P}{\omega} \\ F_c &= \frac{P}{v} \end{aligned} \tag{2.3}$$

$$F_s = 0.5 \cdot F_c$$
$$\delta = \frac{F_s}{K}$$

where v is the peripheral cutting speed ($v = \pi \cdot D \cdot N$), N is the spindle speed, D is the tool diameter, δ is the tool deflection at the tool tip, K is the tool stiffness at the tool tip, F_c is the average tangential cutting force and F_S is the side thrust force. This model can be used basically to check average power and torque against the maximum values of the machine tool being used. Its prediction of cutting forces and corresponding deflection are very approximate and thus unreliable.

The only difference between the second and third type of mechanistic force models is that the latter goes one step further and computes the tool deflection using the instantaneous cutting force. The basic difference between these two models and the first one is that they predict instantaneous (as opposed to average) cutting forces. Here, the instantaneous force is written in terms of the chip thickness (or chip load). Because the chip thickness varies with the angular position, the total cutting force applied to one cutting edge is computed as the summation of incremental forces, each applied to an infinitesmal element of a certain width (dz). The incremental force applied to element i of a cutting edge is computed as follows:

$$dF_{t,i} = K_t \cdot dz \cdot f_i \tag{2.4}$$
$$dF_{r,i} = K_r \cdot dF_{t,i}$$

where $dF_{t,i}$ and $dF_{r,i}$ are the tangential and radial components of the incremental cutting force applied to segment i of a cutting edge, f_i is the local chip thickness, K_t is the specific cutting force, and K_r is the ratio between the radial and tangential components of the force. By adding the incremental forces applied to an edge, the

total force on that cutting edge is obtained. Next, by summing up the total edge forces of all the edges of the cutter, the total instantaneous cutting force is found. The latter can then be used to predict the tool deflection and surface finish.

The fourth type of mechanistic models is able to predict more accurate instantaneous forces by closing the loop between the cutting force and resulting tool deflection. First, the instantaneous cutting force is computed as described above and used to compute the tool deflection as follows:

$$\delta_1 = \frac{F_1}{K} \tag{2.5}$$

A more accurate value of the chip thickness for the first pass can be computed as:

$$f_1 = f - \delta_1 \tag{2.6}$$

For the next pass (i.e. the next cutting edge), the tool deflection is:

$$\delta_2 = \frac{F_2}{K} \tag{2.7}$$

The chip thickness expression now has to be modified to account for both the current tool deflection (δ_2) and the material (undulation) left behind from the previous pass. Thus, it is calculated as:

$$f_2 = f - \delta_2 + \delta_1 \tag{2.8}$$

For a tool with multiple cutting edges, the actual chip thickness f_a is computed as:

$$f_a = f - \delta + u \tag{2.9}$$

where f is the static chip thickness, δ is the current tool deflection, and u is the surface undulation from the passage of previous cutting edges. Using the above expression

for the chip thickness results in more accurate estimation of the instantaneous cutting forces (compared to the previous models). Finally, the dynamics of the cutting process can be incorporated in the force model leading to the last and most accurate mechanistic force model, namely the Dynamic regenerative model. The latter uses the same form for computing the chip thickness (i.e. equation 2.9) but with tool deflections and surface undulations that are estimated from dynamic deflections. The model is able to predict dynamic forces, dynamic tool deflection as well as the onset of chatter. Basically, the simulation runs in a series of small time steps (dt). At each instant, the force on each cutting edge involved in the cut is computed, and the forces are summed vectorially. The accelerations are computed in all degrees of freedom based on the force, and these accelerations are integrated twice to produce displacements of the tool. That is, for any coordinate:

$$\begin{aligned} F_i &= m \cdot \ddot{\eta}_i + c \cdot \dot{\eta}_i + k \cdot \eta_i \\ \ddot{\eta}_{i+1} &= \frac{F_i - c \cdot \dot{\eta}_i - k \cdot \eta_i}{m} \\ \dot{\eta}_{i+1} &= \dot{\eta}_i + \ddot{\eta}_i \cdot dt \\ \eta_{i+1} &= \eta_i + \dot{\eta}_i \cdot dt \end{aligned} \tag{2.10}$$

where η refers to either x or y and i is the time step. Additionally, at each time step, the position of any cutting edge in contact with the workpiece is stored as a record of the surface left behind. This information is then used to determine the value of u (surface undulation). More details can be found in chapter 4.

2.2.3 Process constraints

There are several constraints on the machining operation and this requires proper scheduling of the cutting conditions for the particular tool, workpiece and machine tool being used. First, instantaneous maximum and average torque and power must be limited to the capabilities of the machine (basically, spindle shaft and motors). Similarly with the machine tool maximum feed and speed. Next, the instantaneous cutting force should be kept below a certain value to prevent the breakage of the tool shank or cutting edge(s). On the other hand, it may be required to set the cutting conditions in a way to reduce tool-workpiece deflections and the resulting dimensional or surface error. The latter is another constraint that controls the surface finish. Indeed, the deflection of the tool leaves location errors on the finished part surface. Finally, tool wear is another constraint which dictates the tool life. Useful cutting tool life depends on many factors including tool and part material, cutting velocity and feedrate. However, for a given tool, the parameter which mostly affects the tool life is the cutting speed. For this reason, feed is the cutting parameter usually rescheduled when optimizing the process.

2.2.4 Machine tool dynamics

The role of machine tool dynamics on the cutting forces (or cutting process) can be represented as shown in Figure 2.2(a), where:

- The cutting process includes the transfer function which relates the exciting vibration to the cutting forces. The force model must be able to simulate the dynamic deflection and include their effect as well as the effect of surface

undulations in the force computation

- The vibratory system of the machine tool models the latter as an n degrees of freedom mass-damper-spring system. This vibratory model must be designed for each machine tool. The required information (i.e. modal parameters, mode shapes and natural frequencies) can be extracted from the frequency response at the cutting edge measured using, for example, the impact test.

As far as the tool deflection is concerned, the effect of machine tool dynamics can be represented as shown in Figure 2.2(b). However, a more realistic loop should also include the secondary feedback (i.e. surface undulations or waviness); Figure 2.2(c) represents this, where T is the tooth period and f_{act} is given by:

$$f_{act} = f - \delta(t) + u(t - T) \tag{2.11}$$

where δ is the tool deflection and u is the surface undulations. It is such a model which is capable of predicting chatter. Actually, it will be used to determine the stability condition beyond which chatter is initiated. Because the most critical parameter in initiating chatter is chip width (b), the stability limit is expressed in the form:

$$b \leq b_{lim} \tag{2.12}$$

where b_{lim} is the chip width above which chatter occurs.

Theoretically speaking, the dynamics of the whole machine (i.e. spindle, overhang, table, and frame) contributes to the dynamic deflection of the cutting edge and thus to the value of the chip thickness. However, practically, there are some modes which are more significant and relevant than others (e.g. modes that are related to the spindle and overhang), as far as the simulation of the cutting forces is concerned.

2.3 Geometric modelling

2.3.1 Curve (Wireframe) modelling

Overview/Intoduction

Parametric as well as non-parametric curve representations are found in the literature. Parametric representations have shown more advantages than non-parametric ones, especially when used in the context of Engineering applications, including CAD/CAM.

Curves are also divided into analytic and synthetic. The latter can represent more general and complex shapes. Synthetic curves when formulated in a certain way are very practical in and suitable for design and reverse design. The need for synthetic curves in design arises in two occasions. First, when a curve is represented by a collection of measured data points. Second, when an existing curve must change to meet new design requirements [67].

Mathematically, synthetic curves represent a curve fitting problem to construct a smooth curve that interpolates or approximates a given set of data points. Polynomials are most suitable for this task. They are easy to store, differentiate and integrate, and allow for a very practical and powerful curve representation and design tool to be implemented (as shall be seen in the sequel). Bezier and B-spline curves are the two most widely used formulations for generating synthetic curves as polynomials or piecewise continuous polynomials. They have been adopted as the standard of CAD/CAM technology and are now supported by major CAD/CAM and solid modelling packages.

Fundamentally, B-spline curves are the natural extension and generalization of Bezier ones. Indeed, they consist of piecewise continuous Bezier curves. Different degrees of continuity can be produced at the junction points. C^1- continuity is the minimum acceptable order for curves used in engineering applications. However, C^2- cubic B-splines are most widely used as they allow for true 3D curves (i.e. non-planar or twisted curves). Rational B-splines and Non-uniform rational B-splines (NURBS) are the more general and flexible extension of B-splines curves.

Linear interpolation: the basic procedure

The inherent polynomial nature of Bezier, B-spline and NURBS curves can be seen in the derivation of these curves from a sequence of linear interpolations.

- Parametric representation of a line

Let P_0 and P_1 be two points in E^3 and $u \in R$, then $P_0^1(u)$ given by

$$P_0^1(u) = (1-u) \cdot P_0 + u \cdot P_1 \quad 0 \leq u \leq 1 \tag{2.13}$$

describes the linear segment between P_0 and P_1

- Higher order polynomials

Above, we have seen that a set of two points is all what is needed to represent and generate a linear segment. The operation is very simple as well as attractive from storage point of view. Because these two points control the resulting (linear) curve, they are usually called control points.

Now to produce a second order polynomial, all what is needed is a third point (or control point). A point on this parabola is generated through two consecutive linear

interpolations (i.e. a two-level linear interpolation) as follows:

$$\begin{aligned} P_0^1(u) &= (1-u)\cdot P_0 + u\cdot P_1 \\ P_1^1(u) &= (1-u)\cdot P_1 + u\cdot P_2 \end{aligned} \tag{2.14}$$

By linearly interpolating between $P_0^1(u)$ and $P_1^1(u)$, we get:

$$P_0^2 = (1-u)\cdot P_0^1(u) + u\cdot P_1^1(u) \tag{2.15}$$

$P_0^2(u)$ is a point on the parabola (see Figure 2.3). This same procedure can generate a polynomial of order n. The beauty of this approach to curve representation and generation lies mainly in the following:

1- Simplicity

2- The interplay and relation between the shape of the control polygon (polygon made of the linear segments connecting the control points) and the resulting curve.

3- The efficient storage and transformation of the curve as it can be done using the control points only. In fact, this is a direct consequence of the property of invariance under affine maps that Bezier curves enjoy. Because most of the transformations used in CAD environment are affine maps (e.g. rotation and translation), there are a number of advantages in using a Bezier representation in the current work. For example, to translate or rotate a cutting edge (represented as a Bezier curve), one needs only to transform the control points then generate the points on the curve (i.e. the cutting edge at the new position). This is of course much more efficient than having to transform all the points on the curve or to generate a new one.

4- The provision of a very powerful and practical design tool. This is because a Bezier curve mimics the shape of its control polygon. Also, not only the original

control points control the shape of the resulting curve but the latter can be modified by changing the location of one or more of the control points; in other words, it lends itself beautifully to the iterative design procedure. A special design case would be a curve whose polynomial equation is known. For that particular case, one can derive the control polygon which yields an identical curve. Then, modifying the original curve can be done through the control points.

5- Bezier and B-spline curves and their rational versions are numerically the most stable among polynomial bases currently used in CAD systems [20]. They lend themselves best to computer implementation and application.

The de Casteljau algorithm generates points on an n^{th} order Bezier curve by performing n linear interpolations between $n+1$ control points. The de Boor algorithm is its equivalent but for generating B-spline curves [20].

2.3.2 Surface modelling

Parametric curves are the mapping of a one dimensional parametric space (e.g. in u) to the 3D Euclidean (or Cartesian) space. Parametric surfaces are the mapping of a two dimensional parametric space (e.g. in u and v) to the 3D Euclidian space. Based on this, surfaces can be considered to be the extension of curves. Also, most of the basic concepts, categorization and properties presented in the above section about curves, apply to surfaces. Surfaces are used to model geometric objects, including surface and solid models. Surface modeling describes objects precisely and accurately. The information that a surface model is able to provide allows for many engineering applications to be implemented (e.g. generating NC tool paths for continuous path machining). An important disadvantage of these models is that they

are still ambiguous in some applications.

Surface representation

The choice of the surface form depends upon the application. In addition, all surface forms must be easy to differentiate to determine surface tangents, normals, and curvatures. Next, surfaces should not exhibit unwanted oscillations or be expensive storagewise. This makes cubic polynomial preferred for most (practical) surface applications. Furthermore, the surface mathematical form must be applicable to both surface design and surface representation. The first involves using key given data and making interactive changes to obtain the desired surface. The second involves the use of given data to display and view the surface. Finally, in many of today's mechanical parts, complex (e.g. free form) shapes are encountered. Thus, the surface representation used must allow for the modeling of such shapes. This is achieved by allowing the representation of synthetic surfaces.

Existing CAD/CAM systems provide the designer with both analytic and synthetic surface entities. Parametric representation is being used for its many advantages over the non-parametric one.

Definition of parametric representation of a surface

The parametric representation of a surface means a continuous, vector-valued function $P(u,v)$ of two variables or parameters, u and v, where the variables are allowed to range over some connected region of the $uv-$ plane and as these variables assume values over that region, $P(u,v)$ assumes every position on the surface. The function $P(u,v)$ at certain u and v values is the point on the surface at these values; this can

be expressed as follows:

$$\begin{aligned} P(u,v) &= [x(u,v)\ y(u,v)\ z(u,v)]^T \\ u &\in [u_{min}, u_{max}] \\ v &\in [v_{min}, v_{max}] \end{aligned} \tag{2.16}$$

For most surfaces, the u and v intervals are $[0,1]$. This equation suggests that a general 3D surface can be modeled by dividing it into an assembly of topological patches. This surface is then referred to as composite surface. A patch is defined as the basic mathematical element to model a composite surface. It can be triangular or rectangular.

Parametric representation of analytical surfaces

Analytical surfaces are based on wireframe entities and include plane surface, ruled surface, surface of revolution and tabulated cylinder. The simplest analytical surface is the plane. It can be defined by three points, one point and two direction vector or one point and one normal direction. The parametric equation can be written as follows:

$$\begin{aligned} P(u,v) &= P_0 + u \cdot (P_1 - P_0) + v \cdot (P_2 - P_0) \\ u &\in [0,1] \\ v &\in [0,1] \end{aligned} \tag{2.17}$$

where P_0 is the point defined at $u = 0$ and $v = 0$, P_1 and P_2 are the other two points defining the plane. The tangent vectors at point P are:

$$P_u = \frac{\partial P}{\partial u} = P_1 - P_0 = constant \tag{2.18}$$

$$P_v = \frac{\partial P}{\partial v} = P_2 - P_0 = constant$$

The surface normal at point P is:

$$n(u,v) = \frac{P_u \times P_v}{|P_u \times P_v|} = constant \tag{2.19}$$

The curvature of a plane surface is zero. A database structure of this type of surfaces would typically include the unit normal to the plane and a point in it.

Details about the representations of other analytical surface can be found in a number of references in the literature (e.g. [67]).

Parametric representation of synthetic surfaces

Synthetic surfaces are formed from a given set of data points or curves and include mainly Bicubic, Bezier, B-spline and Coons patches, all of which are based on polynomial forms. They provide designers with better surface design tools than analytic ones. There are few methods to generate this type of surfaces, namely: 1) Tensor product, 2) Rational method, and 3) Blending method. Tensor product method is the most popular one and is widely used in surface modeling. Its wide spread use is largely due to its simple separable nature involving only products of univariable basis functions, usually polynomial. It introduces no new conceptual complications due to the higher dimensionality of a surface over a curve.

A tensor product Bezier surface is an extension of the Bezier curve in two parametric directions u and v, where an orderly set of data or control points is used to build a topologically rectangular surface. The surface equation can be written by

extending the curve equation, that is:

$$\begin{aligned} P(u,v) &= \sum_{i=0}^{n}\sum_{j=0}^{m} P_{ij} \cdot B_{i,n}(u) \cdot B_{j,m}(v) \\ u &\in [0,1] \\ v &\in [0,1] \end{aligned} \tag{2.20}$$

where $P(u,v)$ is a point on the surface, $B_{i,n}$ are the Bernstein functions in the $u-$ direction, $B_{j,m}$ are the Bernstein functions in the $v-$ direction, and P_{ij} are the points which form the vertices of the control or the characteristic polyhedron of the resulting Bezier surface. The points are arranged in an $(n+1)\times(m+1)$ rectangular array. The properties of the Bezier surface is similar to those of the Bezier curve. Their main disadvantage is the lack of local control. For this reason, designers resort to B-spline surfaces.

A tensor product B-spline surface is an extension of the B-spline curve in two parametric directions u and v. A B-spline surface can approximate or interpolate the vertices of the control polyhedron. The properties of B-spline surfaces are the same as B-spline curves. The details of the formulation of this type and other surface patches can be found in a number of references in the literature (e.g. [67]).

The rational parametric surface is considered a very general surface. The shape of the surface is controlled by weights (scalar factors) assigned to each control point. Rational Bezier and B-spline surfaces are available in major CAD/CAM systems. The rational B-spline is considered a unified representation that can define a variety of surfaces. The premise is that it can represent any surface.

Any of the available surface patches is seldom enough by itself to model complex

surfaces typically encountered in design and manufacturing applications. These complex surfaces are known as sculptured or free-form surfaces. They arise extensively in many industries, including automotive and die/mold making.

A sculptured surface is defined as a collection or sum of interconnected and bounded parametric patches together with blending and interpolation formulas. The surface must be susceptible to some machining language (e.g. APT) processing for NC machine tools. From a modeling point of view, a sculptured surface can be divided into the proper patches which can be created to provide a C^0 or C^1 continuous surface.

2.3.3 Solid modelling

A solid model is an unambiguous (complete) representation and therefore can provide enough information to implement, in principle, any geometric procedure. There are many fundamental issues related to the design of a solid modeling representation and more generally, a solid modeler. The interested reader can refer to a number of references such as [47], [8] and [39]. The current thesis uses existing solid modelers to implement a machining process simulation and is therefore faced with the issue of selecting an appropriate representational scheme and a suitable solid modeler.

A three-level view of solid modelling

The formalization of the solid modelling theory is based on a distinction of three different levels of modeling (Figure ??):

1. The physical object which is the actual object in 3D real world.

2. The mathematical object which is a suitable idealization of the real 3D physical object.

3. The representation which is assigned to the mathematical object to allow for computer manipulation

The idea here is to design an idealized representation of the actual object in a way that will allow for the analysis of the properties (of the idealized object). The results obtained from this analysis provides the required information on the actual real object [39].

Solid (mathematical) models

In geometric modelling, E^3 subsets are the abstract geometric entities (models) used to model physical objects. However, only very few subsets of E^3 are adequate models of physical solids. They are the ones which capture mathematically the following properties: 1) Rigidity; 2) Homogeneous three dimensionality; 3) Finiteness; 4) Closure under rigid motion and regularized Boolean operation; and 5) Boundary determinism [47].

The mathematical implications of the above properties mean that valid solid models are bounded, closed, regular, and semi-analytic subsets of E^3. These subsets are called regularized sets (or r-sets).

Two approaches for characterizing a solid were adopted to develop mathematical models that satisfy the above requirements:

1. Surface-based characterization which yields surface-based models

2. Point-set characterization which yields point-set models

In the first type, a solid is defined as a bounded, closed subset of the 3D Euclidean space. In the second, a solid is described by its bounding surfaces. In order to be a definition of r-sets, surfaces must satisfy particular properties; basically: 1) Closed; 2) Orientable; 3) Non self-intersecting; 4) Bounding; and 5) Connected [8]. Some of these properties can be enforced just by structural means (i.e. through the design of the data structure). For example, the first property can be enforced by demanding that each edge occurs in exactly two faces (in the case of manifold solids). Other properties requires special checking of the data structure which are usually computationally expensive or can be achieved by limiting the user to validity-enforcing description mechanisms [39].

Representational schemes

The successful representation of solid models in computers and their utilization in engineering applications depend not only on their properties but on the properties of the schemes representing them as well. A representation scheme is defined as a relation that maps a valid point set into a valid model.

The formal properties of a representational scheme which determine its usefulness in geometric modeling are: 1) Domain; 2) Validity; 3) Completeness (unambiguousness); 4) Uniqueness. Other (informal) properties are: 1) Conciseness; 2) Ease of creation; and 3) Efficacy in the context of applications. The last property is crucial in the current work which consists of developing new applications using solid modelers as the geometric engine.

A number of representational schemes have been designed with the above properties in consideration. The two most popular schemes are B-rep (Boundary representation) and CSG (Constructive Solid Geometry). CSG models are in general the most concise but are procedural (i.e. unevaluated). Moreover, the directly available methods for the generation of graphical output or the creation of data for numerical algorithms are slow. Their expressive power depends on the class of half-spaces available; in general, it can not be extended to cover surface patches. B-rep models are explicit (i.e. evaluated) and thus are directly useful for graphical applications. The modeling space of boundary models depends on the selection of surfaces that can be used. In general, the boundary models can be used to represent objects from more general modeling space than that possible for CSG. On the other hand, boundary models of useful objects may become large, especially if curved objects are approximated with polyhedral models. In addition, algorithms based on boundary models become quite difficult if representation of objects from more general modeling spaces then polyhedral must be processed [39].

Existing solid modelers

The B-rep method is widely used in commercial CAD systems such as CATIA. Some of the commercial systems use an implicit CSG history tree to store the design steps and explicit B-rep for actual solid representation and interrogation.

As mentioned above, the types of surfaces supported by a given B-rep solid modeller governs the kind of shapes it can represent as well as the accuracy of the representation. For example, in GWB, Geometric Workbench [39], solids are represented as polyhedrals which means that only planar surfaces are used. Theoretically, any

shape can be accurately represented if enough facets are used. However, the representation is not exact. Besides, a complex part shape (e.g. die or mold with free-form shape) would require a significant number of facets for an accurate representation to be achieved. Working with such a representation can be (very) expensive computationally and storage-wise. Alternatively, more general surface representations can be used to model the geometry of complex parts [15]. NURBS are widely used for this purpose and have become the standard in CAD/CAM systems. One of the main reasons is that they offer a common mathematical form for representing and designing both standard analytical shapes and free-form curves and surfaces. Consequently, both analytical and free-form shapes are represented precisely and a unified database can store both [45]. Existing solid modelers do not support explicit geometric procedures for general and comprehensive machining process simulation.

2.4 Milling process simulation

The central quantity upon which models for predicting cutting forces in milling are based is the chip load on the cutter. The fundamental expression used to evaluate the chip load is that of the chip thickness. In fact, Martelloti [40] derived formulae for the average undeformed chip thickness (used in the calculation of the metal removal rate) as well as instantaneous undeformed chip thickness (used in more accurate force calculations). The author has shown that the tooth path is trochoidal but can be approximated very reasonably as an arc of circle since in milling the feedrate is small relative to the rotational speed of the cutter. Based on this approximation, he derived

a simple equation for the instantaneous chip thickness:

$$t(\theta) = f \cdot \sin(\theta) \tag{2.21}$$

where f is the feed/tooth and θ is the tool (or tooth) angular position. The fact that the chip thickness depends on the angular position makes the geometric simulation of the milling process an involved task whose most basic requirement is the determination of angle θ (and thus the instantaneous chip thickness).

The need to predict cutting forces for the purpose of process planning has motivated several studies of the cutting mechanism and the force system. Following Martelloti [40], Tlusty and MacNeil [58] made another major contribution towards a fundamental understanding of the end milling process. They developed a mechanistic model to explain the force variations of flat end milling process at constant radial and axial depths of cut as well as in transient cuts such as when the cutter is entering the workpiece. This work showed that the force predictions (in the case of flat end mill) can be made by considering the tangential cutting force to be proportional to the chip load and the radial force to be proportional to the tangential force:

$$\begin{aligned} F_t &= K_t \cdot a \cdot t_c \\ F_r &= K_r \cdot F_t \end{aligned} \tag{2.22}$$

where F_t is the tangential force, F_r is the radial force, K_t and K_r are empirical constants (derived experimentally), a is the axial depth of cut and t_c is the chip thickness. These two equations form the basis of many of the mechanistic models developed later.

In the case of helical cutters, the forces given in the above equation are replaced by incremental ones acting on elemental disks of the cutter (Figure 2.5 [33]). The axial

depth of cut is substituted by the thickness of the elemental disk. The total force on the cutter is then calculated by summing all forces acting on the individual elements (disks). Tlusty and MacNeil performed a closed form integration of the incremental forces to get the resultant forces in the x- and y- directions.

Later, Kline and co-workers used the above equations and developed a mechanistic model for prediction of cutting forces in flat end milling and applied it to the studies of cornering cuts, cutter deflection, surface error generation and the effect of tool runout on cutting forces ([12], [13], [32], and [35]). Contrary to Tlusty and MacNeil, they performed a numerical integration to compute the total forces experienced by the cutter. This approach allows for more complex expressions of forces and practical factors such as cutter runout and tilt to be handled.

The model of Kline et al [33] is best described as a mechanistic model conditioned by experimental average force data. The authors have shown that the parameters K_t and K_r vary with cutting conditions and cut geometry. Thus, they must be determined for each combination.

The above work was mainly motivated by the need to predict end milling forces for the purpose of process planning, including surface accuracy and tool breakage considerations. Another group of researchers concentrated on other aspects of the milling process such as vibration and chatter (Andrew and Tobias [5]; Sridhar et al [54]; Tlusty and Ismail [57]; and Tlusty and Smith [59]). In general, they have studied the dynamic behavior of the components of the chatter loop, including the milling process and the machine tool structural compliance. For example, Tlusty and Ismail used the chip load model of Kline and Devor to simulate chatter. They modified the expression of the tangential force to account for the effects of dynamic tool deflection

and surface waviness on the instantaneous chip thickness. The resulting expression can be written as:

$$F_t = K_s \cdot f \cdot (h - z + z_{min}) \tag{2.23}$$

where h is the current chip thickness, z is the current deflection of the cutter normal to the surface of the cut and z_{min} is the deflection of the tooth which previously generated the surface (lowest value from previous passes). F_x and F_y are computed from F_t then substituted in the vibration equations to determine the tool deflections in x and y directions and hence z at each time increment (More details are given in Chapter 4).

Fussel and Srinivasan [22] proposed a model of the end milling process which can be used in a constraint-type adaptive control system. The force model incorporated into the system is one by Kline and Devor because it uses K_t and K_r which are function of chip thickness. Unlike Kline and Devor, the authors here explicitly consider the effect of cutter and workpiece compliance on the uncut chip geometry.

In all of the above works, one can notice the following: Only flat end mill is considered. The cut geometry is simple (e.g. full-immersion). Geometric parameters are given not extracted. Only 2 1/2 Dimensional parts are considered and only 2 1/2-axis milling is simulated. Also, no attempt is made to integrate the force model with a CAD (or CAD/CAM) system.

This limitation in the capability of the milling process is mainly due to the lack of a powerfull geometric tool that can handle the complex geometries involved in machining, especially in 3-5 axis milling of general 3D part shapes. For this reason, it was only possible to overcome this bottleneck by introducing solid modelling. Wang [63] recognized the fact that the cut geometry in many operations is more complicated

than the ones simulated by the existing models. In addition, this geometry is changing from one path to the other and within the same path. Therefore, he proposed the use of solid modelling to assist in the extraction of the geometric information required for computing the milling forces. The objective of Wang was a real time solid modelling based computer simulation of end milling for optimizing the MRR through adaptive feedrate control. The system works as an off-line adaptive controller before NC codes are downloaded to the CNC controller. It produces feedrate which maximizes the productivity without violating constraints. Average force is evaluated using the removed material volume extracted by the solid model. This is based on the assumption that average forces are proportional to the metal removal rate. Next, the feedrate is automatically adjusted to increase the productivity under some constraints. In this way, the author avoided the time consuming and complicated process of extracting the geometric information required to compute the instantaneous cutting forces. In addition, because the system was to run real-time, the expensive 3D Boolean operation was reduced to one dimensional by using the z-buffer technique (Roth [49]). The system was the first to simulate 3-5 axis milling of complex parts and achieved 30% savings in the machining time by scheduling the maximum safe feedrate. By using the removed volume, it was possible to calculate the cutting forces for the case of ball end milling.

In summary, in this work, the requirements of a complex milling operation were reduced to the modelling of swept volume and efficient Boolean operation. It was possible for the first time to simulate complex 5-axis milling of parts with sculptured surfaces with different tool shapes in real time. However, the model is approximate and no optimization is possible within an NC block.

The computational problem of solid modelling appears to be an important concern for other researchers too who resorted to one or more of the following:

- To introduce some approximations in one of the following: The force model, the type of geometric data that can be extracted, or the technique used in extracting the geometric data.
- To limit the complexity of the simulation by considering one or more of the following: 2 1/2- axis milling, 2 1/2 D part shapes, Flat end mill, or straight cuts.
- Use computational techniques to improve the efficiency.

Takata et al [55] used the more accurate mechanistic model developed by Kline and Devor. They were interested in the evaluation of machinability of products in the environment of computer integrated manufacturing. The application consisted of the simulation of cutting forces and machining error induced by tool deflection. The authors underlined the fact that the cutting simulation (to be used in such environment) should not be a simple tool path verification system but rather an evaluation tool of the cutting process. Such a cutting simulation system would include functions of evaluating the physical process parameters such as cutting force, generated heat, machining error and tool wear in addition to the function of geometric simulation. The system which consisted of a geometric simulator and a physical simulator was limited to flat end milling. In addition, the experimental verification did not involve the machining of complex geometries. Takata and co-workers used the contact area between the tool swept volume and the workpiece to extract the cutter immersion geometry and axial depth. This is possible only if each NC block is divided into much

smaller segments (e.g. 1 mm) before the tool swept volume is computed. Similar to Wang, they used the z-buffer technique to reduce the computational time of the Boolean operation. Beside the approximation and limitations involved in the one dimensional Boolean operation, the value of resolution used (which is an important issue in the geometric verification of machined parts) is below satisfactory considering the required accuracy of modern parts.

Armarago and Deshpande [6] have recognized the need for arriving at a comprehensive and reliable machining performance prediction system which is required in modern manufacturing industry. They developed an end milling module for inclusion in a CAD/CAM system. Only flat end milling was considered and no application to complex machining was reported.

Yamazaki et al [64] proposed two counter measures to avoid problems that can occur due to inaccurate manual NC programing, and to produce a highly reliable and productive machining operation. The first measure consists of the pre-detection of undesirable cutting operation in an on-line or off-line manner. The second measure is a real-time modification of machining including cutting conditions. The authors developed a Dynamic Machining Simulator for on-line and real time control of cutting conditions. The latter uses a surface model and updates the workpiece, calculates the material removal and displays a 3D view of the workpiece, all in real time. A grid point technique was used to evaluate the volume removed and correspondingly the resulting shape. This information could only be used in the computation of average cutting forces. Also, the accuracy of the simulation is very dependent on the resolution of the mesh.

The work by Kendall et al [31] is along the same line of that by Takata et al, but

recognizes the importance of developing a methodical system. This system consists of an object-based end mill engagement extraction system for surface machining using a 2 1/2-axis CNC milling machine. The cutter-part intersection is extracted from a 3D Boundary representation of the workpiece geometry for the programmed tool trajectory as specified by the CNC instruction set. The current implementation covers only linear cuts with a flat end mill tool.

The long term objective of Altintas and co-workers consists of the development of an integrated solid modeller based process simulation, planning, monitoring and control system to improve the productivity and accuracy of computer assisted milling operation. However, in their recent work ([51] and [3]), they mainly concentrate on the integration of cutting mechanics with the CAD environment. Such an integrated system can predict the cutting forces, torques and deflections in advance and automatically schedule the maximum safe feedrate before machining takes place. In addition, the programmed cutter trajectory can be improved. The use of such a system can be extended to assist in on-line control and monitoring operations. The resulting multi-functional integrated system is centered around a solid modeler which supplies the data required by any of these applications. The main concern of the authors was accuracy and consequently, they neither used the approximate MRR force model [63] or an approximate representation of the cutter-part intersection geometry [55]. In order to keep an efficient simulation, they had to resort to the following measures. First, only 2 1/2-axis milling with flat end cutters is simulated and part shapes were limited to 2 1/2D. Consequently, the axial depth was constant and known and the only parameters required to be extracted are the entry and exit angles of the tool with the workpiece (i.e. cutter immersion interval). This greatly simplified the cutter-part

immersion calculation which is reduced to 2D curve/curve intersection. Second, an analytical force model was developed from which the extreme values can be directly extracted. This eliminates the need to repeatedly sample at small angular increments as the cutter rotates and thus improves the solution efficiency. Next, they used techniques from computational geometry to improve the efficiency of the extraction of the cutter-part intersection geometry. Finally, using a CSG representation, the intersection of the milling cutter is separately analyzed with each block and cylinder primitive. Individual results are then combined by applying the same Boolean operations and tree used in the overall part definition. This reduces the complexity of the immersion calculations with general part shapes.

The objective of the work by Fussell and Ersoy [23] is very similar to those by Wang, Takata and Yamazaki. It consists of the development of a computer system for automatic generation of feedrates for improving the efficiency of CNC machining of complex sculptured parts using end milling operation. However, they stressed the importance of an extremely accurate as well as a fast simulation. On the one hand, the geometric modelling algorithms must maintain an extremely high degree of accuracy for industries in which the ratio of the nominal part dimensions to the allowable tolerance variation is often greater than (10,000:1). On the other hand, the geometric simulation must be fast as the number of tool movements in a NC program can exceed 10,000 for a sculptured surface machining. The authors then identified the main reason behind computationally expensive geometric algorithms to be the surface/surface intersection operation. Consequently, they avoided 3D Boolean operation by keeping only a solid representation of the tool swept volume and using a surface representation of the part which is converted to a vector-point representation

before it is intersected with the swept volume. The intersection operation produces the removed volume, cutter entrance and exit angles, and axial depth of cut for each tool movement. After breaking the workpiece into many discrete vectors, for each small tool movement the intersection of the resulting tool envelope with the block vectors is found. Then, by assigning a surface area to each vector, the volume removed by the tool movement can be calculated by summing the product of the vector area with the length of the vector removed by the tool. The first step in the extraction of the other geometric parameters is to keep track of the surface point vectors that are cut by the tool; then, the position of the cut vectors relative to the cutter is then used to find the entrance and exit angles. The latters are obtained from the intersection of the tool envelope and the maximum and minimum y coordinates of the surface point set vectors. The determination of the axial depth of cut is a worse case scenario where the longest vector removed during the cutting step is used as an approximation. The authors proposed the use of the MRR force model for rough cutting and the more accurate mechanistic model for finishing. The system is currently capable of generating tool paths for 3-axis cutting only. However, only experimental verification of 2D slot cuts was reported.

Force models for ball end mills have been developed only recently because of the high complexity involved in their modelling. Yang and Park [65] have analyzed the ball end milling process and developed a semi-empirical force model that can predict accurate instantaneous cutting forces.

Yücesan and Altintaş [66] have developed a semi-mechanistic analytical force model which is based on general mechanisms of chip formation and tool flank-workpiece contact. The mechanisms are correlated to the cutting forces with experimentally

identified pressure and friction coeficients on the rack and flank contact surfaces.

Feng and Menq [21] developed a rigid force model for ball end milling similar to that developed by Kline and Devor [33] for flat end milling. This model was then used by Lim et al [38] to predict the dimensional errors caused by tool deflection when machining sculptured surfaces using ball end mills. In this case, the empirical constants K_t and K_r are function of z, the axial depth. Also, they derived a more general expression for the chip thickness in the case of ball end milling:

$$t(\theta, z) = R_2(z) - R_1(z) + f_H \cdot \sin(\theta) \tag{2.24}$$

where $t(\theta, z)$ is the undeformed radial chip thickness, z is the axial depth, $R_1(z)$ and $R_2(z)$ are the local radii at $z = z_1$ and $z = z_2$ respectively. The authors expressed the need to avoid the computationally expensive solid modelling techniques or the approximate z-buffer technique for the extraction of cutter-part intersection geometry. Their approach was to use the cutter geometry and the feed direction to derive a mathematical expression for the entry angle. In this way, only the exit angle needs to be extracted using a solid modeller. However, they derived such an analytical expression for slotting only. Also, the system was not used in simulating complex machining.

2.5 Summary

In summary, one can identify three major phases of milling process simulation. The first phase (e.g. Koenigsberger and Sabberwal [36]; Tlusty and MacNeil [58]; Kline [34]; and Fussell and Srinivasan [22]) is characterized by the following. Using the basic formulae of the undeformed chip thickness (Martelloti [40]), and given all the cutting

conditions parameters including the entry and exit angles and the axial depth of cut, it is possible to compute the cutting forces for the case of flat end mill. The simulation can cover cutters with straight teeth as well as helical ones. These works suffered from limitations due to the lack of a powerful geometric tool that can handle the complex geometries encountered in modelling the milling process. Only flat end mills were modelled, and the tool path and part geometry were very simple. Geometric parameters such as tooth exit and entry angles were supplied rather than extracted. No attempt was made to integrate the force model with CAD/CAM modules, which is essential to achieve an industrially applicable solution.

The second phase of milling force simulation is characterized by the introduction of solid modelling to simulate the machining process based on the idea by Voelcker and Hunt [62]. The research, however, was restricted to the geometric aspect, and the new modelling tool was used for NC geometric verification only. To avoid the computationally expensive direct Boolean operations, different researchers resorted to alternative approaches such as image space Boolean operation based approach (Van Hook [61]; Atherton et al, [7]) and surface based approach (Chappel [11]; Oliver and Goodman [43]; Anderson [4]; Jerrard et al [29]). The next phase is characterized by extending the geometric modelling based simulation to cover the physical aspect. Three different approaches can be identified. Two were directly inherited from the previous phase (i.e. image space and surface based approaches). The third one is referred to as purely solid modelling approach. The basic difference between the different approaches is how the Boolean operation is performed. The first approach, which encompasses the work of Wang [63], Takata et al [55], Park et al [44], and Yamazaki et al [64], uses the ray casting technique to reduce the 3D Boolean operation

to one dimension. The second approach which includes basically the work by Fussell and Ersoy [23] uses a point vector representation of the part to replace the Boolean intersection with a line/surface intersection. The last approach is characterized by the use of direct Boolean operation. This includes the work by Dewaele and Kinzel [14] and Spence and Altintaş [53].

In none of the above works was a general approach reported. The techniques were designed and developed for modelling and simulating only the milling process. Their simulation capabilties are restricted. The work in the first two categories is, in addition, limited in accuracy. The main reason is that methods similar to the ones originally designed and implemented for NC geometric verification (i.e. image space Boolean operation and surface based approaches) were used for NC process simulation. Those techniques were, however, unable to accurately supply the information required for computing different process parameters. The reason behind their use was to avoid the computationally expensive direct Boolean operation, in an attempt to implement a fast simulation. In many of today's industries, however, accuracy is a major concern and thus should not be sacrified for computational speed. Besides, the absolute computational time of computer algorithms in general is decreasing everyday with the very fast development of computer technology. The basic requirement of an accurate and complete simulation was recognized by Spence and Altintas [53]; however, their work was restricted to 2 1/2-axis milling. The current work adopts a similar view but implements an approach that is general in all respects. It can represent any part or tool shape, any cutting edge geometries and model any machining process. The objective in the current work is thus to implement a technique for a general and accurate NC simulation of the machining process.

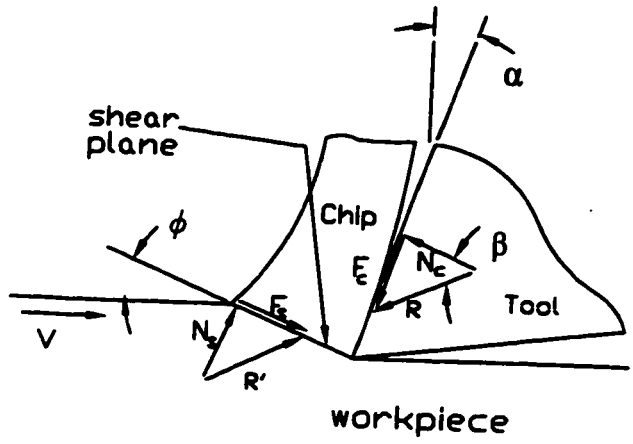


Figure 2.1: Resolving resultant force into components

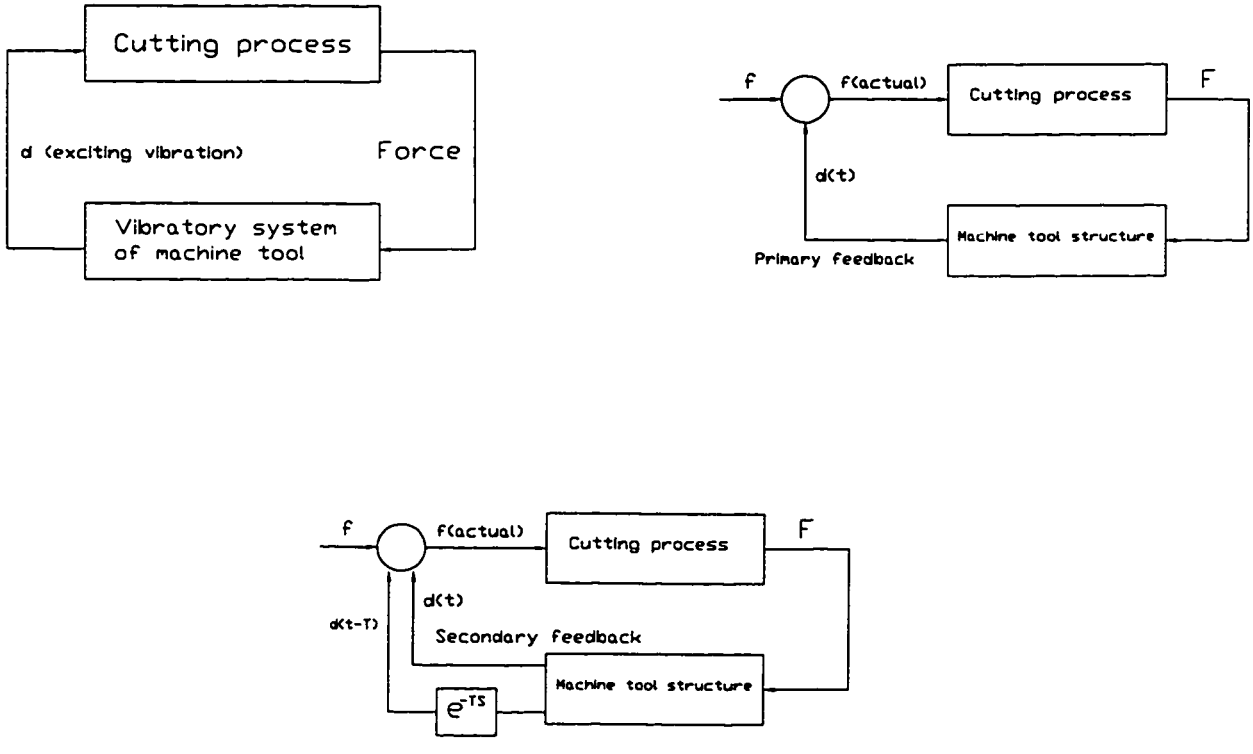


Figure 2.2: (a) Role of machine tool dynamics on the cutting forces; (b)Effect of machine tool dynamics on tool deflection; (c) Effect of machine tool dynamics on tool deflection: A more realistic loop

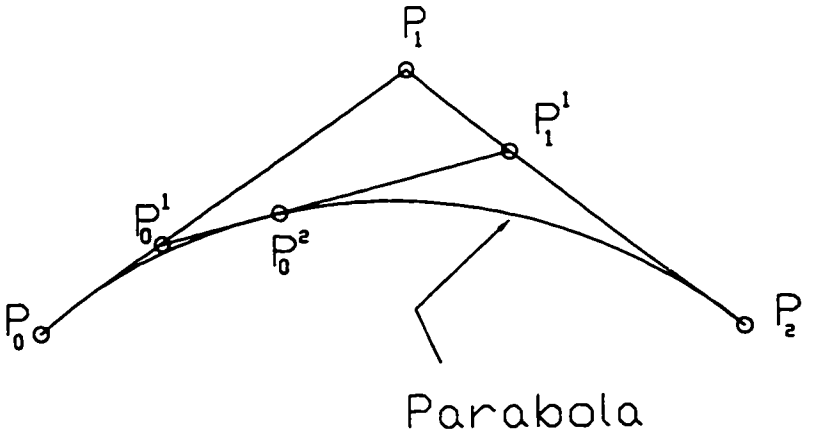


Figure 2.3: Generating a parabola through two consecutive linear interpolations

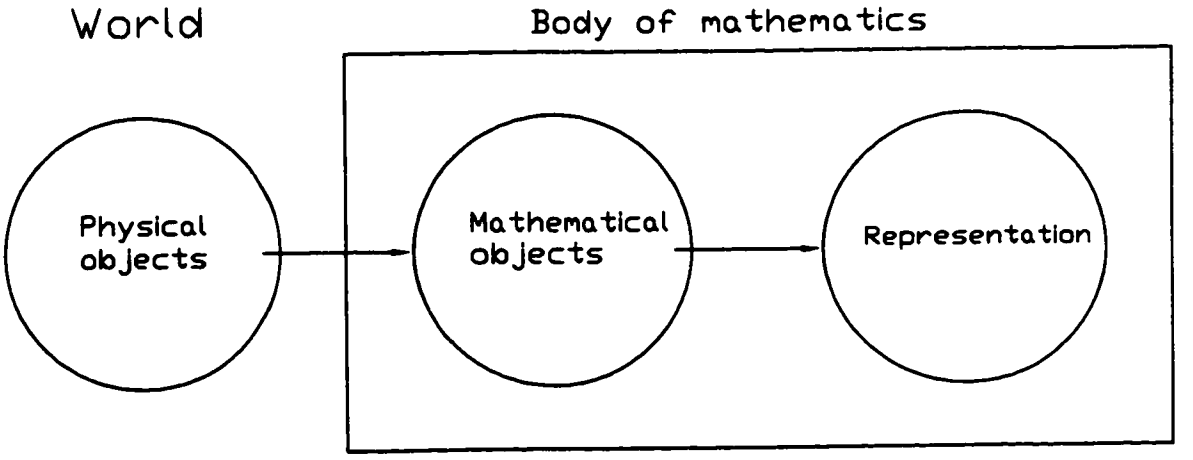


Figure 2.4: A three-level view of Solid modeling

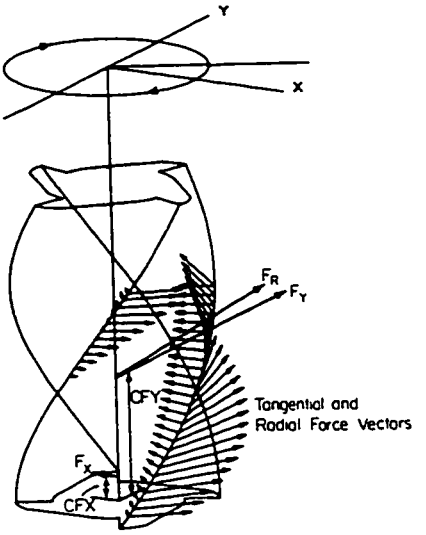


Figure 2.5: Cutting force distribution in case of helical end mills

Chapter 3

Methodology

3.1 Identification of basic geometric entities

A comprehensive geometric simulation of the general machining operation requires the representation of the following basic geometric entities:

- Part geometry
- Tool swept volume
- Cutting edge geometry
- Rake face geometry

The part geometry and tool swept volume are enough to produce the material removed during machining by using the appropriate Boolean operation, as shown in Figure 3.1 (Voelcker and Hunt [62]). However, this information would only allow for average forces to be computed (Wang [63]). On the other hand, to compute the instantaneous cutting forces and other process parameters, the chip load needs to be extracted. This

can only be done if the geometric simulation goes one step further to capture the cutting edge status during machining, thus, requiring a representation of the cutting edge geometry (see Figure 3.2). In addition, a representation of the rake face of the cutting edge is needed to extract the distribution of the rake angle and/or the normal to the rake surface at the edge. This information is needed to determine the direction of the instantaneous differential cutting forces and to model the mechanics of cutting locally.

3.2 Identification of required geometric information

The information required to compute the instantaneous chip load consists of:

- In-cut segments of the cutting edge(s)
- Tool angular position
- Chip thickness (Undeformed radial chip thickness)

The cutting edge has in general the shape of a wedge. When it is cutting, the resulting instantaneous differential force and its direction depend on the corresponding local geometry of the cutting edge and its orientation with respect to the material being machined; more specifically, they are function of:

- the inclination of the cutting edge with respect to the direction of the movement of the cutting tool, defined as the inclination angle.

- the inclination of the cutting edge relative to the material being machined, defined as the rake angle (see Figure 3.3).

For empirical and semi-empirical force models (used here), the variation of the local tool inclination needs to be extracted. In another word, the distribution of the rake angle is a required geometric information.

Finally, information about the final part (or its surface) is needed to compare it with the desired shape (i.e the design). This information would consist of points generated on the surface of the resulting part.

3.3 Basic idea of the geometric modelling approach

The geometric modelling approach proposed here integrates curve, surface, and solid modelling. The basic idea is to simulate the actual machining process at the level of the geometric modelling in a way that:

- allows any geometric entity encountered in the machining process to be accurately represented in its most general form
- makes the critical geometric information (required for predicting different process parameters) a natural output of the simulation

Table 3.1 describes the methodology proposed here. First, the different entities that are encountered in any machining process (e.g. workpiece, removed material, machined workpiece, cutting edge, and rake face) are modelled at the level of the geometric simulation as one of three representations: 1) Solid, 2) Surface or 3) Curve.

The next step is to simulate the machining and extract the required geometric information. The simulation of machining runs as follows. For every completed tool path (e.g. a block in the NC code), the tool swept volume is generated and intersected with the part, yielding the corresponding removed material volume. The part is updated by subtracting the swept volume from the part. The extraction of the in-cut segments proceeds as follows. First, the in-cut segments of the engaged cutting edge(s), are the natural output of the intersection and classification of (a) space curve(s) (the cutting edge(s)) and a solid model (the updated part or removed material). Second, the rake angle distribution is extracted from the distribution of the normal to the surface representing the rake face.

This methodical approach makes any extension a natural and systematic task. For example, adding a new tool shape would require, from the geometric point of view, the addition of an algorithm to generate the corresponding swept volume as well as the fitting of B-spline curve(s) to the cutting edge(s).

It should be noted that the modeling of the machining process as approached here is finite (not differential) in that a tool path is modeled as non-infinitesimal directed and connected segments of space curves rather than as differential elements.

3.4 Generic geometric modelling approach

3.4.1 Solid modelling for representing removed material volume and updated part

Solid modelling offers generality, accuracy and completeness in 3D part representation. This makes it appropriate for machining process simulation where general part shapes that need to be accurately represented are encountered. As presented in Chapter 2, a solid model has a number of important characteristics. First, it is informationally complete and thus allows, in principle, for any geometric procedure to be implemented. The geometric modeling approach proposed here involves the following procedures:

- Generation of the in-process part shape
- Generation of swept volumes for different tool shapes and motions
- Extraction of the in-cut segments of the engaged cutting edges during machining

Second, it allows for and is a key in the integration of the different activities involved in the product cycle (from design to manufacturing). More generally, it is a key to automation.

One of the most widely used and recognized solid modelling representational schemes is B-rep (Boundary representation). A B-rep which supports general surfaces such as B-splines and NURBS is suitable for representing and designing free-form shapes which are becoming more common nowadays, especially in the automotive and die and mold industries.

3.4.2 Surface modelling for representing rake face and part surface

In practical machining cases, surface representations might need to be generated from existing shapes. In addition, some of the geometric information required for the physical or geometric simulation might need to be extracted experimentally. In particular, a distribution of the rake angle (or the normal to the rake face) along the cutting edge is needed to accurately compute the components of the cutting force experienced at the rake face. This information can be extracted from the surface representation of the latter.

3.4.3 Curve modelling for representing cutting edge

In order to allow for a general approach, a technique for representing any shape of cutting edge is required. Moreover, because of the multi-axis nature of machining processes, it must be possible to efficiently represent the edge at any location in space.The current work proposes the use of 3D cubic polynomial curves for the representation. The latter is implemented in the form of B-spline curves which are characterized of being general and computationally efficient. In addition, they are numerically the most stable among polynomial bases used in CAD systems (Farin [20]).There are several reasons for restricting to polynomial of order 3. It is the minimum order that yields non-planar curves. Polynomial curves of higher orders are computationally expensive and tend to oscillate about control points.

The modelling of a cutting edge using the current approach involves three basic steps: 1) Representation; 2) Fitting; and 3) Generation. The first step consists of

providing a symbolic, mathematical representation of the cutting edge. It can either be exact, or a set of measured points. The next step consists of fitting a B-spline curve to the cutting edge. The final step in the modelling consists of generating the cutting edge at any location in the 3D space. Since B-spline curves are invariant under affine transformations, such generation is performed by translating and/or rotating the control points only. This is far more efficient than transforming all the points of the curve. From a storage point of view, only the control points need to be stored because they carry complete (unambiguous) information about the corresponding polynomial curve.

3.4.4 Geometric procedure for extracting the critical geometric information

The critical geometric information to be extracted consist of the in-cut segments of the engaged cutting edge(s) and rake angle distribution (or distribution of the normal to the rake surface).

Extraction of in-cut segments

This operation falls under the more general class of problems known as set membership classification. The classification of a candidate set X with respect to a reference set S is a segmentation of X into three subsets (Requicha and Voelcker [46]):

1. Subset of X inside S
2. Subset of X on the boundary of S
3. Subset of X outside S

In case the reference set is a solid represented by its boundary (i.e. as a B-rep model) and the candidate set is a curve, the basic algorithm intersects the curve with the faces of the solid and determines the type of transition at each intersection point by examining the point's neighborhood (Requicha and Voelcker [46]). The classification status may or may not change at intersection points but only changes at such points.

Previous work on curve/solid classification (e.g. [46]; [56]; Roth [49]) was limited to only line segments. Here, curves are extended to include cubic polynomials in Bezier and B-spline forms.

There are two possible ways of approaching the extraction procedure that produces the in-cut segments of the cutting edge(s):

- As a Face/Edge intersection and classification
- As a Solid/Edge intersection and classification

Face/Edge approach Here, the first step consists of intersecting the tool geometry with the part shape to produce the contact face (between the tool and the part). Next, each cutting edge of the tool is intersected with the contact face, or more exactly, the boundary of this face. In fact, the boundary of this contact face includes all the geometric information required for computing 2- and 3- axis instantaneous in-cut segments.

The intersection can be done in 2D or 3D. In the first case, the procedure consists of mapping the 3D cutting edge and contact face representations onto a 2D plane and performing the edge/face intersection in that plane (e.g. Spence [52]). Figure 3.4 illustrates this for the case of a flat helical end mill. In the second case, the intersection is done in true 3D space, Figure 3.5 (Imani et al [26]).

In both cases, the basic operation is a curve/curve (or edge/edge) intersection. For example, in semi-finishing using ball end mills, the boundary of the contact face consists of two B-spline curves and one arc. Using the B-spline endpoints, the cutter radius and the slope of the toolpath, the arc can be constructed without performing a Boolean operation at the given tool position. The intersection points of the interpolated cutting edge and the boundary of the contact face are the integration limits on the cutting edge required to compute the instantaneous cutting forces (El-Mounayri et al [15]).

Solid/Edge approach This is the approach used here and described in Table 3.1. The basic operation is a curve/surface (or edge/face) intersection. The output of the algorithm consists of two basic sets: the portion of the curve inside the solid, and the portion outside the solid. These are determined by intersecting each cutting edge of the cutter with the removed volume (obtained from intersecting the part with the tool swept volume) or the updated part (obtained from subtracting the swept volume from the part). Figure 3.6 illustrates this for the case where the updated part is used. This step is followed by classification. Classifiying the intersection points can be implemented in different ways, two of which are:

- Mid-point status
- Intersection type

In the first case, the procedure starts with intersecting the cutting edge with every face of the solid. Next, the candidate segments are determined. Each two consecutive intersection points (starting from the point with the smallest $u-$ value, where u is

the curve parameter) form one of these segments. The following step is to classify the mid-point of a given segment with respect to the solid. If this point is inside, then the candidate segment is classified as as in-cut; otherwise, it is classified as out-of-cut.

In the second case, the idea is to classify the type of face/edge intersection (step 1 above) as one of three: 1) On the face boundary (on vertex or on edge); 2) Inside the face; or 3) Outside the face. This information is then used to extract the transition type when classifying the candidate segments with respect to the solid. More details are presented in Chapter 4.

Determination of rake angle distribution

The procedure for extracting the distribution of the rake angle along the cutting edge starts with the extraction of the distribution of the normal to the rake face ($\hat{n}_r$) from the fitted B-spline surface. Next, the unit vector normal to the cutting edge and lying in the rake face (Figure 3.7) is computed:

$$\hat{n}_e = \hat{t} \times \hat{n}_r \tag{3.1}$$

The radial rake angle (α_r) which is shown in Figure 3.7 is the acute angle between the radial line and the projection of $\hat{n}_e$ on the horizontal (i.e. XY) plane, ν_h :

$$\nu_h = (n_{ex}, n_{ey}, 0)$$

Thus, the radial rake angle can be computed as a function of the z-coordinate using the cutting edge and rake face geometries.

Actual Machining Process	*Geometric Modelling and Simulation*
Workpiece	Solid model (e.g. B-rep model) of part
Removed material	Solid model = Part $\cap^*$ swept volume
Machined workpiece	Solid model = Part $-^*$ swept volume
Cutting edge	Curve model (B-spline)
Machining	Space curve/solid(surface) model intersection
Rake face	Surface model
In-cut segments	Solid (contact face)/intersections classification
Rake angle distribution	Distribution of the normal to surface

Table 3.1: Basic idea of the generic geometric modeling approach

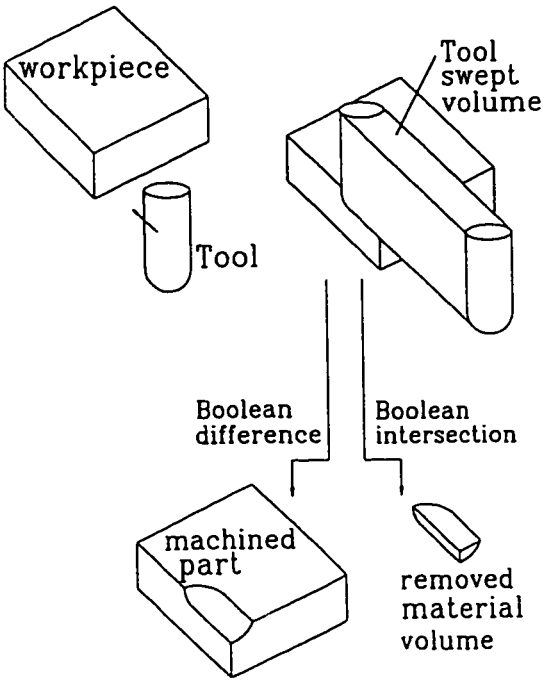


Figure 3.1: Solid modeling based simulation yielding removed material

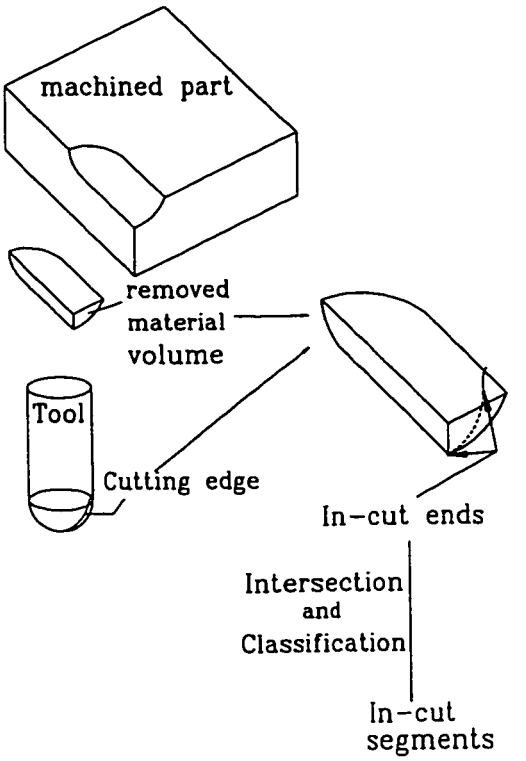


Figure 3.2: Geometric modeling based simulation yielding in-cut segments

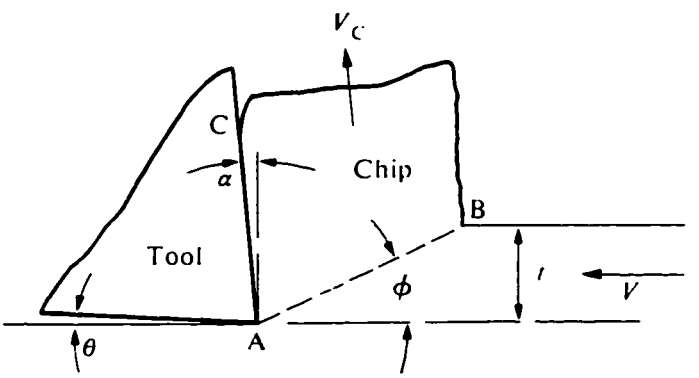


Figure 3.3: Definition of the Rake angle

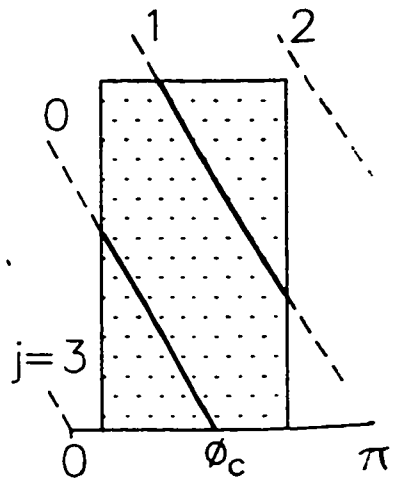


Figure 3.4: 2D approach to extracting in-cut segments using contact face

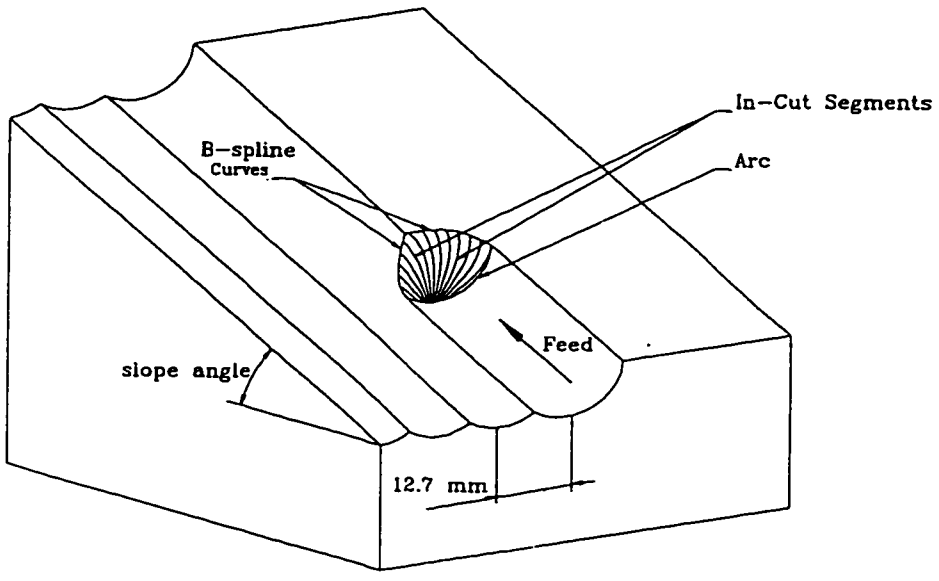


Figure 3.5: 3D approach to extracting in-cut segments using contact face

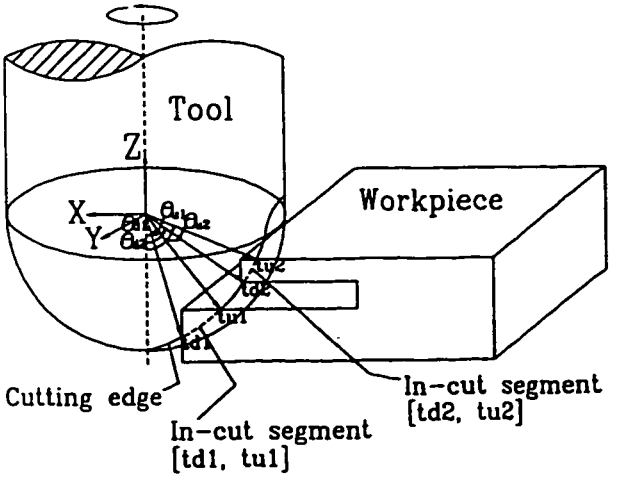


Figure 3.6: 3D approach to extracting in-cut segments using solid

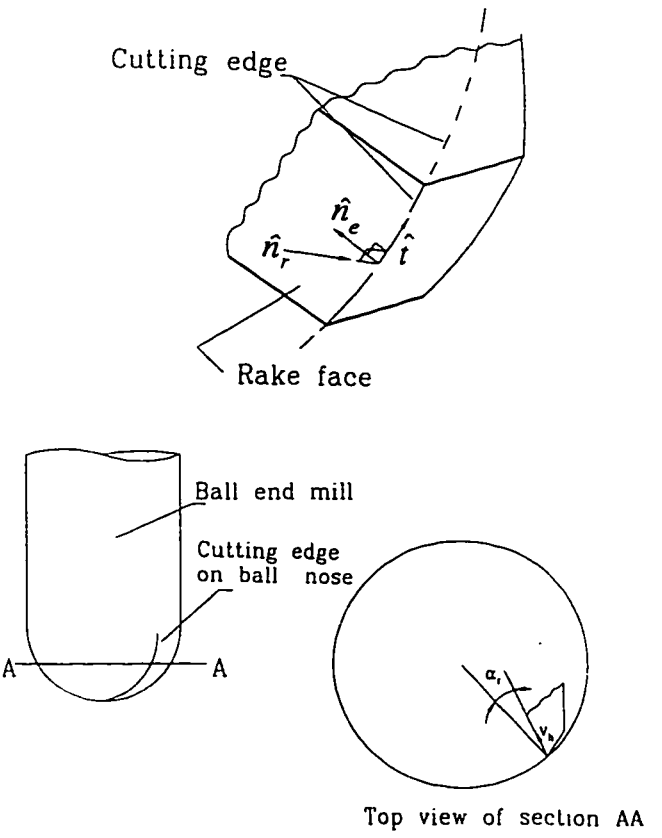


Figure 3.7: Definitions of normals to rake face and radial rake angle

Chapter 4

Generic geometric approach Implementation

4.1 Introduction

4.1.1 Simplified geometric approach

A simplified geometric approach is implemented first to verify the methodology presented in the previous chapter before the technique is actually used to implement a process-based CAD/CAM system (Chapter 5). The basic idea is described in Table 4.1. Polyhedral B-rep models are useful for numerical applications as they allow for the implementation of relatively simple procedures. Unified geometric procedures are also possible as only planar surfaces are encountered when dealing with this type of solid models. Any shape can be represented (within certain tolerance). Although their creation is difficult in general, manipulation operators (e.g. Euler operators) reduce the complexity of this task. In addition, they ensure the manifold topological

validity of a created or processed solid model.

In the event that a single cubic Bezier curve is sufficient to accurately model the cutting edge, the procedures for extracting the in-cut segments are further simplified (as shall be seen in the sequel).

In this approach, the removed material volume (represented as a polyhedral model) is used to extract the in-cut segments. The rake surface which is required to extract the distribution of the rake angle is not modelled. Consequently, a constant/average rake angle is assumed in the process simulation.

4.1.2 Selecting a geometric modeler

Based on the above considerations regarding polyhedral models, GWB (Geometric Workbench), a polyhedral-based solid modeller, was selected for the current implementation. An important characteristic of this modeler is the provision of an open architecture for extending its functionality.

4.2 Overview of GWB

A detailed coverage of the theory behind the development of GWB can be found in Mäntylä [39]. The basic ideas are as follows. A surface can be modelled and its properties studied through a two-dimensional model. Plane models are used for this purpose. The next step is to define a special topology on plane models which when taken over to surfaces will guarantee that a surface is realizable (i.e. has a realization in the 3D Euclidean space) and satisfies the topological properties. The 3D realization should be, in addition, valid and closed. A plane model is a planar

directed graph {N,A,R}, where N, A, and R are the numbers of vertices, edges, and polygons respectively. On the top of the planar graph, a special topology is defined by identifying edges and vertices of individual polygons with each other. The identification is the basis for defining the properties of plane models. In addition, it allows for the representation of nonplanar models. To represent the similar information, a data structure must be capable of representing both the graph structure and the identification structure of a plane model. The identification structure of a realizable plane model is defined as follows. Each edge is identified with one other edge, and the polygons identified at each vertex form a single cycle. GWB uses a Half-edge data structure to represent the above features. The graph {N,A,R} is simulated by a five-level structure, consisting of nodes of types: Solid, Face, Loop, Halfedge, and Vertex. For representing the edge identification, an additional node type (Edge) is included, which associates each halfedge with exactly one other halfedge. Consequently, the structure consists of a doubly-linked list of Solids, Faces, Loops, Edges, Half-edges and Vertices (Figure 4.1). Topological integrity (which distinguishes manifold surfaces from non-manifold ones) can be enforced by structural means. For example, by designing the data structure so that each edge occurs in exactly two faces, no edge can be the boundary of a missing part of the surface. The winged-edge data structure (Baumgart [9]) satisfies this criterion because it does not allow for edges occuring in just one face to be represented. The Half-edge data structure used in GWB is a variation of the full winged-edge data structure (the most general variation of the winged-edge data structure). In addition, the Half-edge data structure has the following characteristics. It makes easier and more natural the derivation of Euler operators. It allows for the representation of unintuitive models (which do appear

in the course of creating or modifying B-rep solids). Finally, it is able to store the geometric information.

4.2.1 Euler operators

As mentioned above, the theory of plane models allows the derivation of manipulation operations of plane models that produce realizable plane models. In addition, these operations are enough to model all plane models of interest. The basis for the derivation of manipulation operators for plane models is the invariance theorem (see [39] for more details). The manipulation operations on plane models have their counterparts in the form of Euler operators that act on the topology of the boundary model data structure (to create and/or manipulate boundary models). Euler operators maintain the validity and topological integrity of the boundary data structure by ensuring that all required references to nodes are maintained properly. In addition, all valid data structures can be created with a finite sequence of Euler operators. In GWB, Euler operators fall under two categories: 1) Local operators and 2) Global operators. Only global operators can modify the global topological properties of the data structure (e.g. by dividing a solid into two components). The mvfs (make-vertex-face-solid), mev (make-edge-vertex), mef (make-edge-face), kemr (kill-edge-make-ring), mekr (make-edge-kill-ring) are the main local operators. Any of these operators is implemented in two versions, a low-level one and a high-level one. Low-level operators are specified by means of pointers to adjacent nodes of the half-edge data structure, and they can work without any searching of the data structure. High-level operators, on the other hand, are parameterized in terms of face and vertex identifiers, and they will have to search for the relevant nodes of the data

structure. The algorithms implemented in this work to generate swept volumes are based mainly on three operators: mvfs, mve, and mef. The mvfs creates the skeletal model. It creates an instance of data structure that has just one face and one vertex. This means that the new face has one empty loop with no edges at all. The mev subdivides the cycle of edges of vertex into two cycles by splitting a vertex into two vertices joined with a new edge. The net effect of this is to add one vertex and one edge to the data structure. The mef subdivides a loop by joining two vertices with a new edge. Its net effect is to add one new edge and one face to the data structure.

4.2.2 Architecture of GWB

As a software, GWB is constructed in layers, each forming a functionally complete modelling tool suitable for the implementation of the next layer (Figure 4.2).

The innermost layer consists of the internal data representation (the Half-edge data structure) and a set of procedures for its low-level manipulation. The next layer consists of Euler operators and a set of geometric procedures. The fact that Euler operators form the basic modelling tools, that are used to implement all higher-level operations, guarantees the topological validity of created as well as processed solid objects. The following layer consists of various medium-level modelling procedures. As far as implementing a machining process simulation is concerned, they allow for two things. First, the construction of various solid primitives including swept volumes using Euler operators (from the lower-level). Second, various model manipulation tools such as Boolean set operations (which allow for the generation of the removed material volume as well as the machined part for each cutting step). For the current

work, geometric algorithms are developed at the low and medium levels of the architecture. The highest level is the user interface through which the modelling tools of the various levels are made available to the user.

4.2.3 Extending GWB

Applications can be developed by using and/or extending the functionality of GWB. The latter offers a relatively simple data structure (written in C), low level operators, and higher level procedures (e.g. Boolean set operators). Low level operators can be used to create a new solid, access as well as manipulate its data structure. This structure, in the form of a doubly-linked list, allows for a two-way navigation which is more practical and efficient. The C definition of the Half-edge data structure (Appendix C) shows this characteristics which is achieved by using a *next-* and *prev-* pointers for each entity forming the solid model to point to the next and previous entities in the list respectively.

Consequently, a typical extension (i.e. a new application or procedure) is developed at the medium or high level of the architecture, and uses procedures from the same or lower level(s).

4.3 Implementation of the simplified geometric approach using GWB

4.3.1 Solid modelling representation

The basic concept used to simulate a machining process involves a representation of the tool swept volume (as was illustrated in Figure 3.1). By using the boundary data structure and appropriate Euler operators allows the natural implementation of mathematical algorithms for modelling the tool swept volume. The basic element of the sweep envelope is the driving edge, Figure 4.3(a). The latter (also called critical curve) depends on the shape, orientation, and instantaneous path direction of the tool. In 3-axis machining, the critical curve is invariant and is generated based on the shape and initial orientation of the tool. Figure 4.3(b) illustrates the effect of the instantaneous path direction on the driving edge of a cylindrical tool. The determination of the proper driving edge can be done using vector dot and cross product calculations (Elmaragy [18]). Vectors **C** and **D** in Figure 4.4 are the result of cross products and are used to define the start and finish of the semi-circular edge. The surfaces produced by sweeping the driving edge are the path dependent surfaces of the swept volume; these and the path "independent" surfaces form the complete skin of the swept volume (this is illustrated in Figure 4.5 for the case of a cylindrical tool). The path dependent surfaces can be generated as follows. The active (driving) edges are generated at incremental positions. Next, the path lines connecting the active edges form the bounding edges of the sweep surface.

Using a polyhedral representation, the facets forming the surface can be automatically generated as being bound by the path segments or edges and the active edges (see Figure 4.6(a)). The active edges are produced at the incremental positions by applying proper transformation matrices to the vertices forming the driving edge at the initial position, then connecting the vertices in their new location with edges. For a different representation or method which requires more information about the path dependent surfaces, then normals to those surfaces can be produced. Coordinates of points on any of these surfaces are based on the path line and the line defined by the driving edge. Given these, the surface normal can be calculated as the cross product of the circle tangent and the path tangent at the point in question (Figure 4.6(b))

The approach used here to generate the swept volume consists of the following. First, the driving edge is generated. The swept volume is then produced by sweeping the vertices of the driving edge incrementally along the cutting path while creating the corresponding edges and faces using the appropriate Euler operators. The latter consist of mev (make-edge-vertex) and mef (make-edge-face). This is continued until the other end of the interpolation is reached. At this point, the missing part of the swept volume is generated by sweeping the appropriate driving edge(s). In most instances, low level Euler operators are used. These operators are very efficient because they deal directly with pointers and expect the proper half-edges and other parameters (in the data structure) to be directly passed and/or accessed. Another advantage of the current implementation is that path dependent surfaces are the natural result of sweeping the driving edge(s) along the tool path, thus, requiring only the basic information included in the cutter location (CL) data file. Simplifications are made to improve the efficiency further, without compromising the validity of the

simulation. The idea consists of distinguishing between the cutting surfaces of the tool and the non-cutting ones. Then, only the portion of the swept volume formed by the latter needs to be modelled exactly. The fact that other portions can be represented approximately makes possible the reduction of the total number of faces used to model the skin of the swept volume. Here, only the top surface is considered to be non-cutting all the time. This idea was first expressed by Voelcker and Hunt [62] who distinguished between total swept volume and operative swept volume. Figure 4.7 shows a sample of swept volumes for flat and ball end mills along linear and circular paths.

4.3.2 Curve modelling representation

Cubic Bezier curves or piecewise continuous cubic Bezier curves are used to represent the cutting edge. This is useful here as it reduces the cutter-part intersection operation to computing the roots of a cubic equation which is computationally very efficient. In addition, it allows for a unified approach for the geometric simulation. In general, a sufficient number of Bezier segments should be used to achieve an accurate reproduction of the edge.

Single segment fitting

For the particular case where a single segment is sufficient to accurately represent the edge, the following approach can be used. First, to fit a cubic polynomial to the cutting edge (e.g. using the least squares method). The parametric equation of such

a polynomial can be written as follows:

$$x(t) = cx_3 \cdot t^3 + cx_2 \cdot t^2 + cx_1 \cdot t + cx_0 \tag{4.1}$$

$$y(t) = cy_3 \cdot t^3 + cy_2 \cdot t^2 + cy_1 \cdot t + cy_0$$

$$z(t) = cz_3 \cdot t^3 + cz_2 \cdot t^2 + cz_1 \cdot t + cz_0$$

Second, to derive the four Bezier (control) points which corresponds to an identical cubic polynomial. These are given as follows:

$$b_0 = (bx_0, by_0, bz_0) \quad b_1 = (bx_1, by_1, bz_1) \quad b_2 = (bx_2, by_2, bz_2) \quad b_3 = (bx_3, by_3, bz_3)$$

where:

$bi_0 = ci_0$

$bi_1 = 1/3 \cdot ci_1 + ci_0$

$bi_2 = 1/3 \cdot (ci_2 + 2 \cdot ci_1 + 3 \cdot ci_0)$

$bi_3 = ci_3 + ci_2 + ci_1 + ci_0$

In the above equations, i stands for x, y, or z. The final step consists of transforming the control points to locate the edge at the required position. Table 4.2 shows the control points of the Bezier curve fitted to the cutting edge on the ball nose of a one inch ball end mill. One Bezier segment was sufficient to accurately represent the edge. The analytical equation given by Yang and Park [65] was used in the first step of the modelling process. This equation is written as follows:

$$\begin{aligned} x(\theta) &= R \cdot \cos(\alpha_n) \cdot \sin(\theta) \\ y(\theta) &= -1/2 \cdot R \cdot \sin(2 \cdot \alpha_n) \cdot [\cos(\theta) - 1] \\ z(\theta) &= -R \cdot [\cos^2(\alpha_n) \cdot \cos(\theta) + \sin^2(\alpha_n)] \end{aligned} \tag{4.2}$$

where R is the cutter radius, α_n is the normal rake angle of the plane rake face (e.g. 15^o), and θ is the angle of the point in the spherical coordinate system. In addition to the small error at $\theta = 90°$, equations 4.2 are difficult to transform for 3-5 axis motions, and, being transcendental, are not easily solved for intersection with the part surface. On the other hand, the fitted Bezier curve is of order 3. Its equation can be expressed in terms of the control points as:

$$b(t) = (1-t)^3 \cdot b_0 + 3 \cdot t \cdot (1-t)^2 \cdot b_1 + 3 \cdot t^2 \cdot (1-t) \cdot b_2 + t^3 \cdot b_3 \qquad (4.3)$$

where $b(t) = [x(t)\ y(t)\ z(t)]^T$ and the control points are the ones given in Table 4.2. It is now possible to satisfy the boundary condition $z(\pi/2) = 0$ by adjusting the location of b_4 to coincide with $[x(\pi/2), y(\pi/2), 0]$. This will result in global alteration of the curve. To allow for local control of the shape, a set of Bezier segments or a B-spline curve can be used to represent the cutting edge.

Multi-segment fitting

Above, the part of the cutting edge on the ball nose was closely fitted with a single cubic Bezier segment. On the other hand, multiple segments (i.e. piecewise continuous cubic Bezier segments) are required to closely fit the complete edge or the part of the edge on the flat section of the tool. Table 4.3 shows the control points fitted to a portion of the cutting edge on the flat part of a one inch ball end mill. The analytical equation of this portion of the edge is given by:

$$\begin{aligned} x &= R \cdot cos(\varphi) \qquad (4.4) \\ y &= R \cdot sin(\varphi) \\ z &= \frac{R \cdot \varphi}{tan(\gamma)} \end{aligned}$$

where, γ is the helix angle (30°), R is the tool radius (0.5 inch), and φ is the angular position of the point on the cutting edge measured from the positive x-axis ($\varphi \in [0, 90]°$). Having seven control points modeling the edge means that the latter is actually represented by two cubic segments.

4.3.3 Extraction procedure of in-cut segments

The method used here to generate the cutter-part immersion geometry consists of intersecting each tooth of the cutter with the removed material volume obtained from intersecting the part with the tool swept volume. In GWB, a solid is represented as a polyhedral model which involves only planes. Since a cutting edge is represented as cubic Bezier (or piecewise continuous cubic Bezier) curve(s), the highest and only polynomial order used is cubic. Consequently, there is only one intersection operation involved during the process of extracting the cutter-part immersion geometry: plane/cubic polynomial intersection. This is equivalent to solving a third order equation which is computationally simple and highly efficient, having a closed form solution. In addition, the current implementation is based on a unified approach. Indeed, at any time during the simulation, only one type of surface (planar) and one type of curve (cubic polynomial) need to be processed. As far as the extraction of the cutter-part intersection geometry is concerned, the basic operation consists of solving the system made of equation 4.1 which represents a cubic polynomial curve and the planar face equation:

$$A \cdot x + B \cdot y + C \cdot z + D = 0 \tag{4.5}$$

By substituting x, y, z in the plane equation by $x(t)$, $y(t)$, $z(t)$ representing the coordinates of a point on the cutting edge or one of its segments (i.e. equation 4.1),

the following equation is obtained:

$$K_0 \cdot (1-t)^3 + K_1 \cdot t \cdot (1-t)^2 + K_3 \cdot t^3 + K_4 = 0 \quad t\epsilon[0,1] \tag{4.6}$$

where

$$\begin{aligned} K_0 &= A \cdot bx_0 + B \cdot by_0 + C \cdot bz_0 \\ K_1 &= 3 \cdot (A \cdot bx_1 + B \cdot by_1 + C \cdot bz_1) \\ K_2 &= 3 \cdot (A \cdot bx_2 + B \cdot by_2 + C \cdot bz_2) \\ K_3 &= A \cdot bx_3 + B \cdot by_3 + C \cdot bz_3 \\ K_4 &= D \end{aligned}$$

Solving the above equation for t yields the intersection between the tooth and the plane containing one of the faces of the removed material volume. Next, a containment test is performed to determine whether a given intersection is inside, outside, or on the boundaries of the face (see Figure 4.8). The algorithm used here is based on the following. A ray is sent from the intersection point to infinity and the number of intersections of this semi-line with the edges of the face is counted; if this number is even, then the intersection vertex is outside the face, otherwise it is inside (Figure 4.9). The algorithm also reports intersections on the boundary of the face (Mäntylä [39]). By repeating the above steps for each face forming the removed material volume, and then classifying the resulting intersections, the segments of the cutting edge lying inside that volume can be extracted in the following form:

$$[td_i, tu_i] \qquad i = 1, 2, ...n$$

where n is the number of in-cut segments (i.e. portions of the cutting edge which are cutting material), and td_i and tu_i are, respectively, the lower and upper ends of

the *ith* in-cut segment (refer back to Figure 3.6). Note that multiple in-cut segments are possible. The idea here is to start at the tooth's bottom ($t = 0$) and move towards the top ($t = 1$). The usual case is illustrated in Figure 4.10 on the left, where the portions of the cutting edge inside the material are: $[P(t_1), P(t_2)], [P(t_3), P(t_4)]$ and $[P(t_5), P(t_6)]$. In degenerate cases, however, the edge intersects the boundary of the removed material volume at an edge or vertex. For instance, $P(t_5)$ and $P(t_6)$ (in Figure 4.10 on the right) are intersections points, but $[P(t_5), P(t_6)]$ is not an in-cut segment. Differentiating between transition types robustly requires use of three dimensional edge (or vertex) neighbourhood procedures. The details of such a procedure can be found in Requicha and Voelcker [46]. Here, the type of intersection as reported from the facet/intersection classification (i.e. inside, outside, on edge or on vertex) is used to help distinguishing between transition cases. To avoid tolerance problems at the ends of the cutting edge (bottom and top) when performing the intersection procedure, an extended edge can be used (i.e. $t \in [-0.1, 1.1]$ instead of $t \in [0, 1]$). In this case, the algorithm for extracting the in-cut segments starts from the intersection point with the lowest $t-$ value (first intersection). The intersection point with the next higher $t-$ value is the second intersection. This and the first one form the first candidate segment. Now depending on the type of intersection of the ends of this segment, the latter is classified as in-cut or out-of-cut. For example, if the first intersection is of type "on edge" or "on vertex" and the second intersection is of the same type, then the segment is out-of-cut; however, if the second intersection is of type "inside", then the segment is in-cut. This information (i.e. status of the first candidate segment) is used together with the intersection type of the next two intersections to classify the next (second) candidate segment. This procedure is

continued until the last two intersections are classified. The fact that a tooth only cuts on the front 180° dictates a final filtering of the results of the above classification (i.e. the intervals $[td_i, tu_i]$) which is implemented as follows. The plane that sets the boundary between the contact area (between the removed material volume and the front 180° of the tool) and the removed material volume (Figure 4.11) is generated for each tool position and intersected with the teeth during entry and exit. This plane is vertical (i.e. perpendicular to the XY-plane) and its normal is parallel to the instantaneous direction of the tool motion. The results are used to update the intersection intervals obtained from the previous step, yielding the final set of in-cut segments:

$$[td_j, tu_j] \qquad j = 1, 2, ...m$$

where m is the actual number of in-cut segments. By superposing the lower and upper ends of the in-cut segments, the immersion geometry is obtained.

4.4 Verification of the geometric approach

The geometric approach is verified here for different cases of ball end milling, including a case of die semi-finishing. A dynamic regenerative semi-empirical force model for ball end milling is used to compute the cutting forces.

4.4.1 Semi-empirical model for ball end milling

Semi-empirical force models are based on the concept of equivalent orthogonal cutting conditions and empirical equations for computing shear angle, friction angle, and

shear strength of workpiece. Yang and Park [65] have recently implemented a semi-empirical force model for ball end milling. In this model, the cutting edge is discretized into very small segments and the differential force acting on each oblique edge is calculated as:

$$dF_r = \frac{\tau_s \cdot dA_c}{sin(\phi) \cdot cos(\phi + \beta - \alpha_e)} \tag{4.7}$$

where ϕ is the shear angle, β is the friction angle, τ_s is the shear strength of workpiece, dA_c is the cross-section area of undeformed chip on the infinitesmal edge (basically a function of the instantaneous chip thickness), α_e is the effective rake angle. The empirical equations (for computing ϕ, β and τ_s) derived by Yang and Park are for 1045 steel. A new set of equations is required for each new material. This force model can compute cutting forces for the case of 3-axis milling (Park et al [44]). The differential torque applied to an infinitesimal element of the in-cut segment is computed as follows (Abrari [2]):

$$dT_z = (dF_r \cdot cos(\beta - \alpha_e)) \cdot R \cdot sin(\theta) \tag{4.8}$$

where θ is the angle which indicates the position of the (infinitesimal) cutting element on the in-cut segment; $\theta \epsilon\ [\theta d, \theta u]$, where θd and θu are shown in Figure 4.12.

4.4.2 Extension to dynamic regenerative model

Modelling the cutting process dynamics

In a dynamic regenerative force model, the dynamic behaviour at the level of the cutting edge is incorporated. The spindle and overhang (e.g. tool and tool holder) represent the most flexible parts of the machine tool structure when performing end

milling (see [37] and [28]). This and the fact that the current work is limited to 3-axis milling where the rotary table (of the 5-axis machine tool) is kept locked during cutting, justify the restriction of the dynamic modelling to the spindle and overhang (i.e. the rest of the structure is assumed rigid). Also, because the focus of the current work is the machining of molds and dies, the workpiece is assumed to be rigid too.

In theory, a machine tool structure is a continuous system which has an infinite number of degrees of freedom. However, it can be adequately modeled as a discrete system with a finite number of degrees of freedom. On the other hand, properties of vibrating systems can be considered (without loosing much accuracy) as lumped into single elements. The latter are springs, dampers, masses which respond with reaction forces to displacement, velocity and acceleration respectively. Consequently, a machine tool structure can be treated as lumped mass system which is basically a particular type of discrete systems consisting of the three discrete elements: Inertias, Dampers, and Springs.

The first step in the modelling procedure consists of extracting the modal parameters for the tool and tool holder of the particular machine tool used in this work, a FADAL VMC 4020 5-axis CNC vertical machining center. Here, the tool holder consisted of the machine spindle, spindle dynamometer, and a tool adaptor (between the tool and the dynamometer). The latter is needed because, though the dynamometer is delivered with an integrated spindle adaptor, a special adaptor is required to mount the cutting tool to it. The constraints in the design of the tool adaptor were geometric and functional. The geometric ones consisted of the location and size of the four holes in the dynamometer to which the adaptor is to be mounted as well as the size of the required tool and its standard dimensions. The functional ones are

related to the mass and stiffness of the tool adaptor which are expected to directly affect the dynamic behavior of the system by increasing its flexibility. Based on these constraints, a special purpose adaptor was designed that can only accomodate end mills with 0.75 inch shank diameter (see Figure 4.13).

The dynamic properties of the process were determined by conducting a standard impact test. The basic idea of the impact test as well as the details of the procedure used to extract the modal parameters for the current tool and tool holder structure are presented in Appendix E. Basically, a Kistler hammer with a force transducer at the tip (the input) and an accelerometer mounted to the tool at the level of the cutting edge (the output) are used. The inertance (acceleration/force) was measured and then integrated twice to produce the compliance from which the modal parameters were extracted. The frequency range of analysis was determined based on the range of cutting speeds and type of tools used during the cutting tests as well as the high power frequencies in the measured cutting forces. These frequencies were identified by performing an FFT of the measured cutting forces (see Figure 4.14). The range was defined as $[0, 1100]$ Hz. The result of the impact test revealed two dominant modes in that frequency range (see Figure 4.15). In the x-direction, the mode frequencies were: $fx_1 = 313$ Hz and $fx_2 = 437$ Hz. In the y-direction, $fy_1 = 456$ Hz and $fy_2 = 553$ Hz.

The second step is to represent the dynamic behaviour of the machine tool structure (spindle and overhang) using a dynamic model. Because there are two significant modes in each direction (x and y), the vibration of the tool was modelled using a lumped mass system with two degrees of freedom in each direction (see Figure 4.16). The modal parameters were extracted by fitting the transfer function of a coupled

two degrees of freedom to the measured compliance. Table 4.4 shows the values of the parameters. Figure 4.17 shows the system which predicts the cutting force in any of the two directions ($x-$ and $y-$), where f is the force component, and η_1 is the vibration at the level of the cutting edge. In each direction, the governing differential equations can be written as follows:

$$M_1 \cdot \ddot{\eta_1} + C_1 \cdot (\dot{\eta_1} - \dot{\eta_2}) + K_1 \cdot (\eta_1 - \eta_2) = f \quad (4.9)$$

$$M_2 \cdot \ddot{\eta_2} + C_2 \cdot \dot{\eta_2} + K_2 \cdot \eta_2 + C_1 \cdot (\dot{\eta_2} - \dot{\eta_1}) + K_1 \cdot (\eta_2 - \eta_1) = 0 \quad (4.10)$$

from which $\eta_{1,x}$ and $\eta_{1,y}$ (which correspond to the tool vibration at the level of the cutting edge in $x-$ and $y-$ directions respectively) are computed. Next, the dynamic chip load is determined and used to compute the force components and torque, as explained in the next section.

Extending static model to dynamic regenerative

Regenerative force means that the force on any tooth in the cut depends on the surface which was left by the passage of previous teeth, in addition to the feed per tooth and the deflection of the cutter (Smith and Tlusty [50]). In order to compute the regenerative forces, the general expression of the chip load which includes the feed per tooth, the cutter dynamic deflection, and the surface waviness is used; the latter can be written as follows [2]:

$$f_{inst} = f \cdot sin(\psi) + x \cdot cos(\psi) + y \cdot sin(\psi) - max(u_1, u_2, u_3) \quad (4.11)$$

where, f_{inst} is the instantaneous chip thickness, f is the feed per tooth, ψ is the rotation angle of the cutter with respect to z-axis, x and y are the tool deflections, and u_1, u_2 and u_3 are the undulations left behind from previous teeth.

The force simulation runs in a series of small time steps (e.g. 1/3600 sec). At each instant, the force on each tooth in immersion with the part is calculated and the forces are summed vectorially. They are then applied to the dynamic system of the machine tool structure to compute the x and y deflections required for the next time step. This is done as follows (Tlusty and Ismail [57]; Smith and Tlusty [50]). The accelerations of the current step are computed in all the degrees of freedom based on the force and then integrated twice to produce displacements for the next step which lead to the displacement of the tool. These deflections together with the static chip thickness (of the next step) and undulations left behind from previous passes are used to compute the instantaneous chip thickness (equation 4.11) required to predict the dynamic cutting forces for the next time step. The extended (regenerative) force model also simulates the ploughing forces. The corresponding equations, for the case of ball end milling, can be found in [2].

Based on the history of system deflections, the work surface can also be generated. In addition, unstable cutting condition (chatter) can be detected by analyzing the growth of deflections in the structural system.

4.4.3 Geometric input to force model

The in-cut segment(s) of each tooth in immersion with the part are the basic geometric parameters. Each in-cut segment is expressed in terms of two angles (θd and θu), as depicted in Figure 4.12. For each cutter flute, the set of $[\theta d_j, \theta u_j]$ can be directly computed from the set $[td_j, tu_j]$ extracted by the geometric procedure, using the mapping relation:

$$\theta = \theta(t) \ \ (rad) \tag{4.12}$$

where, in general, $\theta(t)$ is also a cubic Bezier curve. For the ball nose example, it was found that a sufficient approximation is:

$$\theta = \pi/2 \cdot t \quad (rad) \tag{4.13}$$

In equation 4.7, dA_c, the cross-section area of undeformed chip on infinitesimal edge, represents the geometric information. It is a function of θ, where θ describes the position of the infinitesimal element along the cutting edge and is computed from θd and θu (of the in-cut segment to which this infinitesimal element belongs), and the discretization increment (Abrari [1]). More generally, the geometric procedure will supply all the information required by the force module to compute any of the process parameters.

4.4.4 Verification

Measuring forces and torque experimentally

A Kistler type 9123B1311 rotating dynamometer was used to measure the three F_x, F_y, and F_z components of the instantaneous cutting force as well as the torque T_z. A spindle shaft encoder, an external clock, and an external trigger were used to provide synchronization with the tooth position. The cut-off frequency was set to 1.2 kHz. The A/D board used allowed for a sampling rate of 2500 points/sec/channel. The workpiece material was 1045 steel, and the cutting conditions used consisted of 0.1 mm feed/tooth, and 600 rpm spindle speed (The encoder was mounted on the motor and in the range of cutting speeds used, the ratio of motor to spindle speed was about 1.95. The actual sampling frequency was measured using an oscilloscope).

It should be noted that the spindle dynamometer measures the force components

in the rotating coordinate system. This allows measurements to be made with reference to the cutting edge. Consequently, the simulated force components (F_x and F_y) computed in the stationary coordinate system have to be converted to the rotating frame. The relation between F_x and F_y in the stationary frame and their counterparts in the rotating one (RF_x and RF_y) can be written as follows:

$$\begin{aligned} RF_x &= F_x \cdot cos(\phi + \phi_0) - F_y \cdot sin(\phi + \phi_0) \\ RF_y &= F_x \cdot sin(\phi + \phi_0) + F_y \cdot cos(\phi + \phi_0) \end{aligned} \tag{4.14}$$

where, ϕ is the tool rotational angle and ϕ_0 is the angle between the tangential component of the cutting force and the direction of RF_x.

Predicted forces and torque and comparison with measurements

For each in-cut segment of the tooth at a given tool angular position, the corresponding differential force components (and torque) are automatically computed by the force algorithm. The total force applied to a tooth consists of the vector sum of all differential forces. In general, the tool angular increment is in the range of 0.5 to 3.0°.

Up-milling and Down-milling Figures 4.18 shows the cutter-part immersion geometry in the form of lower and upper ends of in-cut segments for two cases of 2D linear interpolation: down milling in 4.18(a) and up-milling in 4.18(b). In both cases, the cutting is limited to the ball nose of the tool. The corresponding cutting forces are shown in Figure 4.19.

Figure 4.20 shows a case of linear interpolation from a machined surface. Figure 4.20(d) shows the geometric parameters for the first and second flutes over one revolution. Figure 4.21 shows a comparison of experimental and corresponding simulated

forces. The power spectrum for experimental force components is shown in Figure 4.22. A comparison between the FFT of experimental F_x and F_y and that of predicted ones in the frequency range $[450, 650]$ Hz is depicted in Figure 4.23.

Figure 4.24 depicts the case of circular interpolation with variable immersion (only the upper end of the in-cut segment is shown). The three different stages of this cut are: entry with variable immersion, followed by full-immersion, and finally a region of variable immersion. Figures 4.25 and 4.26 show the geometric and physical information for a 2D circular interpolation with multiple immersion. Figures 4.27 and 4.28 depict geometric and physical information for another 2D circular interpolation with multiple immersion.

Figure 4.29 shows a case where the z-level of the tool changes during cutting (i.e. 3D interpolation), and correspondingly, the amplitude of the cutting ends decreases. Note that cumulative angular position (i.e. resulting from adding up consecutive tool revolutions) is shown in the horizontal axis of the Figure. Figure 4.30 depicts a case of variable immersion where the variation of forces is captured over the whole cutting path (Figure 4.31).

Semi-finishing of dies and molds Machining costs represent almost 50 % of the total production cost in die/mold making. Around 40 % of this cost corresponds to rough and semi- finishing (Rigby [48]). For these two operations, the current geometric simulation can generate reliable immersion data which is used to predict cutting forces and optimize the cutting conditions. In addition to reducing the cycle time, planning the roughing operation is particularly important in completely automated

or sustained untented jobs. This is due to the fact that in roughing, the level of cutting forces is much higher than that encountered in other operations and thus is more prone to violating the process constraints if cutting conditions are not accurately planned.

Figure 4.32 shows the output of the geometric simulation for a typical case of die/mold semi-finishing. Roughing which is performed with flat end mills leaves behind shoulders, Figure 4.32(a) that are removed using ball end mills. Figure 4.32(b) shows the sequence of the cutting. The immersion geometry corresponding to the second pass of semi-finishing is shown in Figure 4.32(c) where each tooth has two in-cut segments and consequently two immersions. The output is represented in the form of individual tooth immersion(s). This is obtained by superposing the lower and upper ends of each in-cut segment. Forces were measured when the tool was cutting along a circular path. Figure 4.33 shows a comparison of experimental and simulated cutting forces. In general, one can notice a very good agreement between the numerical simulation and the corresponding experimental verification (in most cases, the difference is below 10 %).

Chatter prediction

The current force model is able to predict the onset of chatter. Abrari et al [2] developed stability surfaces for the case of shoulder removal using up-milling at 0.1 mm/tooth with a 24.5 mm, 4-flute varying helix angle ball end mill. For a given depth of cut, the corresponding stability lobes are generated by taking a cross section of the stability surfaces at that value. Figure 4.34 shows the prediction of chatter as characterized by a growth in the tool deflection, Figure 4.34(b), and by the cutting

conditions lying outside the stability lobes, Figure 4.34(c).

4.5 Discussion

In general, there is a very good agreement between the simulated cutting forces and the corresponding experimental measurements. The following aspects form the basis of the comparison between the two:

- The cutting period for one tooth
- The force level (basically, the peak value)
- The force pattern, including the run-out
- The frequency and level of vibration depicted on the force pattern

First, the cutting period of the predicted forces is an outcome of the geometric module. It compares very well with the one captured experimentally. Second, the force level is a function of the in-cut segments (i.e. geometric information) and the corresponding differential forces (i.e. physical information). The very good agreement reflects the accuracy of both modules (the geometric and physical) and thus the overall simulation. Next, the force pattern is very similar and the run-out is simulated successfully. It was measured using a dial gage and found to be 0.03175 mm. Little could be done to reduce this value. This was mainly due to the fact that the tool is mounted to the tool adaptor using a single screw (see Figure 4.13) and this tends to increase the eccentricity between the tool axis and the spindle axis. It should be noted that the run-out was simulated at the level of the physical simulation after the

in-cut segments were extracted by the geometric simulation. The value of the radial run-out measured above was input to the force model which modified the in-cut segments accordingly. It is also possible to model the run-out as part of the geometric simulation by repositioning the tool using value of the measured run-out before intersecting the tool with the part. Third, most cases show a very close matching between the frequency of vibration of the simulated forces and the experimental ones (This good agreement was shown in Figure 4.23 where the output of an FFT analysis of experimental forces was compared with that of their simulated counterpart.

Few other aspects regarding the measured force components need to be considered. First, the measurements show forces when the tool is out-of-cut due to the natural vibration of the system (spindle and overhang) between the intermittent cuts. Since they are not cutting forces, no attempt was made to predict them. Next, the dominant frequency of vibration in the $x-$ direction is higher than the natural frequency of the most dominant mode in that direction (it is actually, very close to the vibration frequency in y-direction). The reason is as follows. The frequency of vibration seen in F_x and F_y comes from the dynamic chip load variation. The latter is the result of the combined effect of vibrations in $x-$ and $y-$ directions. Because the most dominant mode in $y-$ direction is the most flexible one in the XY plane, it dominates the variation in the dynamic chip load and thus the vibration seen in F_x and F_y. Finally, the level of vibration seen on the measured F_z and T_z is higher than that on the predicted ones. This can be attributed to two reasons. First, the dynamics in $z-$ direction was not simulated because the system was assumed rigid in that direction. This assumption might not be very accurate due to the presence of the spindle dynamometer and the tool adaptor which would probably introduce some

flexibility in the tool and tool holder structure in that direction. The flexibility in the $z-$ direction when modelled is expected to have a direct effect on F_z that will also reflect on T_z. Indeed, T_z is computed from F_x and F_y which are functions of the dynamic chip load. The latter will be, in case of simulating the dynamic in the $z-$ direction, a function of the vibrations in the three directions as opposed to two in the current implementation. Second, it might be that the spindle dynamometer is acting as a low pass filter when reading F_z and T_z.

As shown above, the simulation of machining operations is dealt with as a curve/solid model classification procedure. In other words, it involves a curve representation (to represent cutting edges) as well as a solid representation (to represent parts). Intersecting a free-form surface with a Bezier curve is performed by solving the equation:

$$P(u,v) - Q(t) = 0 \tag{4.15}$$

where, $P(u,v) = 0$ is the parametric equation of the surface patch (e.g. plane, B-spline, Coons) and $Q(t) = 0$ is the parametric equation of the curve (e.g. Bezier, B-spline). The above equation is a system of three scalar equations in three unknowns (u, v, and t). In general, this system is non-linear and thus requires iterative methods for its solution. The complexity of the solution implemented herein is reduced by restricting it to planar surfaces and cubic polynomial curves.

The technique developed here to extract the immersion data required for computing instantaneous cutting forces is equally applicable to other machining processes. As long as the geometric expressive power used here covers all the shapes encountered in a certain machining process (to the required level of accuracy), successful geometric simulation requires only the development of a library of swept volumes for

that particular machining operation and the fitting of 3D space curves to the cutting edge. With Bezier curves any cutting edge shape can be used.

The requirement for the physical simulation depends on the type of force model used. A mechanistic force model requires empirical constants for the particular tool/workpiece combination, cutting conditions and cutting edge geometry being used. A semi-empirical model needs empirical equations (for ϕ, β and τ_s) for the material of the part being machined.

As described previously, the basic output of the geometric procedure consists of the immersion angles (or in-cut segments). For each immersion interval, the force model computes the cutting forces at each tool angular position (or time step) after modifying the instantaneous static chip thickness with the tool deflection and undulations left on the surface as expressed in equation 4.11. Although the deflections affect the immersion angles too, for roughing and semi-finishing using relatively large tools, the amount of change is negligible. For example, this was true in the case shown in Figure 4.32. In the event of a significant change, the following approach can be used to account for the effect of the tool deflection on the immersion angles. A swept volume larger than the actual rigid swept volume is generated (the rigid swept volume refers to the swept volume generated assuming a rigid tool moving along the cutting path). This is implemented by using a tool with a radius $R+\delta$, where R is the radius of the cutting tool and δ is a liberal estimate of the maximum expected deflection. At time zero (i.e. the begining of the simulation), the cutting edge lies on a rigid tool. The tool is rotated a small increment $0.5 \leq \Delta\theta \leq 3.0°$, the in-cut segments are extracted, and forces computed from which deflections are estimated. The cutting edge is repositioned accordingly and then rotated about the z-axis by $\Delta\theta$. It is then

intersected with the "larger than actual" swept volume (Figure 4.35) to yield the in-cut segments(s), from which dynamic cutting forces and resulting deflections are computed. The above steps are repeated as the tool is rotated incrementally. This is continued until the end of the tool path. The resulting part surface after completing the tool path will be generated as follows. The tool is moved a step equal to the feed/tooth along its cutting path. For each tool position, the maximum deflection experienced is stored. This deflection is measured along the negative x-direction of the local coordinate system (see Figure 4.36). Using this maximum deflection, the programmed position of the tool center is modified to account for the dynamic tool deflection. After performing the modifications for all tool positions along its cutting path, linear segments are fitted to simulate the corrected tool path. Positions which are colinear are fitted with a single segment to minimize the total number of segments. Next, the swept volume which corresponds to the actual tool path is created and used to update the part (El-Mounayri et al [17]).

Extending the same geometric approach to 4- and 5- axis milling with general tool shapes, needs only a representation of the volumes swept by different tools in 4- and 5- axis motion. This can be done by direct application of the envelope equations to calculate discriminate curves at increments along the tool path as reported by Narvekar et al [42]. This swept volume is intersected with the part to yield the removed material volume which is then intersected with the cutting edge(s) to produce the in-cut segments. The geometric procedures involved are the same as the ones implemented here. In particular, the basic operation of intersecting the fitted curve(s) with the removed material volume is a cubic polynomial/planar face intersection.

The geometric development is general and can simulate the machining of complex parts (e.g. parts with sculptured surfaces). However, a fully featured workpiece would involve a large number of facets for an accurate representation to be maintained. In general, there are two procedures which are directly affected by such an increase in the number of facets: 1) Boolean operation and 2) Cutting edge/part intersection. The current approach, however, intersects the cutting edge with the removed material volume instead of the updated part. The number of facets required to accurately represent the removed material is nearly constant. Consequently, the computational time for extracting the in-cut segments by intersecting the cutting edge(s) with the removed material volume does not present a problem in the event of a fully featured workpiece. On the other hand, the Boolean operation is expected to become computationally more expensive. However, techniques from computational geometry can be used to significantly improve the efficiency. One possible approach is to use isoboxes (see Hoffmann [25]). This can be implemented at two different levels: 1) Prior to the Boolean operation and 2) During the Boolean operation. At the first level, the idea is to localize the search for intersections to the region where the current machining step is taking place. This can be achieved by using an isobox that encloses the tool swept volume for the current machining step to identify the facets in the workpiece that need to be considered for edge/facet comparison. At the second level, facet/edge pairs that cannot intersect among the remaining facets/edges are filtered out. One way of doing this is by enclosing each facet (or edge) in a box whose sides are aligned with the coordinate axes, and construct a list of intersecting boxes. If two enclosing boxes do not intersect, the facets/edges inside them cannot intersect and need not

be considered together. Karasick [30] reported an algorithm that speeds up polyhedral intersection to almost an $O(n \log n)$ behavior when intersecting several hundred randomly positioned cubes.

Intersecting a polyhedral representation of the removed material volume with the cutting edge can cause tolerance problems. As illustrated in Figure 4.37 (a), the edge may intersect with a boundary of the volume at which a condition of tangency is supposed to exist. Such a condition is automatically satisfied when the exact representation of the removed material volume is used. This problem is encountered with the polyhedral representation because the cylindrical/spherical boundaries of the volume, resulting from cutting with the ball nose of a ball end mill, are approximated with planar faces. This issue can not be dealt with by increasing the number of faces, but one solution is to use a larger than actual volume so that the proper tangency conditions are restored, Figure 4.37(b). It was found that this can be achieved by using the material volume removed by a tool with a radius $R + 0.1$ mm instead of R (the actual radius).

Machining Process	*Geometric Modeling/Simulation*
Part	Polyhedral model
Machined material	Poly. model = workpiece $\cap^*$ swept volume
Cutting edge	Cubic Bezier curve
Machining (In-cut segments)	Poly. model/Cubic Bezier intersection and classification

Table 4.1: Simplified Geometric Approach

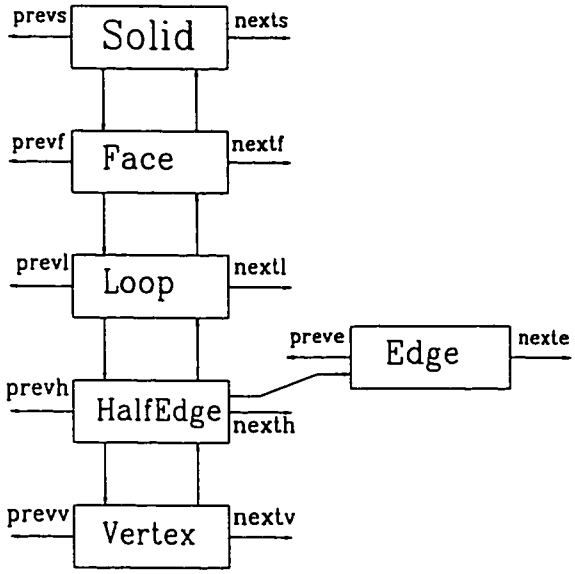


Figure 4.1: Hierarchical view of the Half-edge data structure

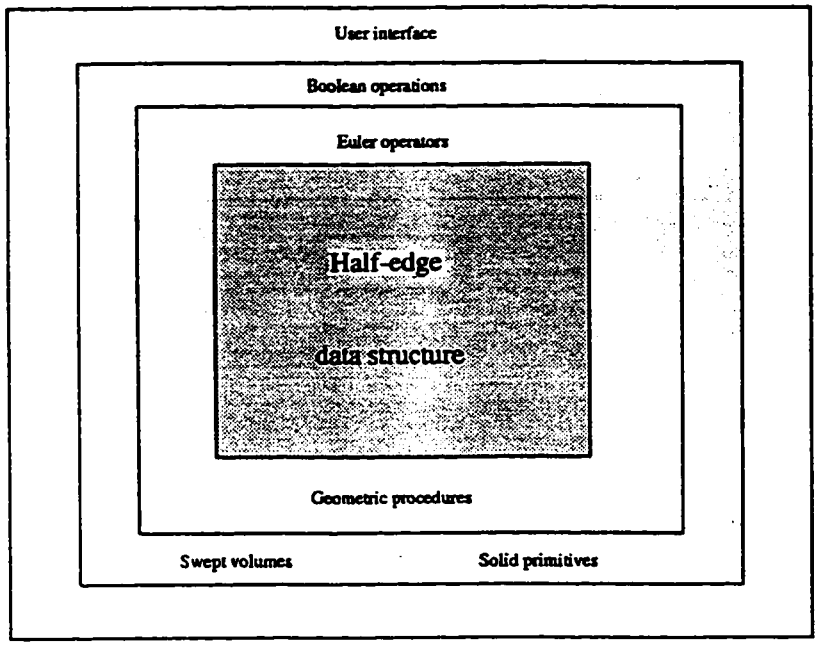


Figure 4.2: Hierarchical layers of GWB

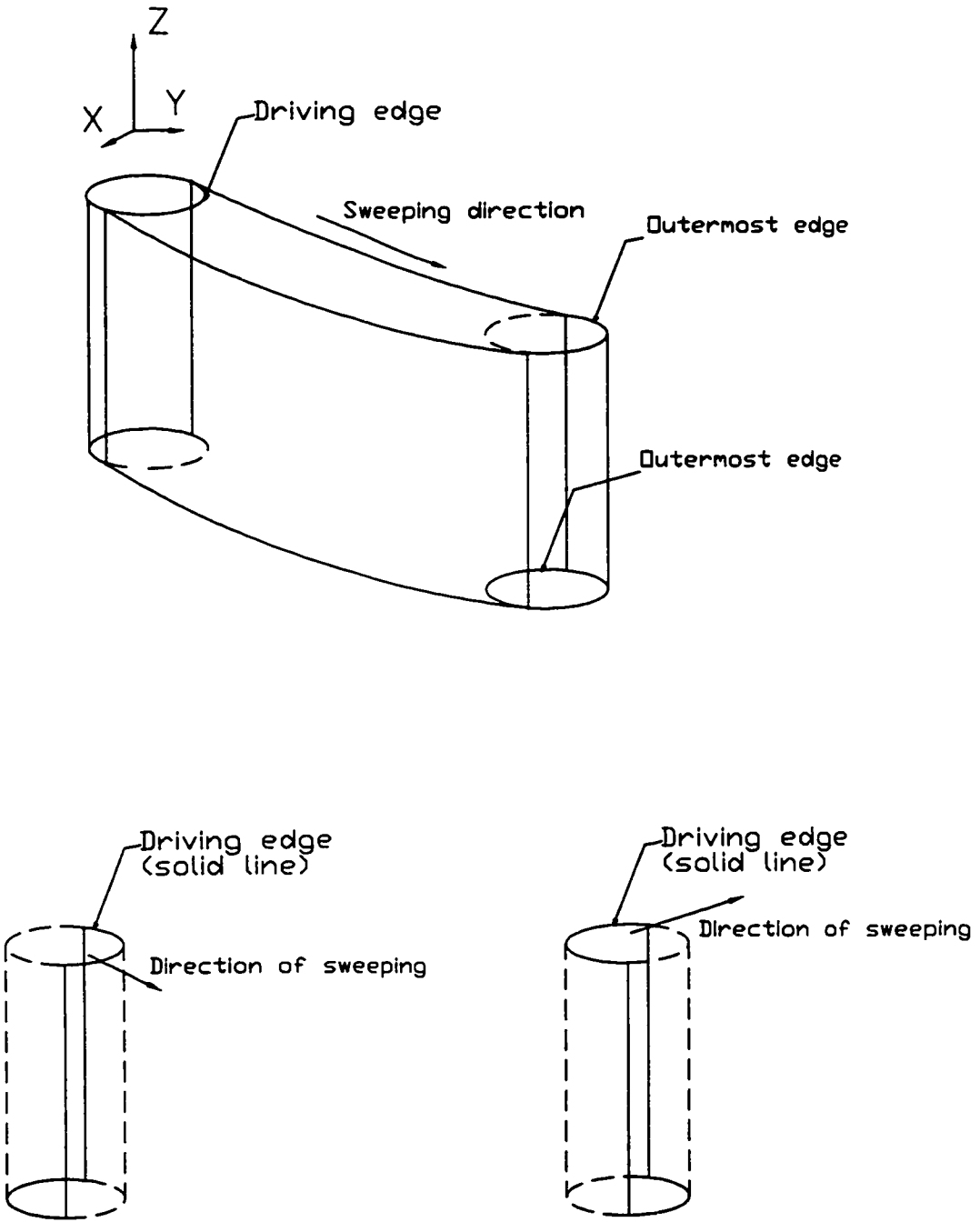


Figure 4.3: (a) Driving edge for the case of a tool with cylindrical shape; (b) Effect of the instantaneous path direction on the driving edge

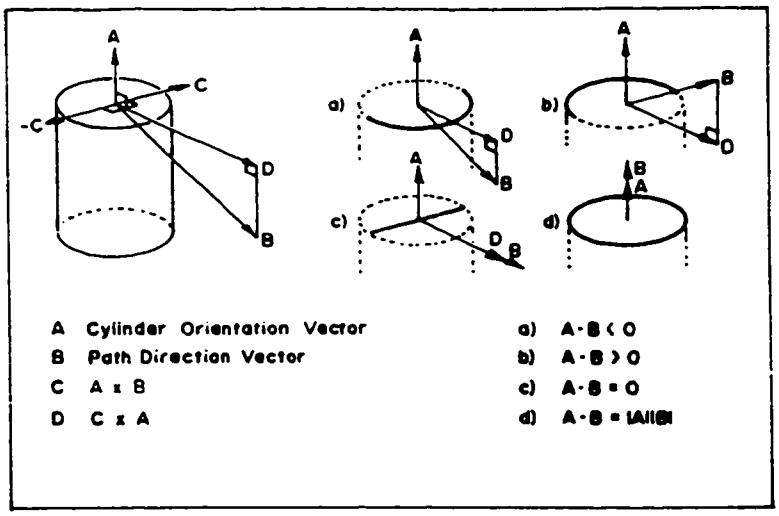


Figure 4.4: Determining the proper driving edge

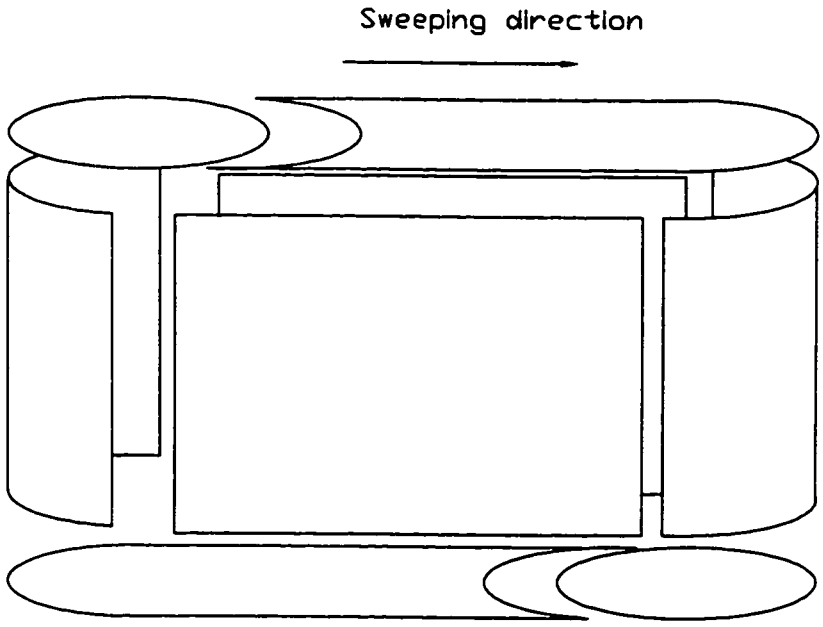


Figure 4.5: Path dependent and path independent surfaces of the sweep envelope

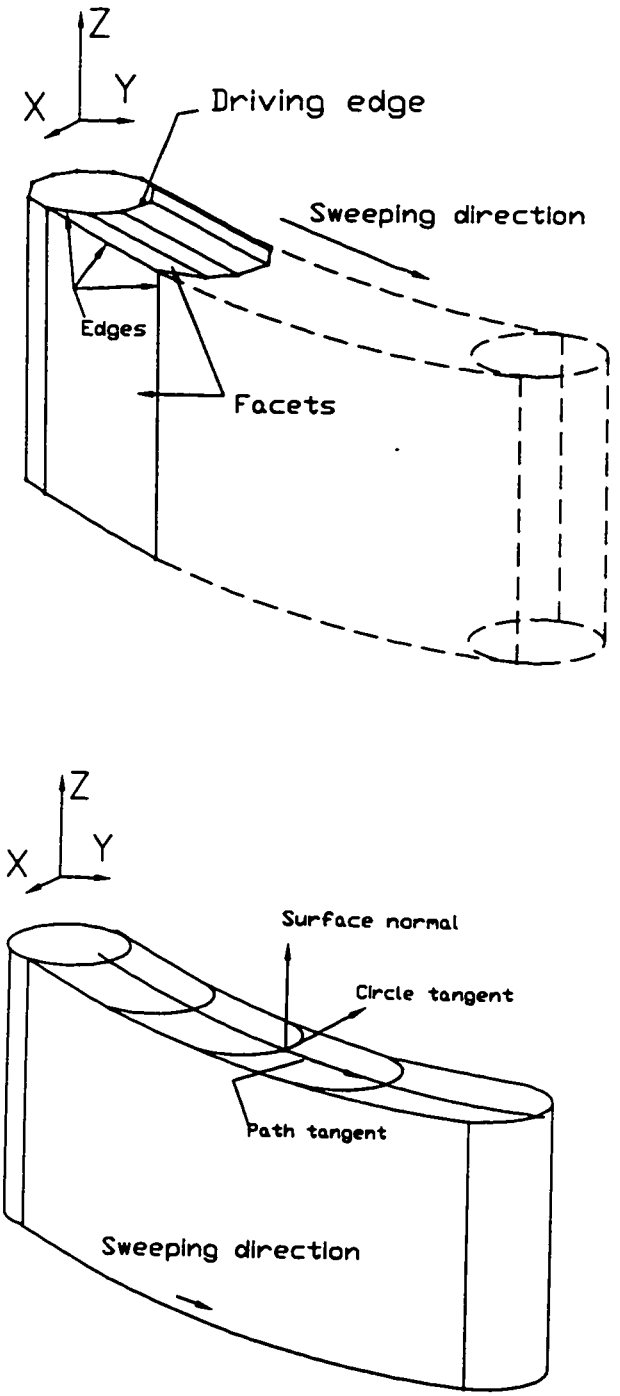


Figure 4.6: (a) Polyhedral representation of path dependent surfaces; (b) Determining the surface normal of path dependent surfaces

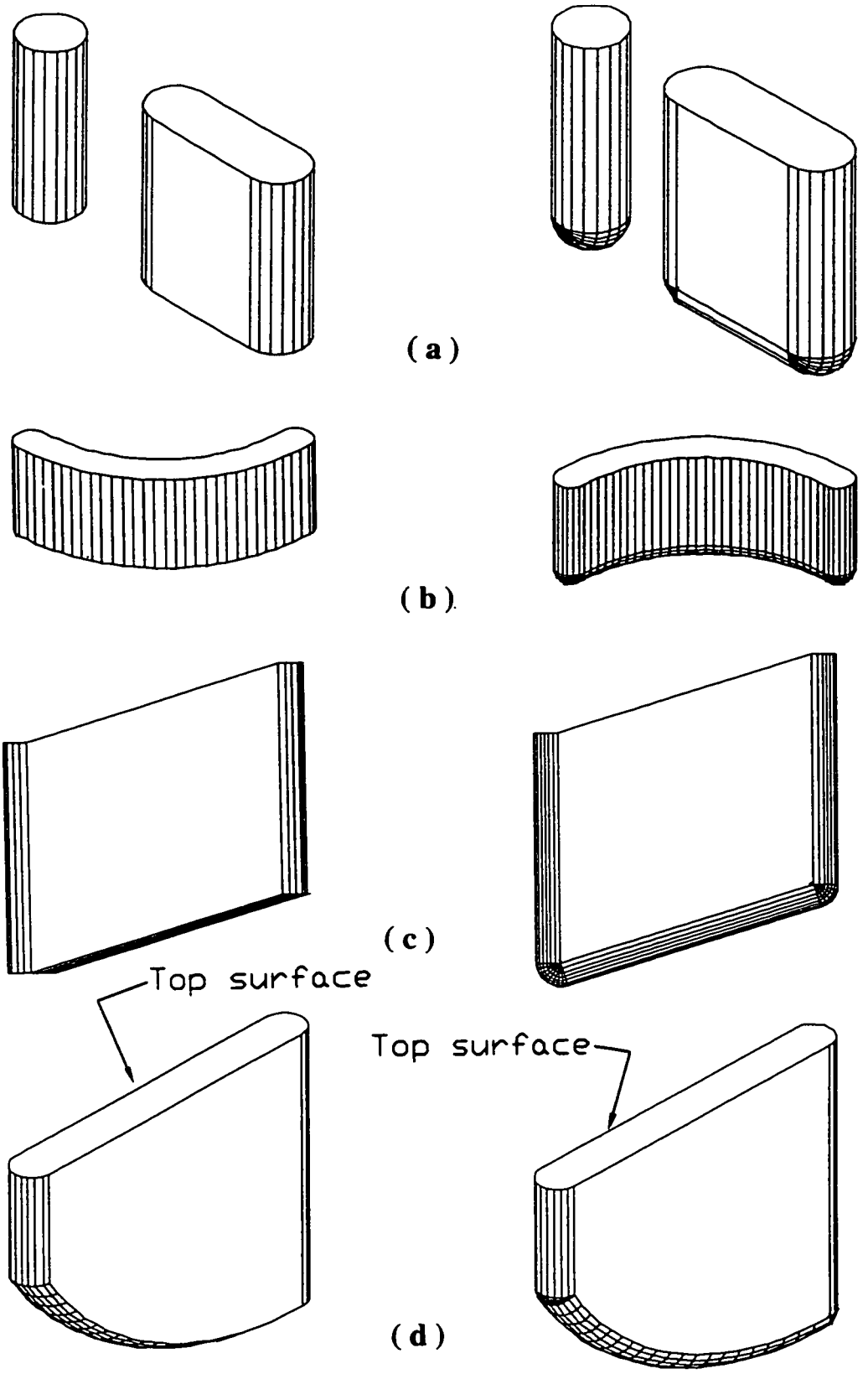
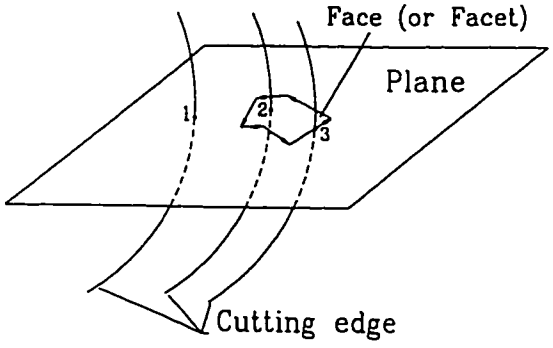


Figure 4.7: Sample of swept volumes for flat and ball end mills along linear and circular interpolations

Control point	Coordinates
b_0	(-0.0277, 0.0000, -12.6680)
b_1	(6.5699, -0.0589, -12.8879)
b_2	(12.4618, 1.4660, -7.1968)
b_3	(12.2342, 3.1736, -0.8239)

Table 4.2: Cutting edge on ball nose of a 1 in diameter ball end mill is fitted with one cubic Bezier segment (4 contol points)



1: intersection outside the face
2: intersection inside the face
3: intersection on the boundary

Figure 4.8: Cutting edge/Plane intersection

Control point	Coordinates
b_0	(6.3500, 0.0000, 0.0000)
b_1	(5.95336, 1.65864, 2.87940)
b_2	(5.55672, 3.31729, 5.75881)
b_3	(4.49013, 4.49013, 8.63822)
b_4	(3.42353, 5.62967, 11.51763)
b_5	(1.68697, 6.3500, 14.39703)
b_6	(0.0000, 6.3500, 17.27644)

Table 4.3: The part of the cutting edge on a 1 in diameter flat end mill and given by equation 4.4 is fitted with two piecewise continuous cubic Bezier segments (7 control points)

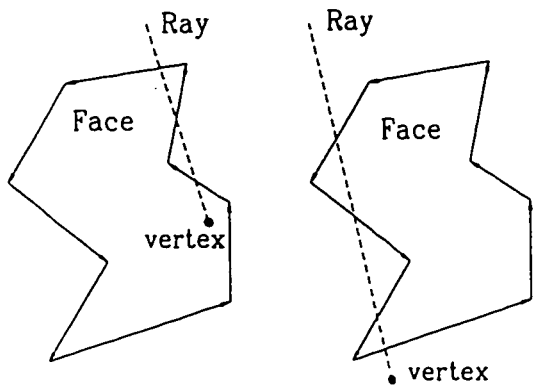


Figure 4.9: Intersection/Facet classification

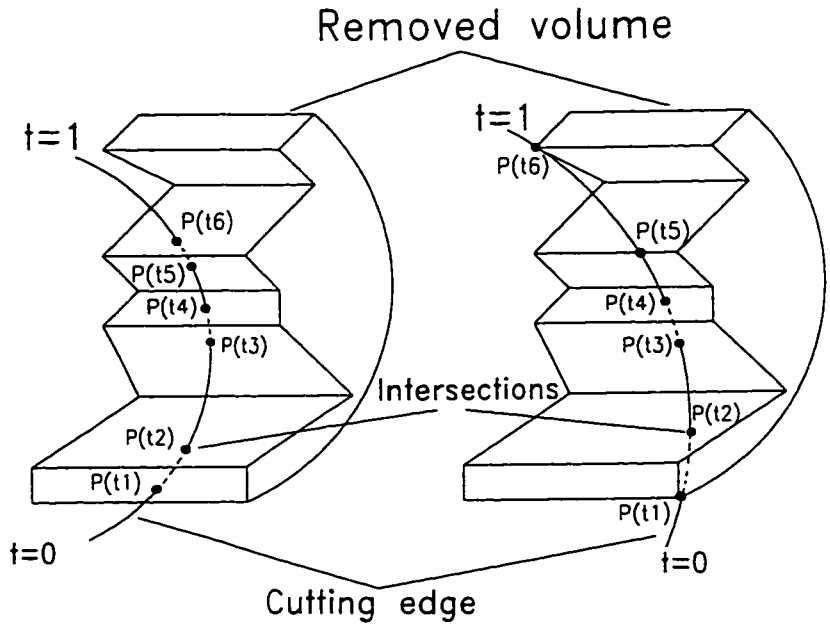


Figure 4.10: Cutting edge/Removed material volume classification

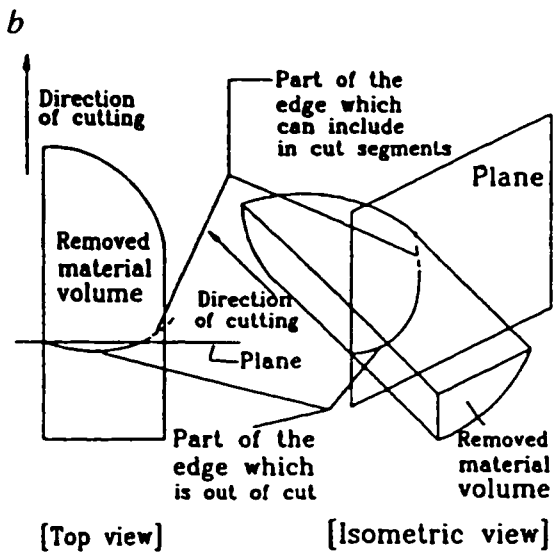


Figure 4.11: Plane that sets the boundary between tool/part contact area and the removed volume

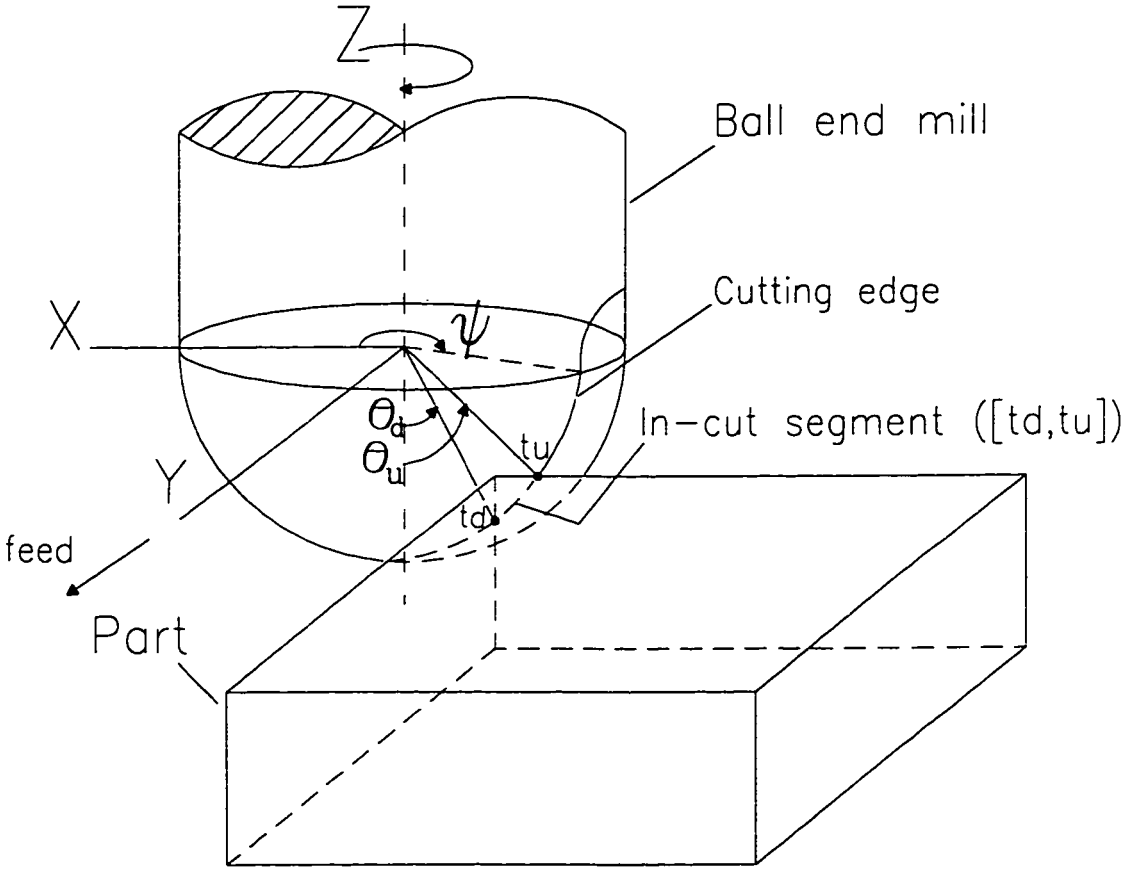
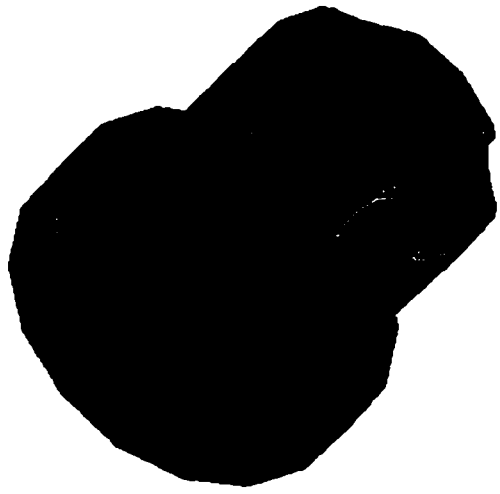


Figure 4.12: Cutting edge on ball nose of ball end mill



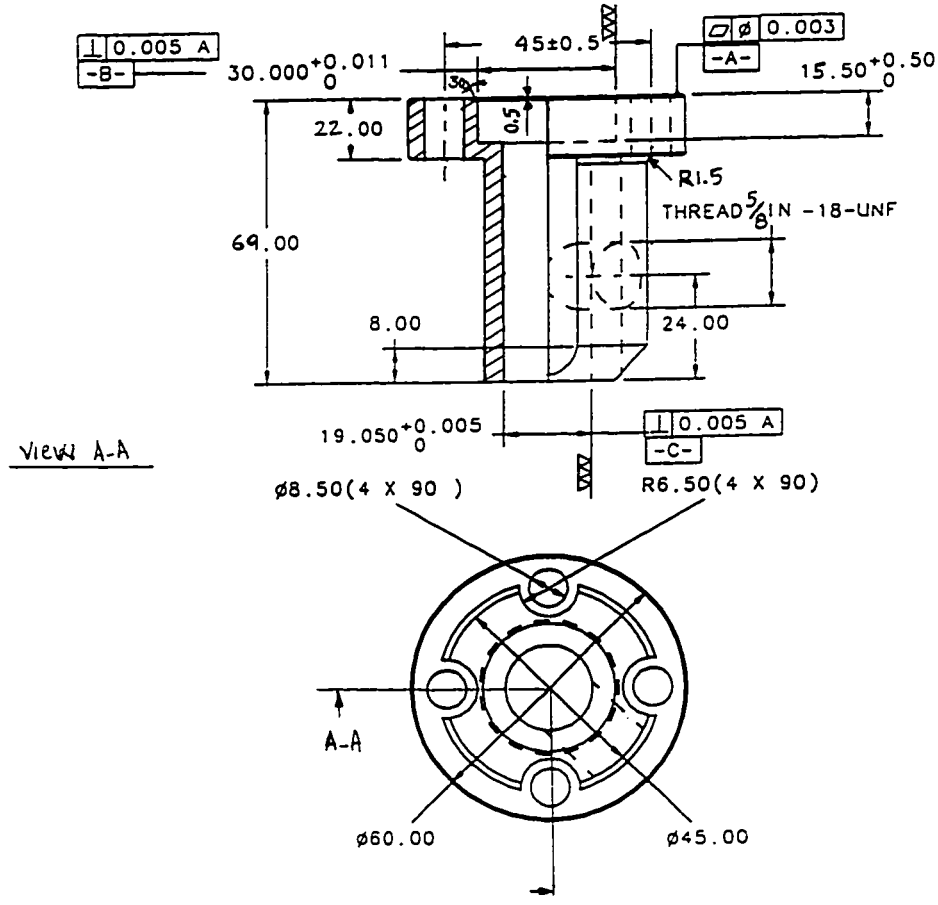


Figure 4.13: Tool adaptor designed to mount tool to spindle dynamometer

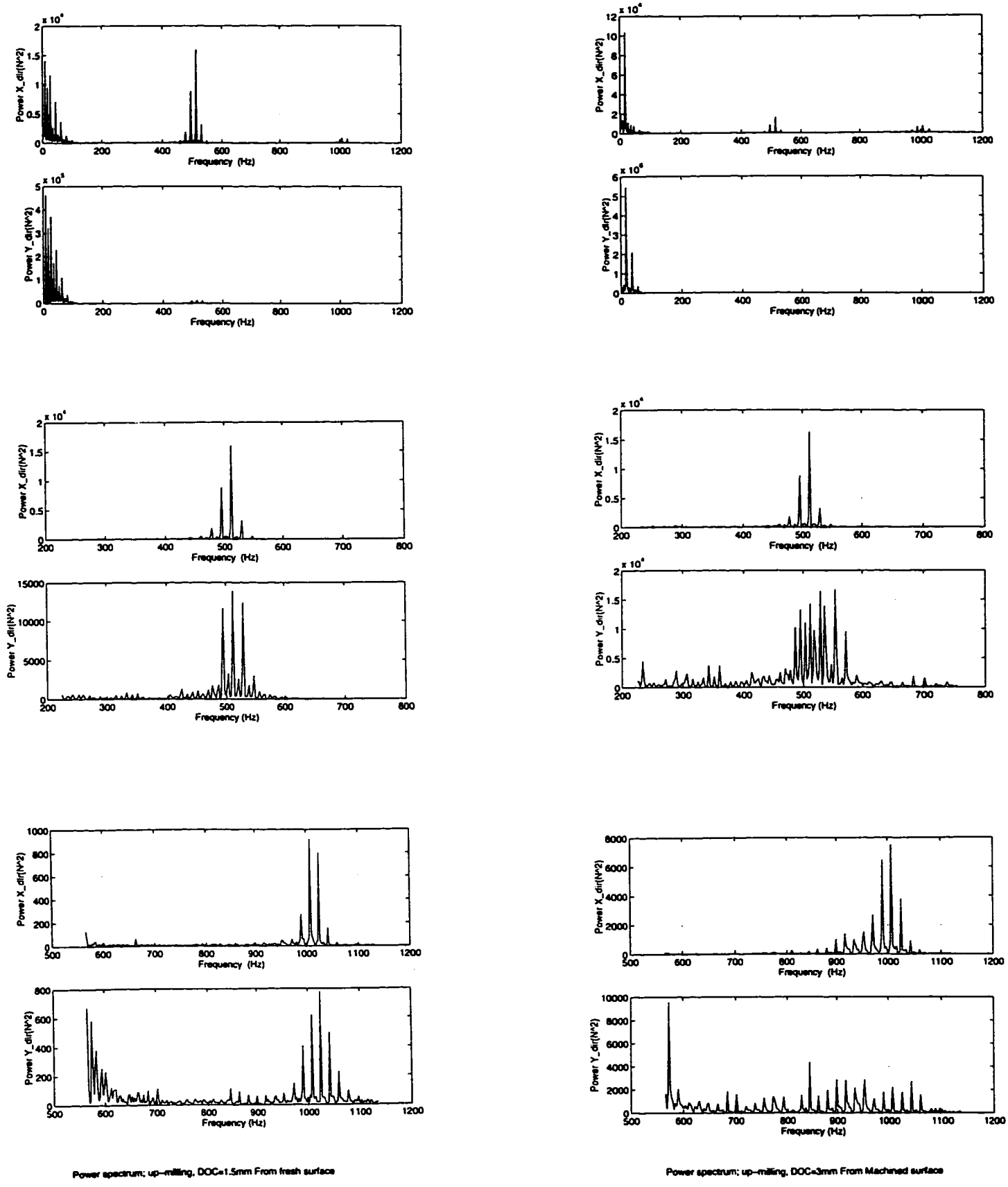


Figure 4.14: High power frequencies in measured cutting forces

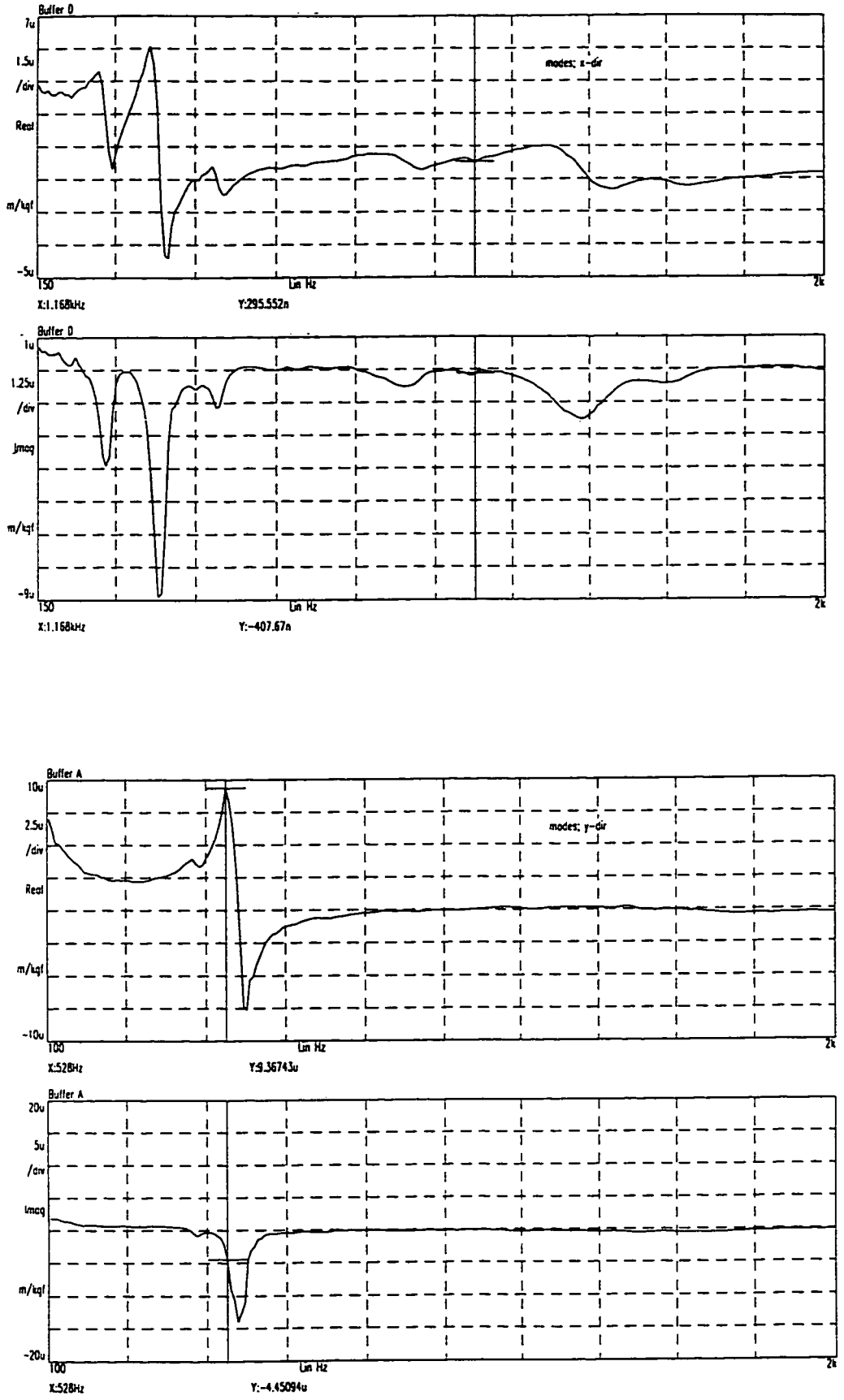


Figure 4.15: Measured Frequency response of tool and tool holder structure

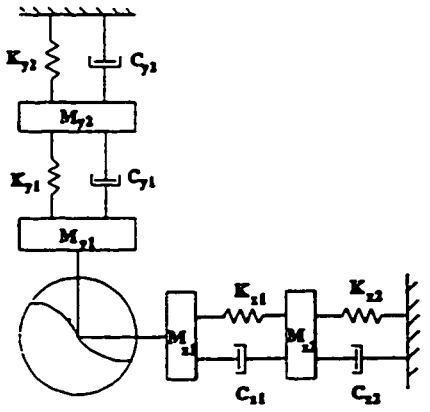


Figure 4.16: Vibratory model for simulating the cutting process dynamics

Modes / Parameters	Mode 1 (x-direction)	Mode 2 (x-direction)	Mode 1 (y-direction)	Mode 2 (y-direction)
f (Hz)	313	437	456	553
ζ	0.051	0.033	0.018	0.0389
C (kg/s)	3047	102.42	12118	94.71
K (N/m)	5.37E7	5.66E6	6.79E8	8.0E6

Table 4.4: Dominant modes in the range $[0, 1100]Hz$

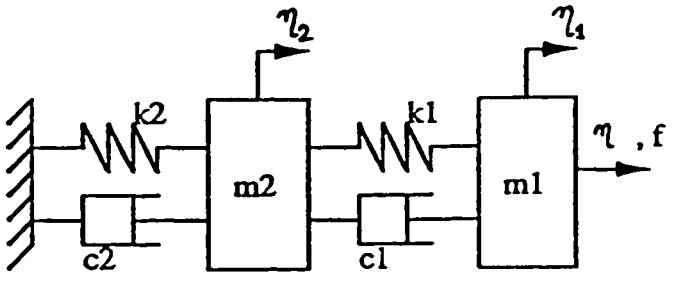
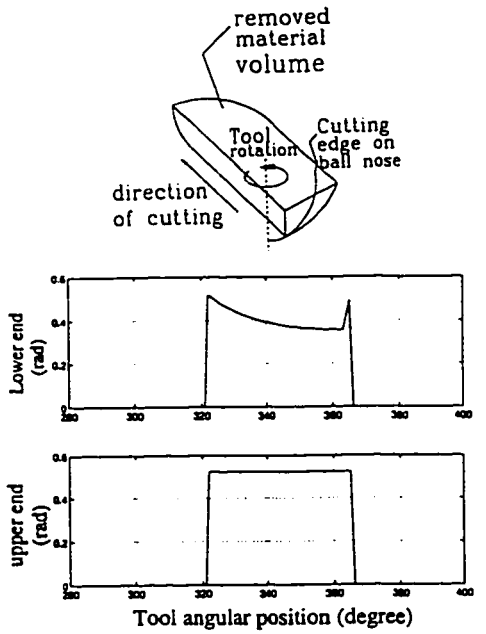
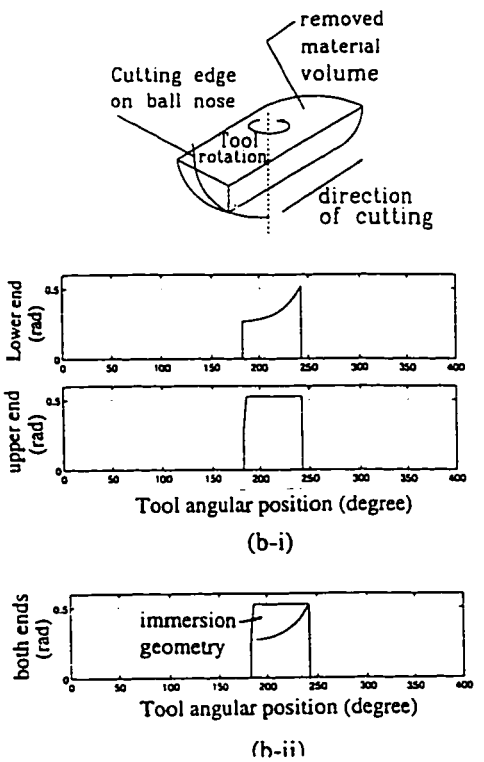


Figure 4.17: System used to predict the cutting forces in any of the two directions



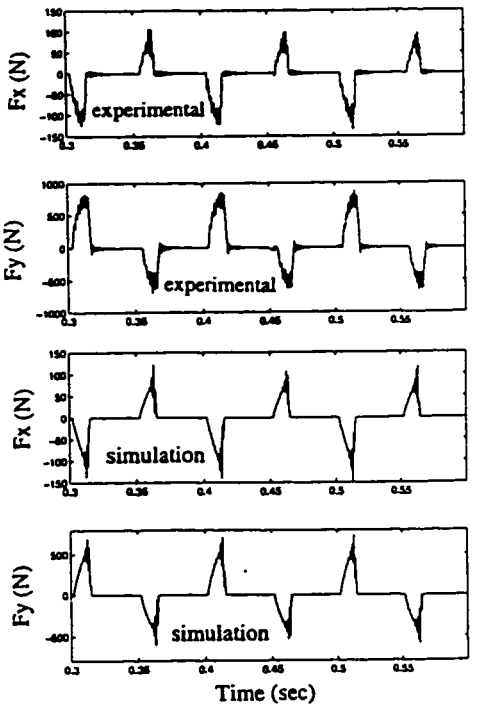
(a)



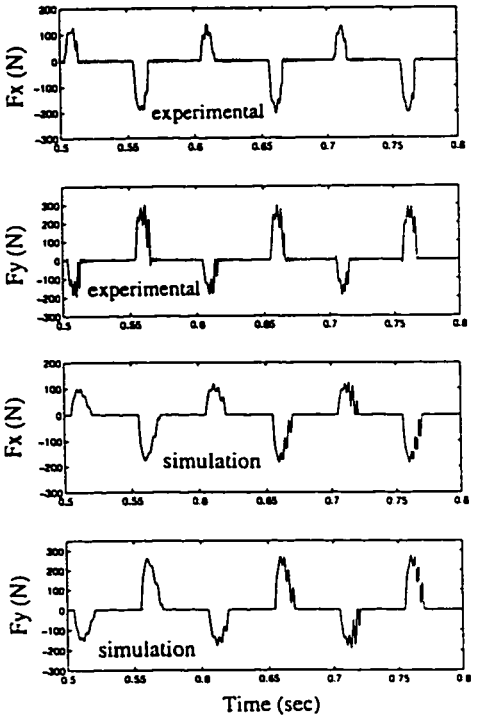
(b-i)

(b-ii)

Figure 4.18: Cutter/part immersion geometry for 2D linear Down-milling and Up-milling

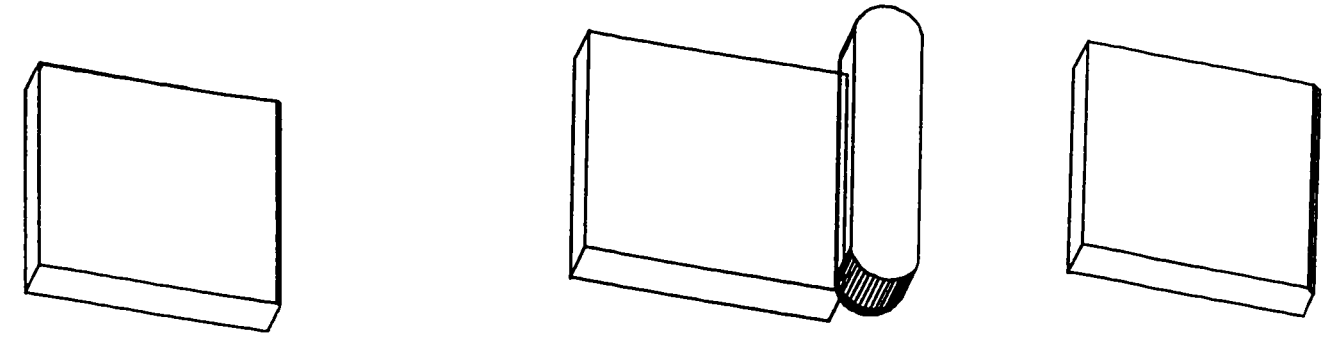


(a)

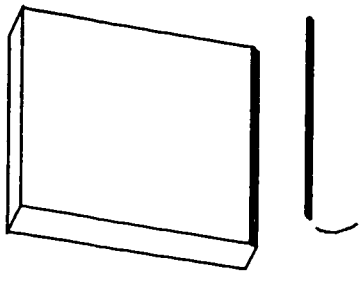


(b)

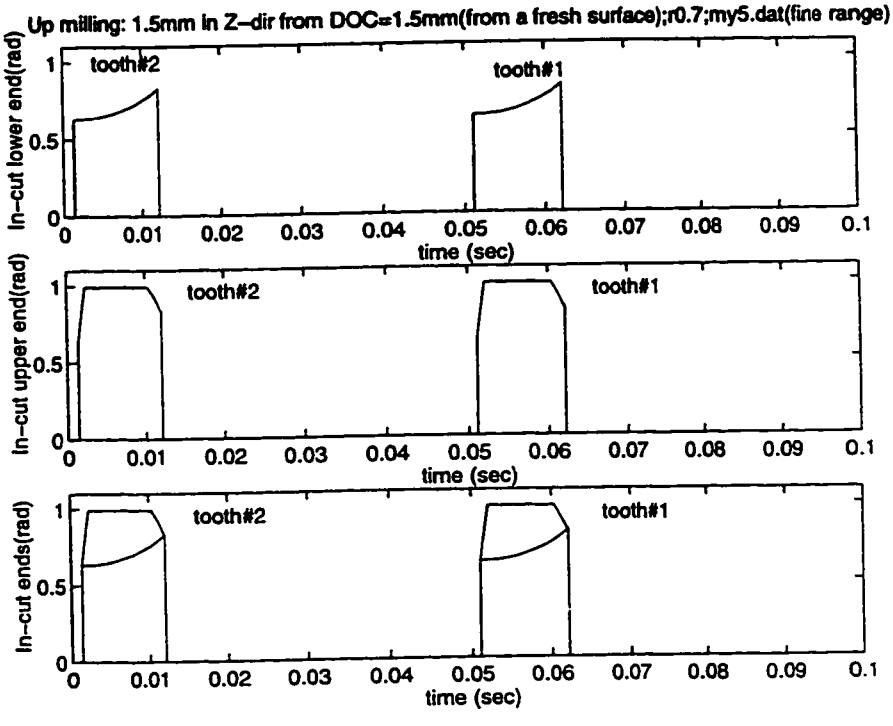
Figure 4.19: Cutting forces in Down-milling and Up-milling



(a) (b)



(c)



(d)

Figure 4.20: Geometric simulation of a case of linear interpolation starting from a machined surface

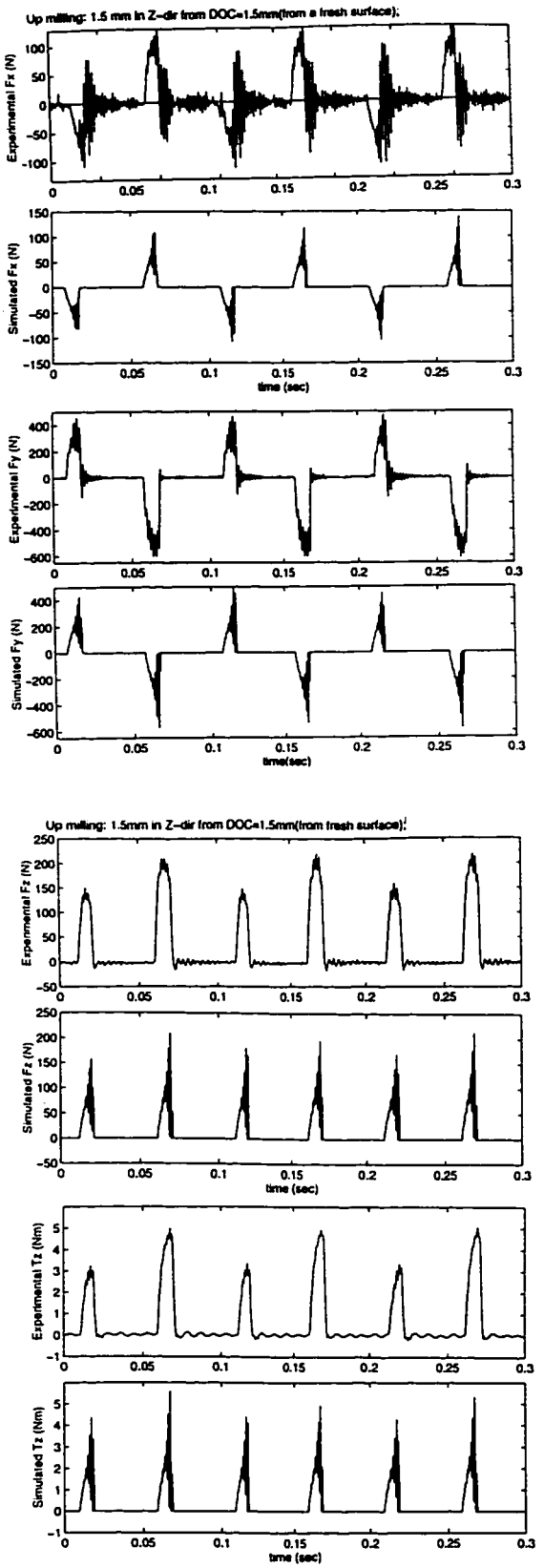


Figure 4.21: Comparison of experimental and simulated forces in the case of linear interpolation from a machined surface

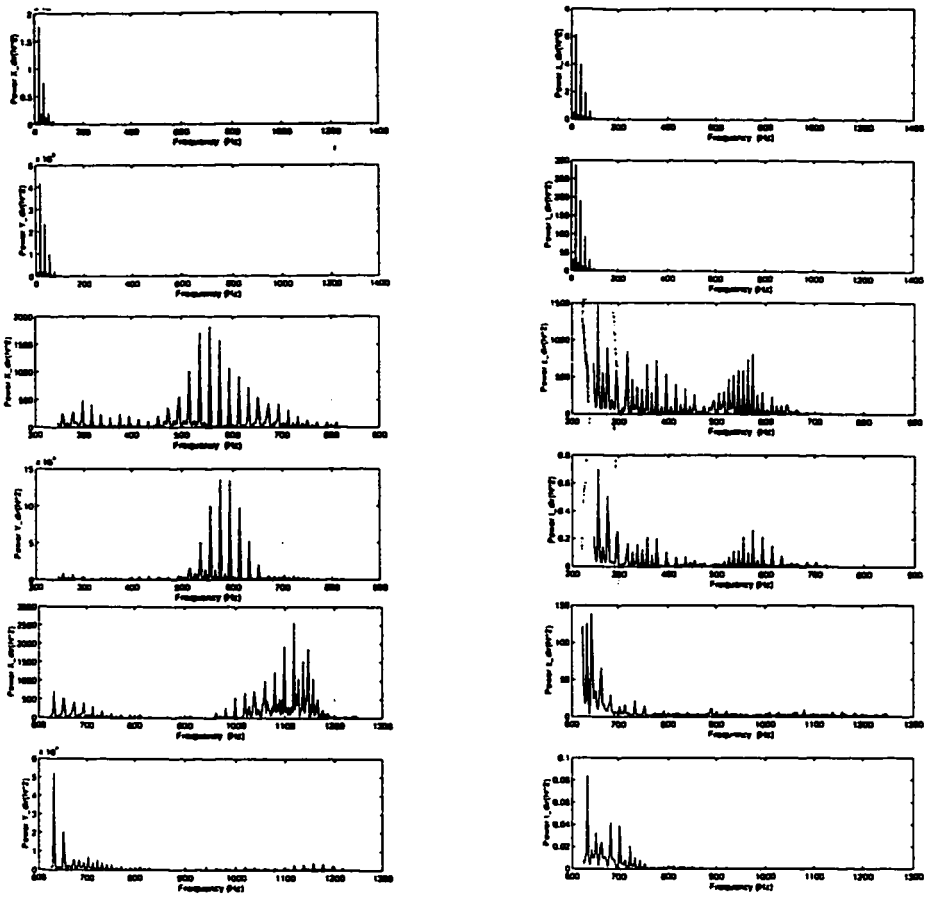
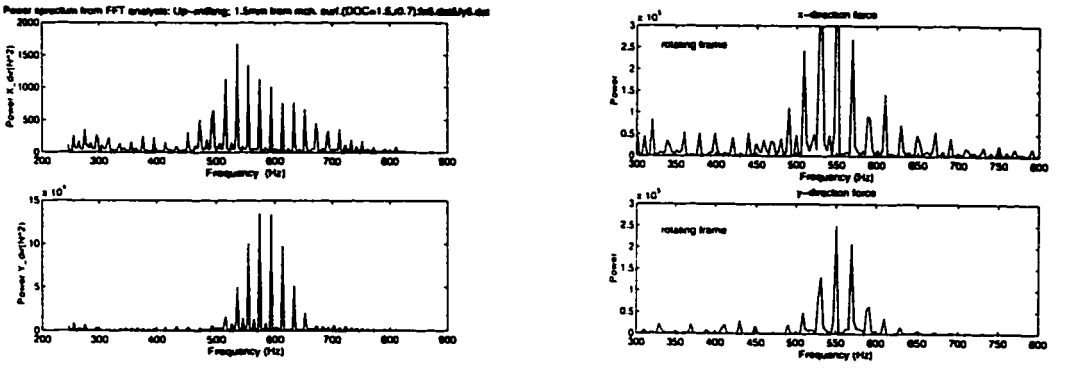
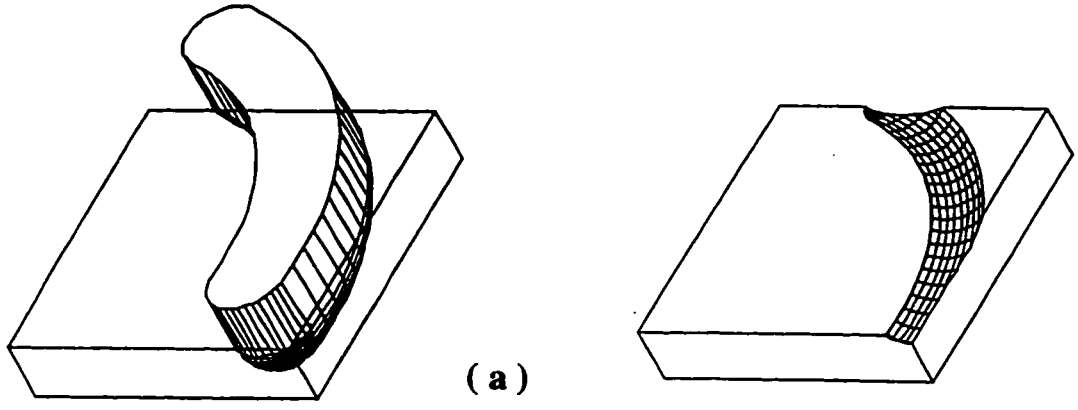


Figure 4.22: Power sprectum of the cutting forces for the case of linear interpolation from machined surface

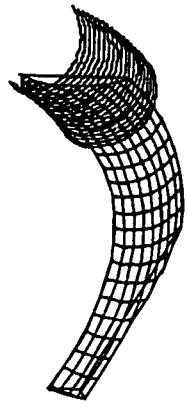


Experimental **Simulation**

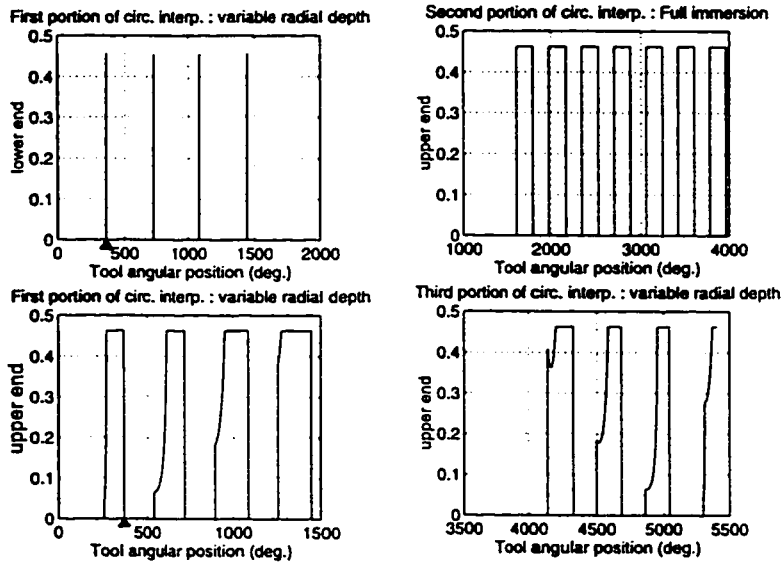
Figure 4.23: Power sprectum of the cutting forces: Experimental v.s. Simulation



(a)

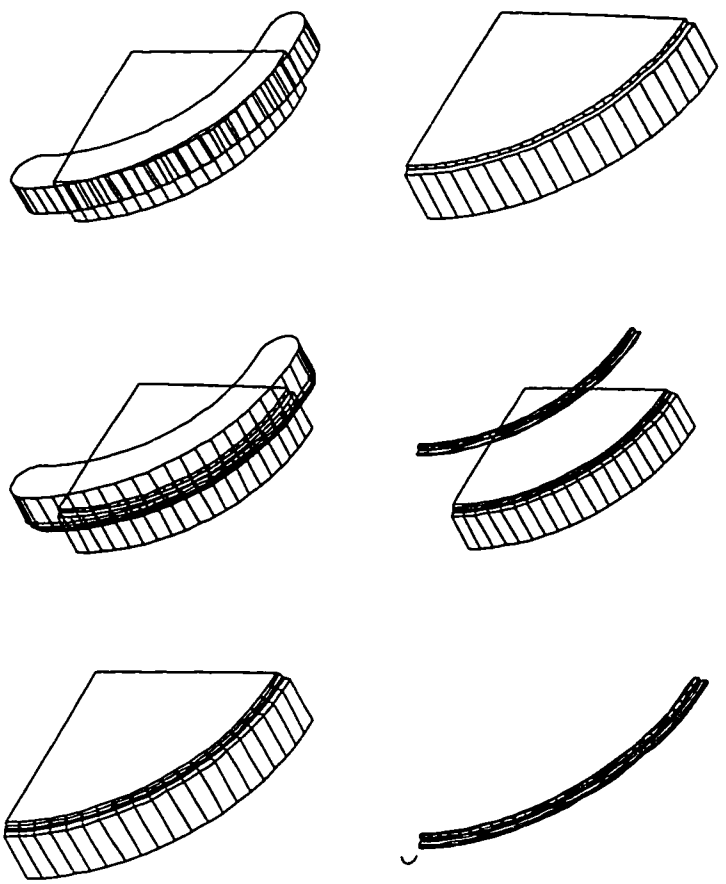


(b)



(c)

Figure 4.24: Cutter/part intersection for a 2D circular interpolation



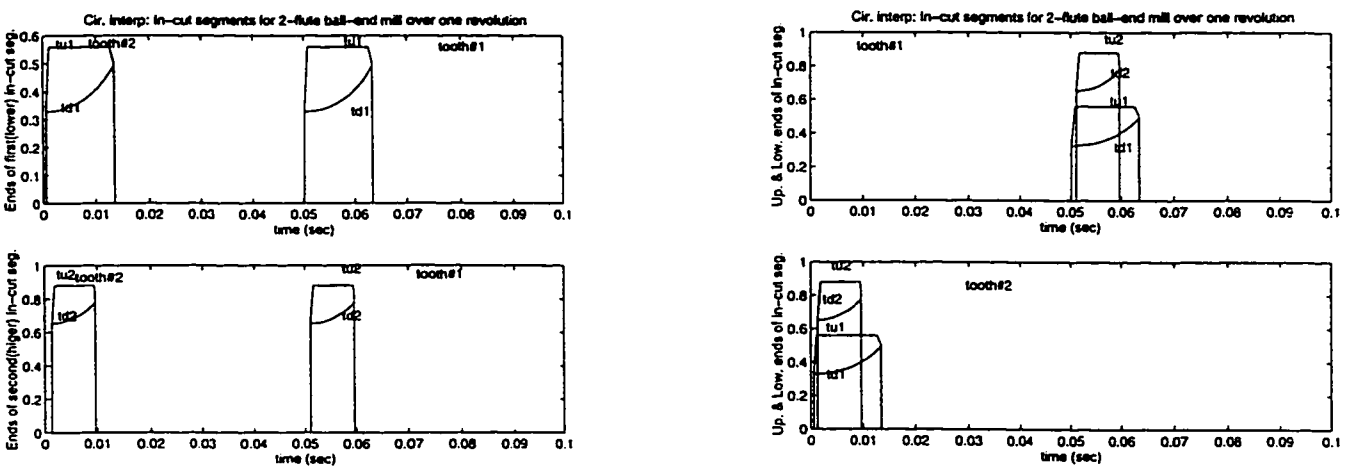


Figure 4.25: 2D circular interpolation with multiple in-cut segments: Geometric information

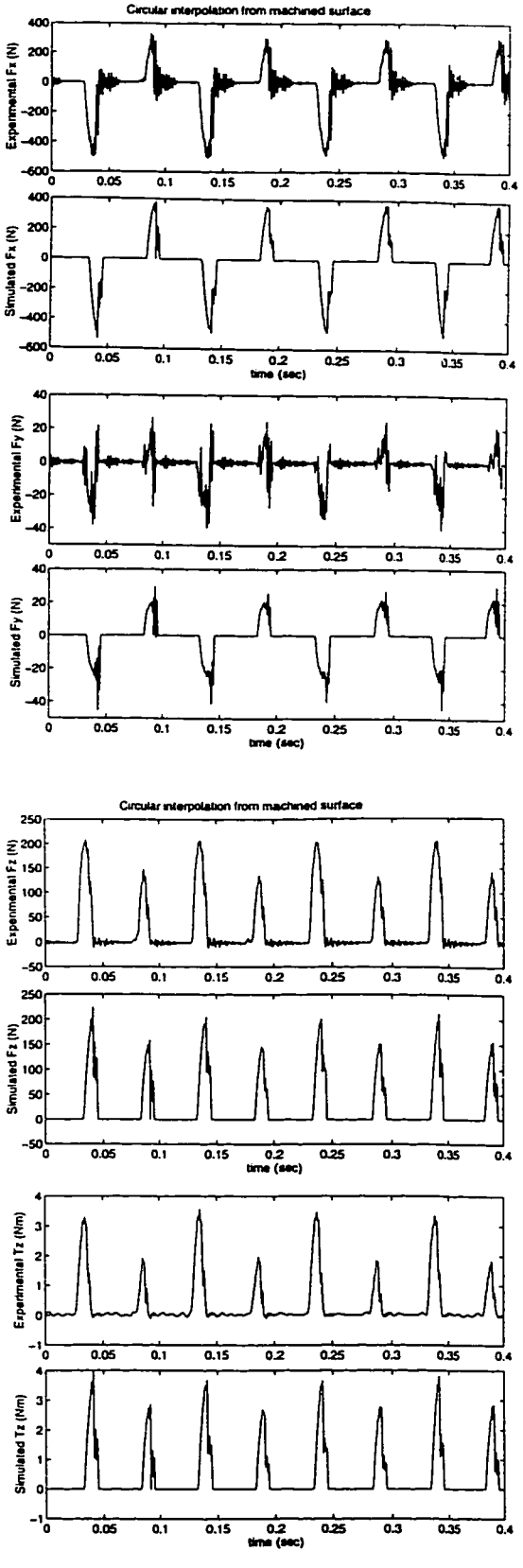


Figure 4.26: 2D circular interpolation with multiple in-cut segments: Physical information

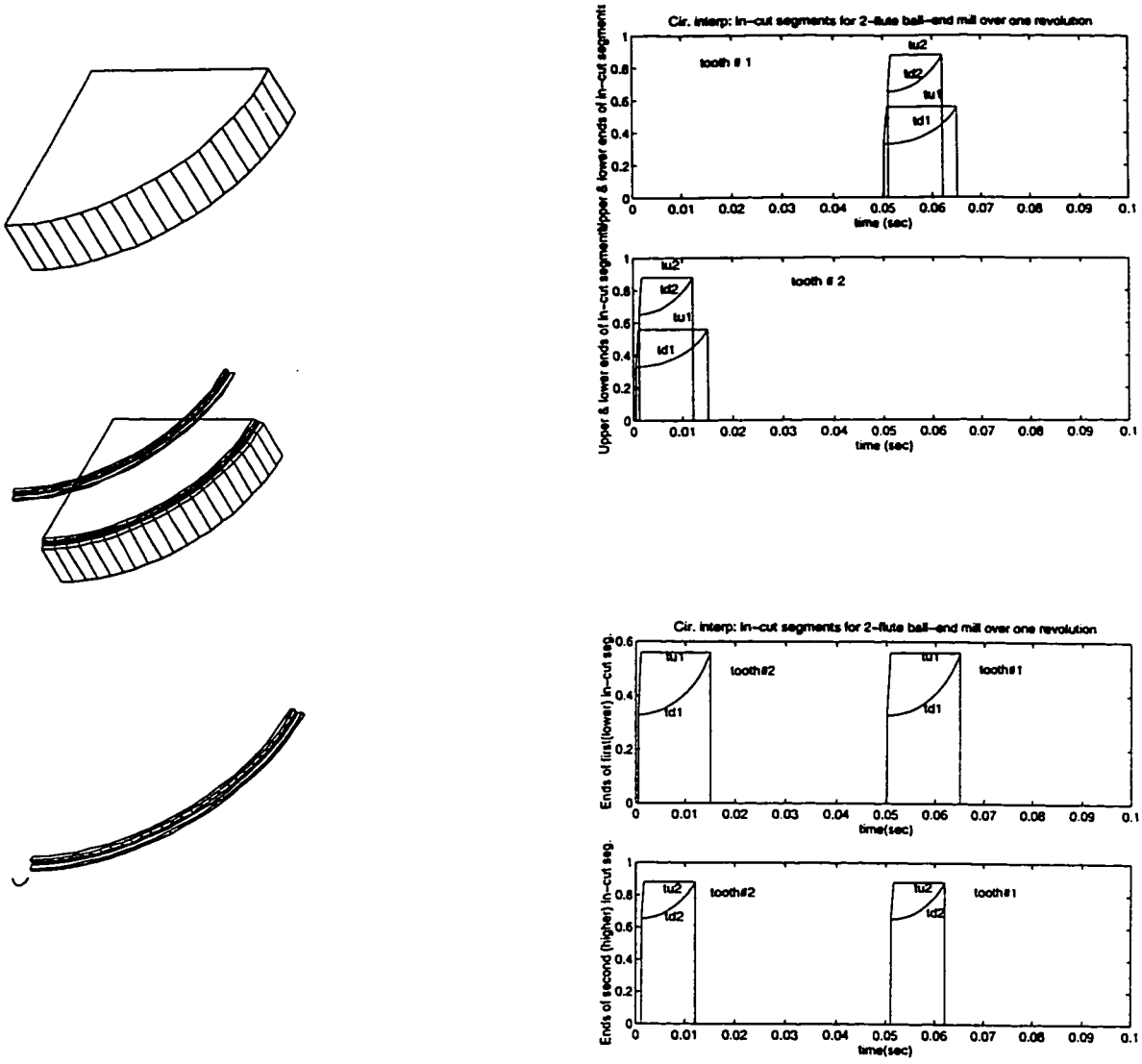


Figure 4.27: 2D circular interpolation with multiple in-cut segments: Geometric information

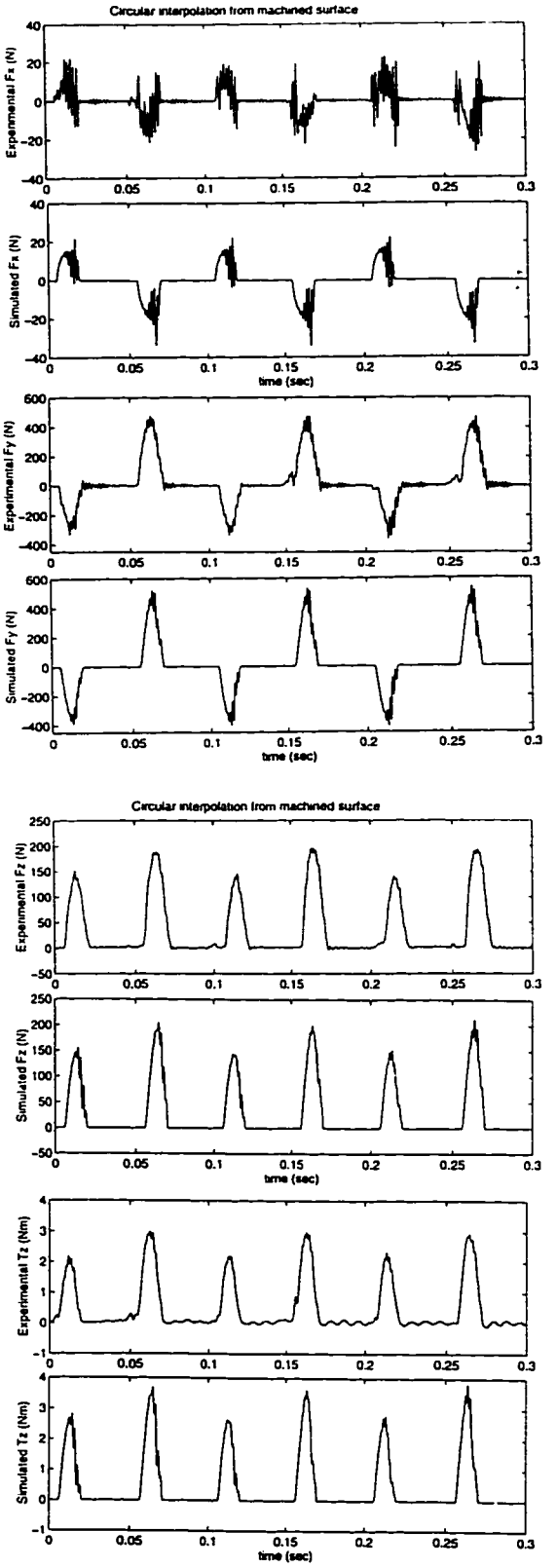


Figure 4.28: 2D circular interpolation with multiple in-cut segments: Physical information

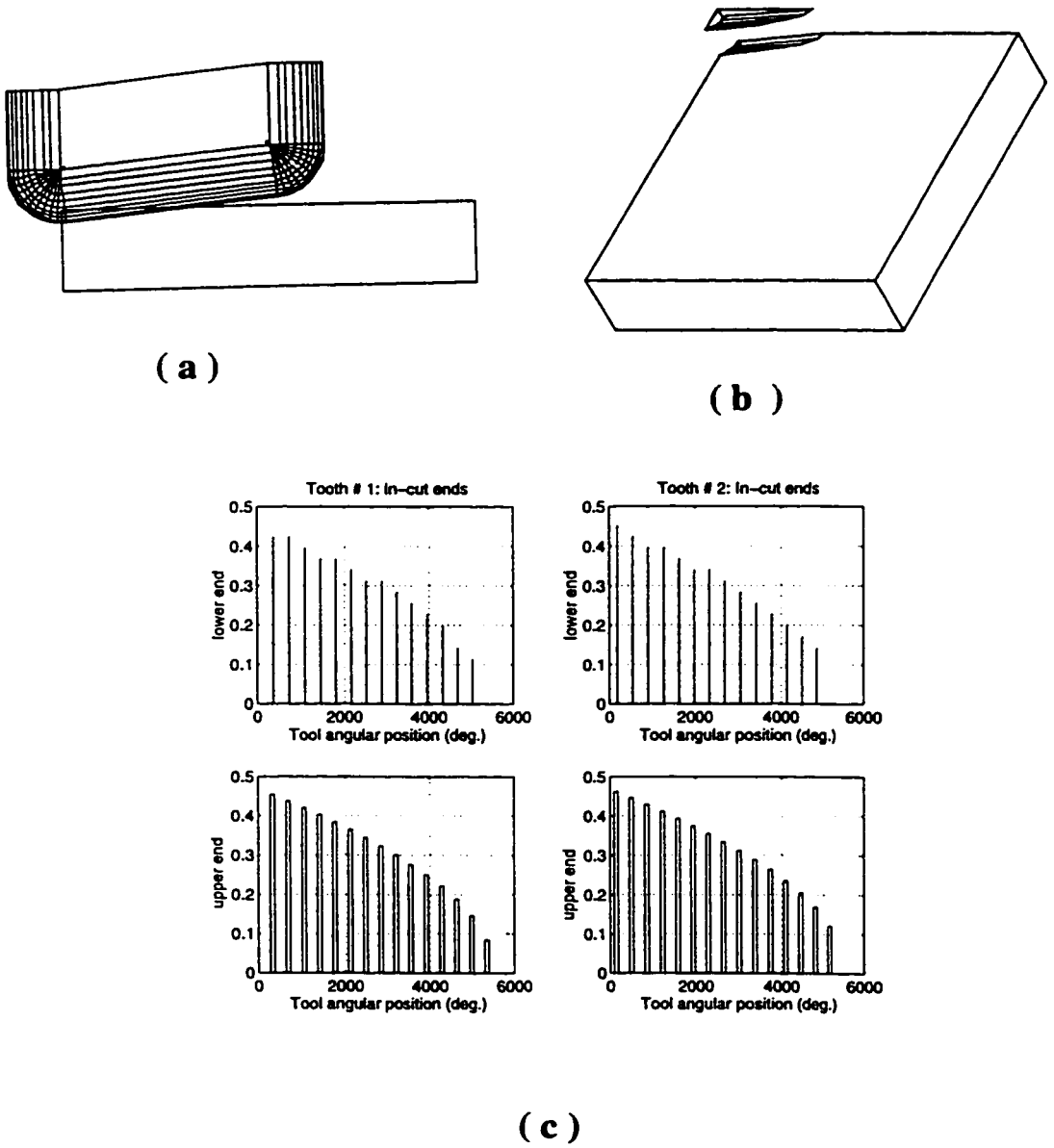
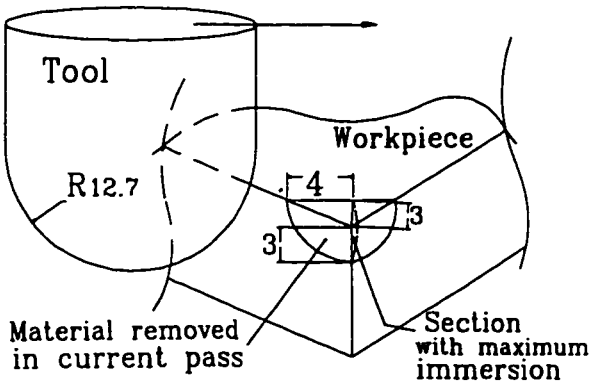


Figure 4.29: 3D linear interpolation: Geometric information



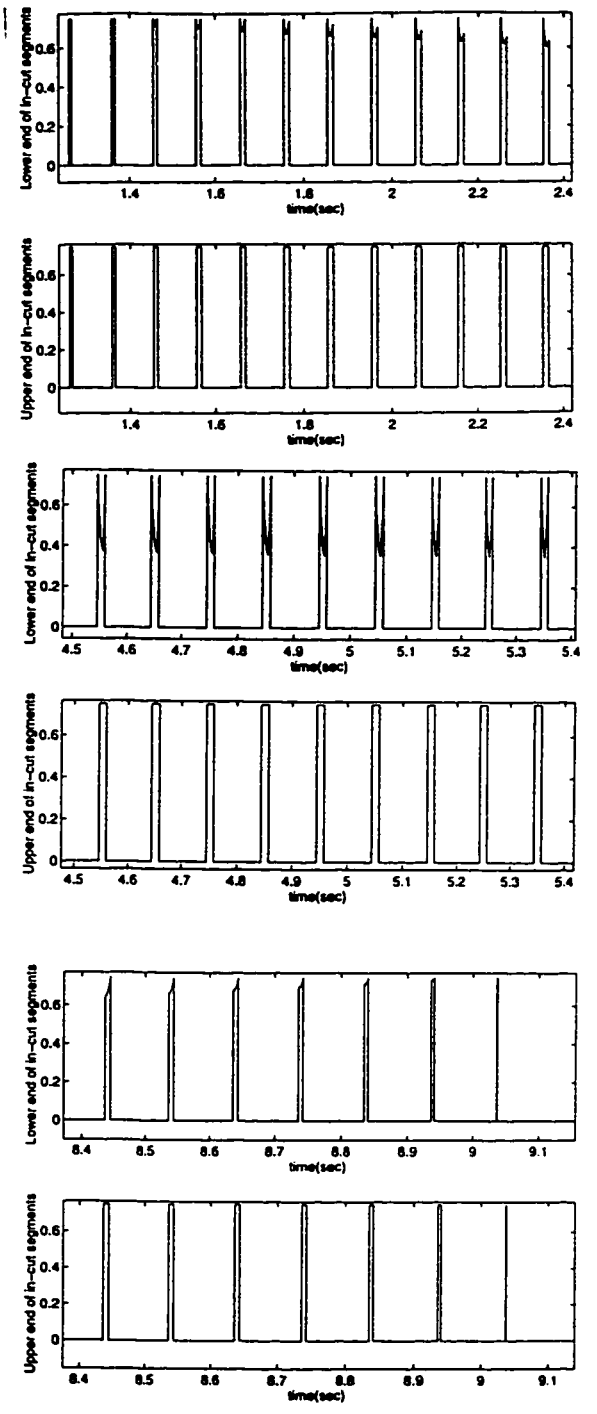


Figure 4.30: Linear interpolation with variable immersion: Geometric information

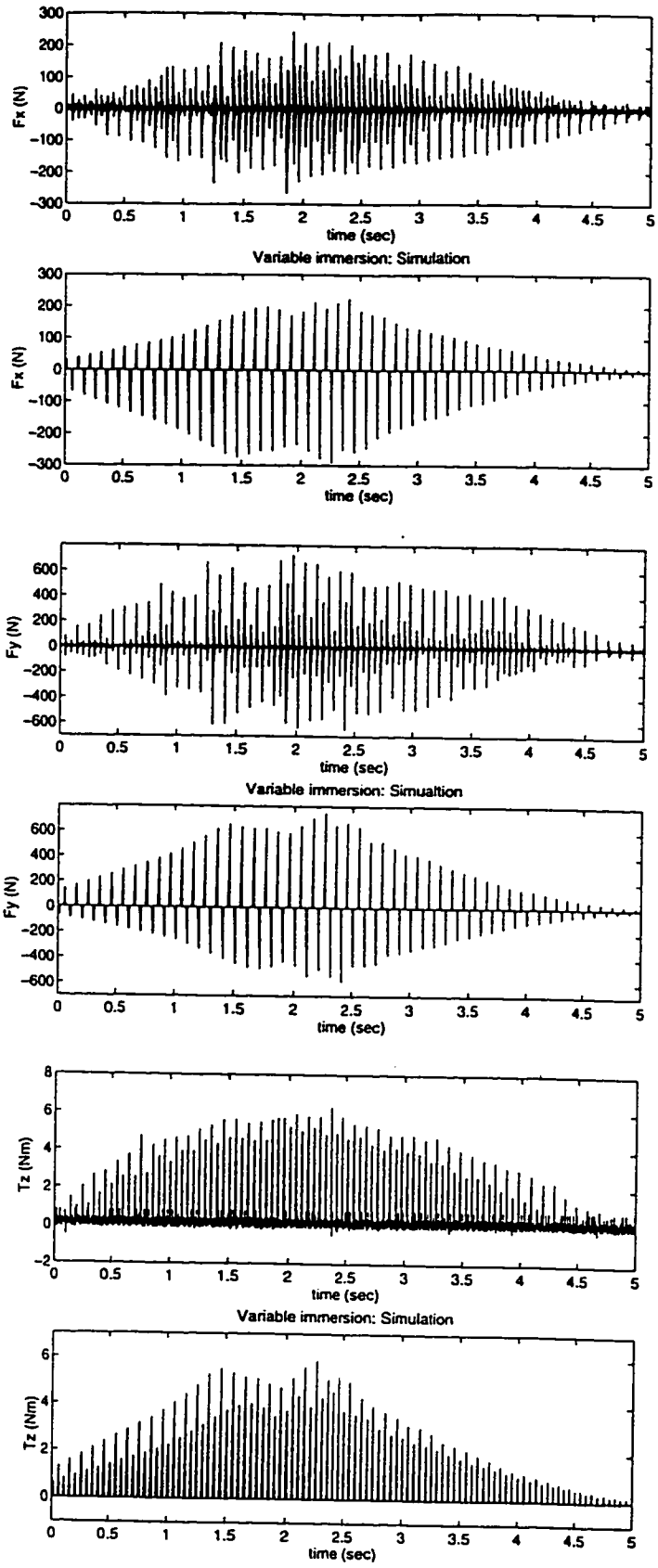


Figure 4.31: Linear interpolation with variable immersion: Physical information

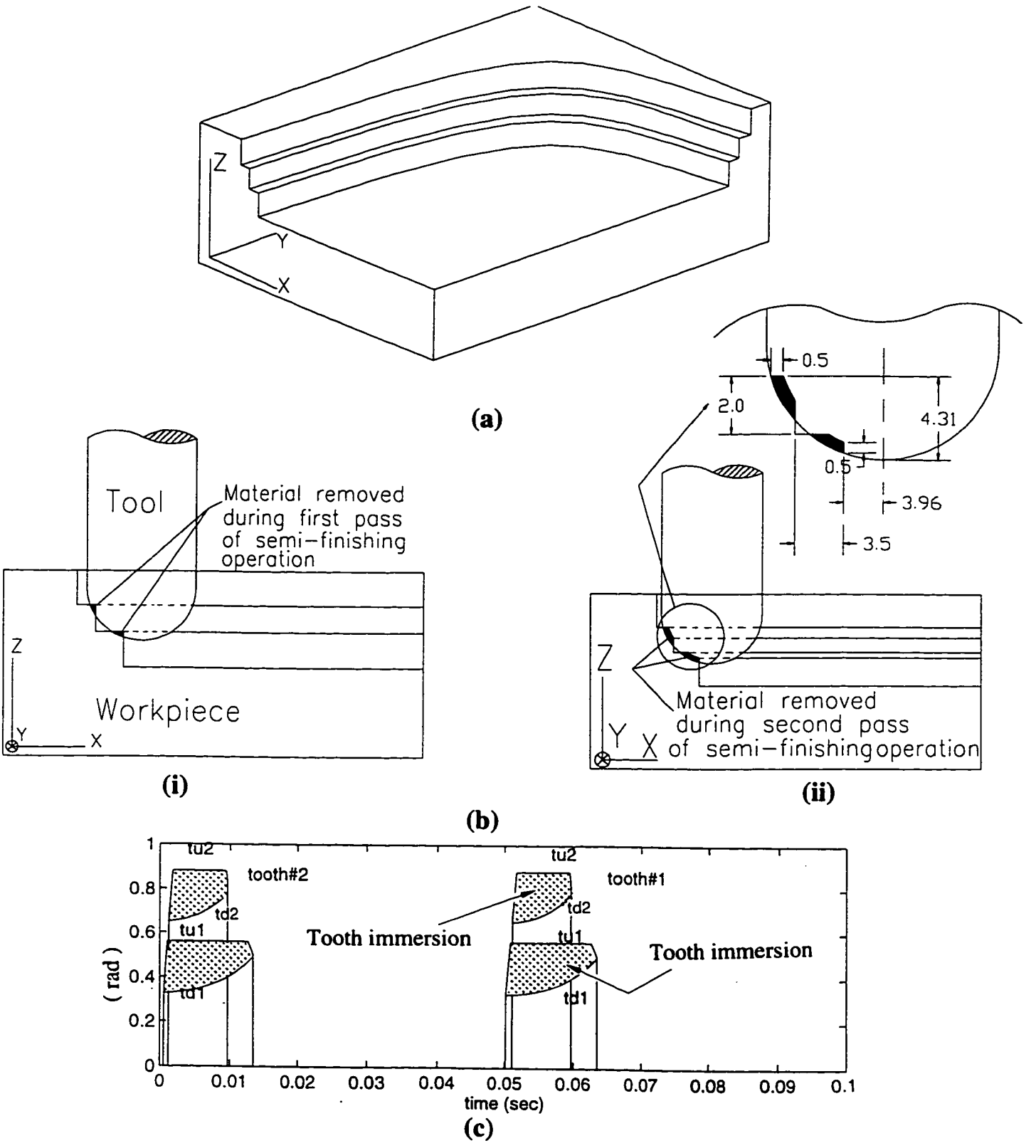


Figure 4.32: Semi-finishing of a die: Geometric simulation (dimensions in mm)

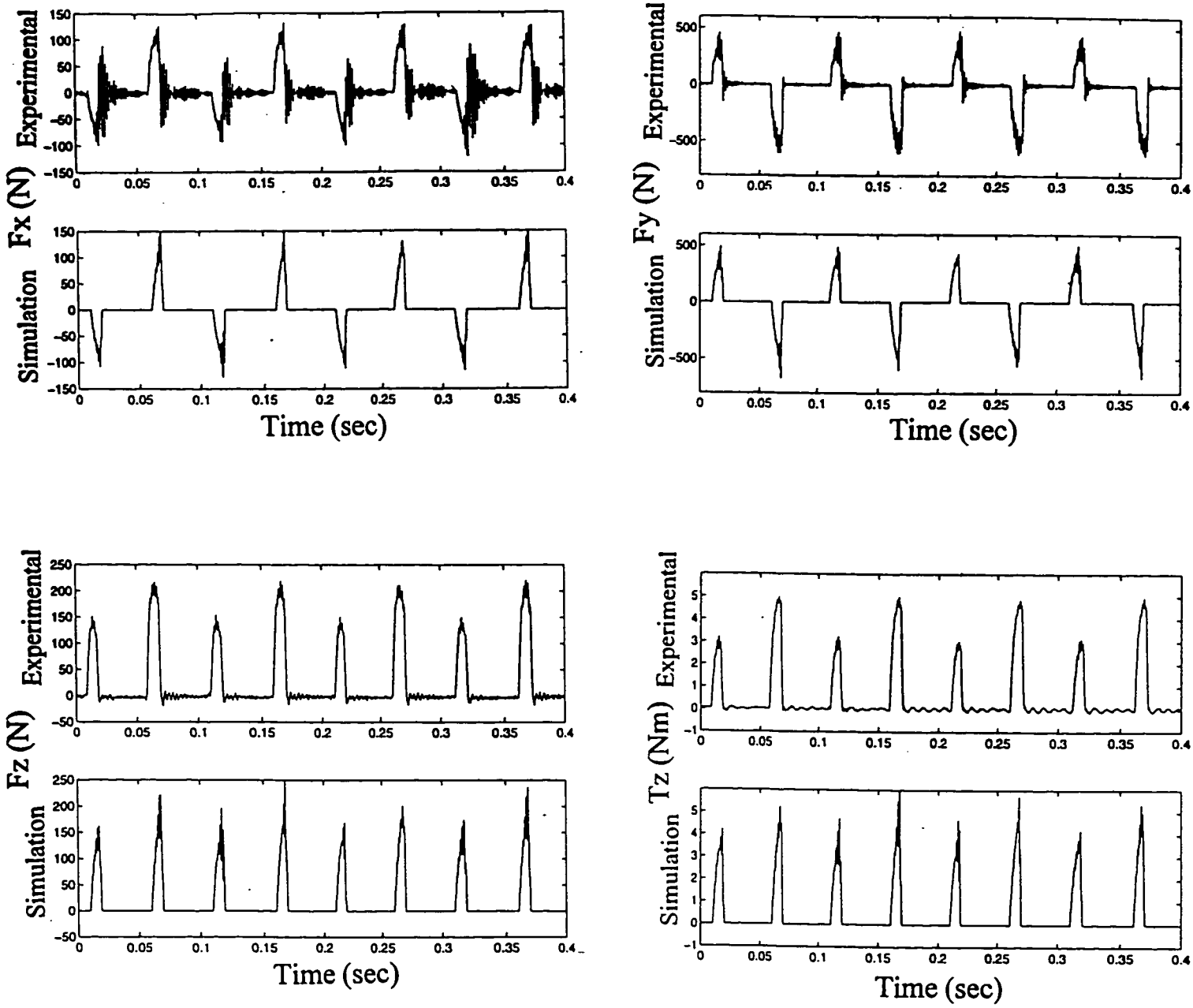
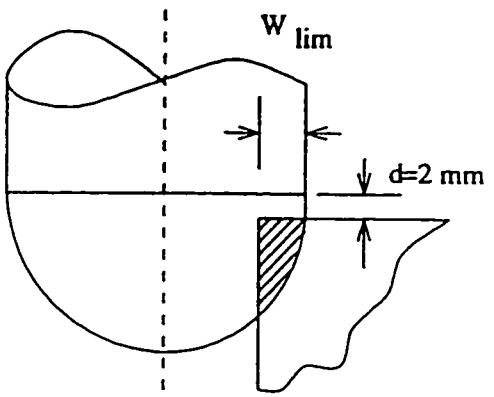
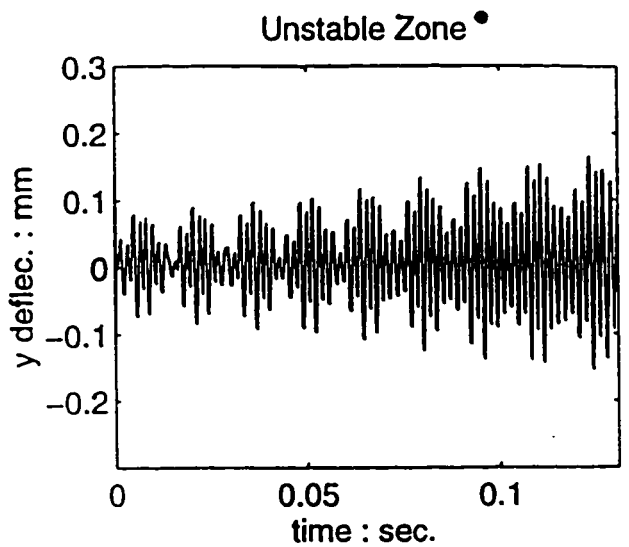


Figure 4.33: Semi-finishing of a die: Physical simulation





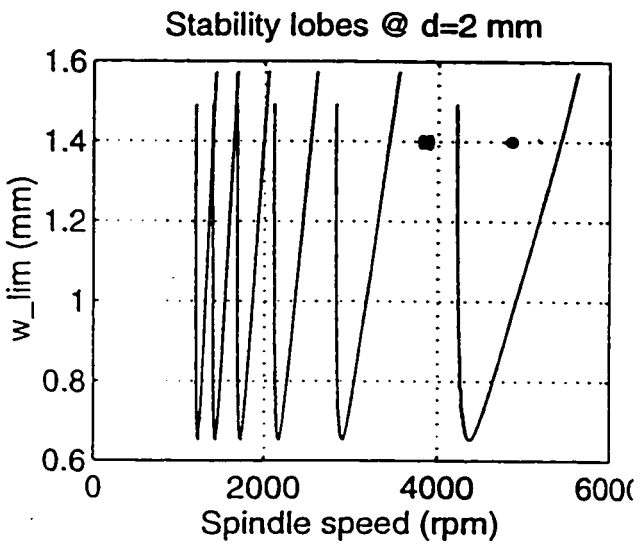


Figure 4.34: Chatter prediction

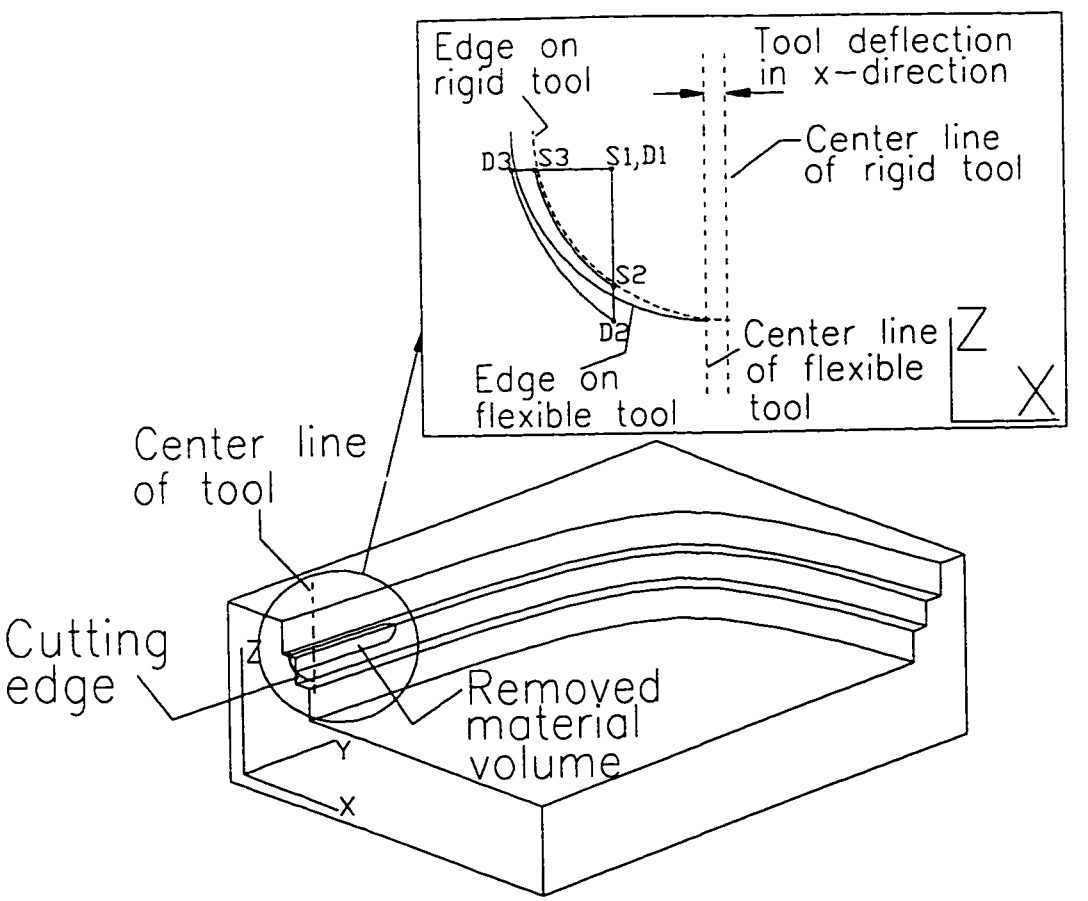


Figure 4.35: Feedback of dynamic deflections to in-cut segments

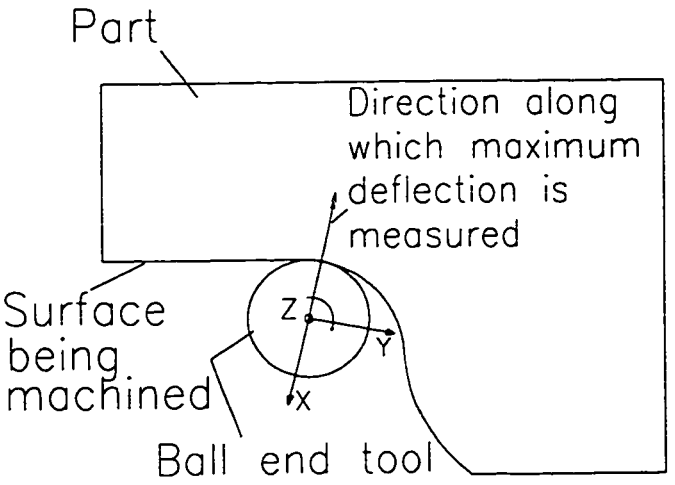


Figure 4.36: Maximum deflection measured in local coordinate system

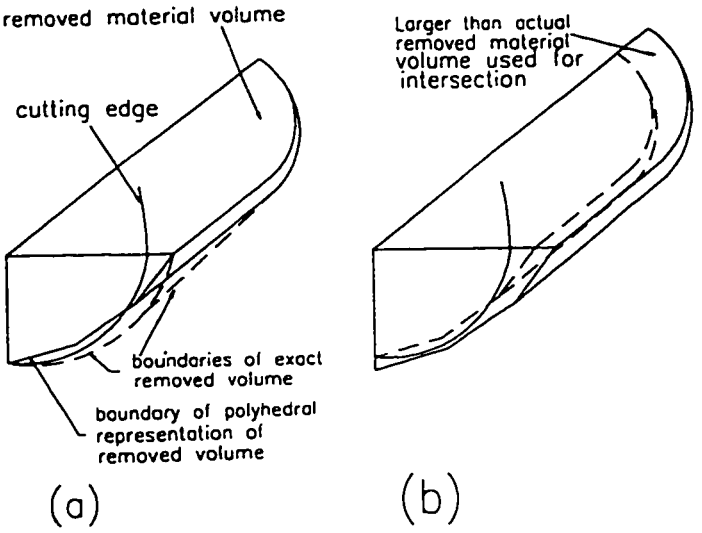


Figure 4.37: Tolerance problem with polyhedral models

Chapter 5

Process based CAD/CAM system

5.1 Overview of system

5.1.1 Description

As mentioned previously, though CAD/CAM systems have become an essential tool in the machining industry, their role in achieving the industry needs is still limited. An enhanced CAD/CAM system for machining process simulation (also called: Process based CAD/CAM system) is intended to overcome those limitations. The system proposed here provides an accurate, comprehensive and industrially applicable solution for machining process simulation. It fills the gap between existing CAD/CAM systems and the machining process simulation.

This system, which is shown in Figure 5.1, consists of the conventional CAD/CAM system enhanced with a geometric modeller based machining process simulator.

The machining process simulator, which consists of a geometric module and a physical module, is designed to simulate general machining process based on the

generic geometric approach presented in chapter 3 and verified in chapter 4. First, it performs a step by step geometric verification. Next, it extracts the critical geometric information. Third, it computes the critical process parameters (e.g. cutting forces and torque). Finally, it predicts other process parameters and reschedule the cutting conditions to optimize the process.

5.1.2 Characteristics

The use of such a system for different machining scenarios is practical and requires few experimental data to be generated. For example, predicting the cutting forces for 3-axis end milling using a new tool shape requires the following:

1. A representation of the cutting edge on the tool
2. An upgrading of the library of tool swept volumes with those corresponding to that tool
3. A representation of the distribution of the rake angle or the normal to the rake face
4. A single calibration experiment for each set of cutting conditions

The calibration is then valid for the workpiece/tool combination used in the test and under those specific cutting conditions. Theoretically speaking, one experiment is needed for each set of cutting conditions. However, calibration can be done at selected conditions and then interpolation is used to produce data for other cutting conditions.

5.1.3 Use of the System

More generally, the Process based CAD/CAM system can be used to assist in:

1. Verifying NC codes, geometrically and physically
2. Optimizing the cutting conditions subject to machining constraints
3. Improving the accuracy of the final product and reducing the required finishing time/work
4. Modifying existing tool designs and designing more efficient tools or cutting edges.

5.1.4 Scope of application of the current system

The current system is limited to 3-axis end milling, which is still the most common practice in today's industries. The geometric simulator developed here can handle 3-axis milling using ball and flat end mills only. The physical simulator is limited to computing the static instantaneous cutting forces for those milling operations.

Due to the fact that the cutting process dynamics is not incorporated and because of the nature of the geometric simulation, the current system is most suitable for modeling roughing and semi-finishing operations. Requirements for extending the current system to other tool shapes, 4- and 5-axis machining, dynamic force prediction are discussed in the next chapter.

5.2 ACIS-based geometric modeler for process simulation of 3-axis end milling

5.2.1 Geometric simulator

Geometric tool: ACIS Solid modeler

Overview of ACIS ACIS [27], a commercial geometric modeler from Spatial Technology Inc., is selected to be the kernel around which the geometric simulator is built. This geometric modeler is an object-oriented geometric modeling toolkit designed for use as a geometric engine within 3D modeling applications. Its main characteristics can be summarized as follows. First, it is a state-of-the-art B-rep modeler that supports curve (wireframe), surface, and solid modelling and thus can be used to represent the different entities involved in the machining operation. The data structure of the solid model representation is based on a topology of C++ classes that are derived from the base ENTITY class (see Figure 5.2). The inheritance property of C++ allows a class to automatically inheret all the properties of the classes at a higher level in the topological hierarchy. Second, it supports the most general surface representation (e.g. in the form of B-splines or NURBS) and consequently can be used to accurately represent any part shape. Third, it is based on CAD/CAM standard technology and in this way facilitates the integration of the simulator with commercial CAD/CAM systems. Next, it provides an open architecture which allows both functionality and structure to be extended. The lowest level of the architecture is written in C++ which is suitable for large, extendable programs. Finally, it provides a Scheme interface that offers a rapid means for application development.

ACIS interfaces Developers can interface with the solid modeler either through the C++ programming language, the Scheme programming language or both.

Applications interface to ACIS through three main ways. First, the Application Procedural Interface (API) is a set of functions that forms a stable, functional interface to ACIS through which developers can *manipulate* the content of a geometric model. A user can't create new API functions but utilizes them to access and manipulate the data structure of a geometric model. Next, the Class Interface (CI) is a set of C++ classes that *define* the form of ACIS geometric and topological models, and other ACIS characteristics. Those classes are used to *access* the information in the data structure of a model. Developers can use these classes and their methods directly, or can derive new classes and methods for special purposes. Finally, the ACIS attribute mechanism provides developers with a way to *attach* application-specific data to any ACIS geometrical or topological entity (e.g. face, edge, vertex etc.). The interfacing is done using C++ and for the first two cases (i.e. API and CI) using Scheme too. Indeed, the Scheme interface is a collection of functions that allows a Scheme-based application to use the API and CI functionality and data of the 3D toolkit.

Extending ACIS ACIS provides an open architecture framework for wireframe, surface, and solid modelling from a common, unified data structure. It is possible to extend both the functionality as well as the structure of ACIS. The structure can be extended by deriving new (child) classes from existing ACIS classes or attaching user-defined attributes to an entity in the data structure. On the other hand, ACIS functionality can be extended through API functions and Husks. First, a developer

can extend ACIS by adding a new functionality, either by creating custom C++ functions that implement existing or new classes and API functions, or by extending the Scheme language. Second, husks are speciality toolkits that extend ACIS and are used by end-user applications. They extend ACIS by adding a unique set of specialized functionality such as rendering or hidden line calculation.

Applications can be written using the ACIS Geometric Modeller or the ACIS 3D Toolkit. The latter which was used in the current work, is a collection of API functions, C++ classes and methods and scheme commands. It is comprised of:

- Development languages: C++, Scheme, and both C++ and Scheme. The latter is an interpreted language like LISP. ACIS includes extensions to the native Scheme language to drive the functionality of the modeler. Scheme being an interpreted language is faster than C++ which requires compilation. In addition, it is easier to handle groups of data (e.g. lists) using Scheme than C++. On the other hand, Scheme is slower and thus less efficient.

- Husks: Basic rendering, Geometric, Graphic interaction, Part Management and Scheme interpreter husks

The current work has extended ACIS functionality by creating C++ functions using the supplied API functions. The development was done at two levels. At the lower one, new scheme commands are written using C++ language. At the higher level, scheme language is used to write the geometric procedure that interfaces with the geometric model and simulate the machining operation.

Geometric representations and procedures

Solid modelling representations and procedures ACIS provides some of the functions that are needed to implement the basic geometric procedures of a solid modeling based machining process simulation. This includes sweeping a wire (along 2D and 3D linear or circular paths), Boolean intersection and difference, curve/solid intersection, and point/solid classification. New extensions in Scheme were written using low level API functions to perform the procedures required to simulate the milling process. This interface forms a safe layer, for manipulating geometric models, that ensures the topological validity of the data structure at all time. The solid modelling-based procedures implemented here generate the part shape as well as the swept volume for different tool shapes (flat and ball end mills) and motions (2D and 3D linear and circular interpolations).

Surface modelling representations and procedures Surface modeling is used to generate a surface representation of the rake face of a cutting edge, from which a distribution of the rake angle can be extracted. The procedure was actually implemented for the case of ball end milling. Measurements were done on a two flute ball end mill with one inch diameter. The measuring procedure is described in Appendix A. The measured points were used to fit a spline surface to the rake face. The surface representation was then used to produce the distribution of the normal along (or close to) the cutting edge. Depending on the formulation of the force model, either the normals are used directly or the distribution of the radial rake angle has to be computed. Figure 5.3(a) shows the fitted surface. The extracted rake angle distribution is depicted in Figure 5.3(b). The measured data points are shown in Appendix B.

Curve modelling representations and procedures Curve modeling is used to generate a curve representation of the cutting edge, which is needed in the extraction of the in-cut segments (as shown in the sequel). The analytical equations found in the open literature for the cutting edges on end mills are not accurate representations of all the tools available in the market as different manufacturers use different edge designs. This is particularly true in the case of ball end mills. The geometry of the edge on the ball nose of these tools involves variable local diameters, helix angles and rake angles.

Figure 5.4 shows some of the analytical equations found in the literature for the cutting edges on ball and flat end mills, where β is the helix angle, R is the (nominal) tool radius, α_n is the normal rake angle of the plane rake face, θ_1 is the angular position of the point on the cutting edge measured from the positive x-axis, θ_2 is the angle of the point in the spherical coordinate system, and θ_3 is the angular position designated to tooth 1 (arbitrarily selected) at the free end ($z = 0$), and $\theta_i(\theta, z)$ is the angular position of a differential element on the ith cutting edge of an $n-$fluted cutter at a distance z from the free end (bottom of the tool).

More reliable representations can be obtained by measuring points directly on the cutting edge(s) of the tool being used for a particular application. The measuring procedure is presented in Appendix A. Figure 5.5 shows a comparison between the cutting edge given by Yang and Park [65] and the B-spline curves fitted to edges measured on a tool from Putnam and another from Osborn (The points measured on these edges are shown in Appendix B). All three cutters are 2-flute, one inch ball end mills.

In the current work, a ball end mill with a constant lead (a tool widely used in

the machining industry) is considered. The cutting edge on the ball nose of this tool can be described by the following analytical equation:

$$\frac{D \cdot \pi}{tan(\phi)} = L \tag{5.1}$$

where L is the lead, D is the local diameter of the tool, and ϕ is the corresponding local helix angle. The equations used to compute points on the edge using this analytical form are shown in Appendix D. The accuracy of such a representation was checked by comparing the latter with an edge fitted to a set of points measured on the actual tool. Ten points on the edge together with initial conditions (i.e. slopes of the edge at start and end points) were used to interpolate a B-spline curve to the actual edge. If the starting angle specified by the manufacture (here 30°) is used, then a significant dsicrepancy is found between the analytical and the experimentally derived representations. By trial and error, a better estimate of the starting helix angle was found (22.5°). Using that value, a very good agreement was established. Another reason for using a synthetic representation of the edge is the fact that the analytical representation of the cutting edge is usually not supported by solid modelers (Imani et al [26]).

Extraction of critical geometric information The basic procedure to extract this information consists of intersecting the cutting edge with every and each face of the updated part. The resulting intersections are then filtered and classified. First, the intersections which do not lie in the front 180° of the rotating tool are eliminated. Next, the remaining intersections are classified into one of two types of segments: 1) Out-of-cut and 2) In-cut. First, the segments are formed by starting at the tooth bottom ($t = 0$) and forming a segment out of each two consecutive intersections. Then,

for each segment $[td_k, tu_k]$, the mid-point (i.e point on the curve which corresponds to $t = (td_k + tu_k)/2$) is classified against the part using Solid/point classification, a API function provided in ACIS. If that point lies inside the solid (i.e. the part), then the segment is classified as "in-cut"; otherwise, it is classified as "out-of-cut". These in-cut segments together with the tool angular position are required to compute the undeformed chip thickness.

Computing the undeformed chip thickness For the case of 3-axis ball end milling, Feng and Menq [21] derived the following expression for the undeformed chip thickness:

$$t_c(\theta, z) \approx R_2(z) - R_1(z) + f_h \cdot \sin(\theta) \tag{5.2}$$

where f_h is the horizontal component of the feed/tooth, R_1 and R_2 are the local radii of the tool at two positions distant by the feed/tooth (along the tool path) measured at the same axial location (i.e $z-$ direction). As observed by Imani et al [26], a more accurate calculation of the undeformed chip thickness can be obtained by using the following modified equation:

$$t_c(\theta, z) = R_2(z) + f_h \cdot \sin(\theta) - [R_1^2(z) - f_h^2 \cdot \cos^2(\theta)]^{\frac{1}{2}} \tag{5.3}$$

5.2.2 Physical simulator

A general physical simulator consists of force models for different machining operations and tools and a feed rescheduling routine for optimizing the cutting conditions. In this work, only force models for flat and ball end milling are developed. For the particular case of ball end milling, an enhanced mechanistic force model that accurately predicts the process parameters in practical ranges of cutting conditions and

for any immersion geometry is implemented.

Enhanced mechanistic force model

A mechanistic force model for ball end milling that does not need to be calibrated for different immersion geometries was proposed by Yücesan and Altintaş [66]. In their work, however, constant/ average values were assumed for the cutting parameters and the chip flow angle. In addition, the testing and verification of the model was limited to simple 2D cases such as slotting and half-immersion. In contrast, the current enhanced model includes the variations of the cutting constants along the cutting edge, the rake angle, and the chip flow angle. In addition, it is tested and verified for the general 3-axis ball end milling. The formulation of this enhanced model is as follows.

The cutting forces are considered as the summation of normal and frictional forces acting on the rake face as well as the cutting edge (Budak [10]). The unit vectors representing the directions of the elemental forces are illustrated in Figure 5.6(a). On the rake face, it is assumed that the friction force is along the chip flow angle (Collinearity assumption), Figure 5.6(b). The chip flow angle (η_c) depends on the tool geometry, cutting conditions and material. A linear variation of the chip flow is assumed in the z-direction; this is expressed as follows:

$$\eta_c = c_1 + c_2 \cdot z \tag{5.4}$$

The elemental forces acting on the rake face (dF_{nr} and dF_{fr}) are proportional to the elemental chip contact area (dA_r). It is assumed that the chip contact length is equal to twice the uncut chip thickness (t), Figure 5.6(c). Thus, the elemental chip contact

area is given by:

$$dA_r(\theta, z_i) = 2 \cdot t(\theta, z_i) \cdot \frac{dz}{\cos(i(z_i))} \tag{5.5}$$

where i , the helix angle, is expressed as follows:

$$i(z_i) = \arctan(\frac{R(z_i) \cdot \tan(\phi_0)}{R_0}) \tag{5.6}$$

where R_0, $R(z_i)$ and ϕ_0 are tool radius, slice radius and helix angle at the start point, respectively. Therefore, the elemental normal and frictional forces on the rake face can be estimated by:

$$\begin{aligned} \|d\vec{F}_{nr}(\theta, z_i)\| &= K_{nr}(z_i) \cdot dA_r(\theta, z_i) \\ \|d\vec{F}_{fr}(\theta, z_i)\| &= K_{fr}(z_i) \cdot dA_r(\theta, z_i) \end{aligned} \tag{5.7}$$

where K_{nr} and K_{fr} are cutting coefficients which can predict the local cutting mechanics in the z-direction:

$$\begin{aligned} K_{nr}(z_i) &= (c_3 + c_4 \cdot z_i + c_5 \cdot z_i^2) \\ K_{fr}(z_i) &= (c_6 + c_7 \cdot z_i + c_8 \cdot z_i^2) \cdot K_{nr}(z_i) \end{aligned} \tag{5.8}$$

The elemental normal and frictional edge forces are proportional to the elemental edge length (ds):

$$\begin{aligned} \|d\vec{F}_{ne}(\theta, z_i)\| &= K_{ne}(z_i) \cdot ds(z_i) \\ \|d\vec{F}_{fe}(\theta, z_i)\| &= K_{fe}(z_i) \cdot ds(z_i) \end{aligned} \tag{5.9}$$

where

$$\begin{aligned} K_{ne}(z_i) &= (c_9 + c_{10} \cdot z_i + c_{11} \cdot z_i^2) \\ K_{fe}(z_i) &= (c_{12} + c_{13} \cdot z_i + c_{14} \cdot z_i^2) \cdot K_{ne}(z_i) \end{aligned} \tag{5.10}$$

Finally, the instantaneous cutting force acting on the cutting edge can be expressed as follows:

$$\begin{aligned}\vec{F}(\theta) &= \sum_{z_{in}}^{z_{out}}([K_{nr}(z_i)\cdot\hat{n}_r(\theta,z_i)+K_{fr}(z_i)\cdot\hat{n}_{fr}(\eta_c,\theta,z_i)]\cdot dA_r(\theta,z_i) \\ &+ [K_{ne}(z_i)\cdot\hat{n}_e(\theta,z_i)+K_{fe}(z_i)\cdot\hat{n}_{fe}(\theta)]\cdot ds(z_i))\end{aligned} \quad (5.11)$$

The integration limits (z_{in} and z_{out}) are extracted using the geometric simulator.

5.3 Verification of the system for the case of 3-axis ball end milling

5.3.1 Experimental measurements

A Kistler type 9255A, 3-force component table dynamometer was used to measure the three F_x, F_y, and F_z components of the instantaneous cutting force acting on the tool. A spindle shaft encoder was used as an external clock to provide synchronization with the cutting edge position. Cutting tests were carried out on a CNC machining center (YAM) which did not exhibit significant vibrations during the performed tests. The dynamometer is relatively rigid with a natural frequency of 2000 Hz. Both 2- and 3- axis ball end milling with different speeds, feeds and immersion geometries were performed on workpieces made of 1018 steel.

5.3.2 Calibration of force model

The calibration method to determine parameters c_1 to c_{14} was implemented for the half-immersion case. Sequential quadratic programming (SQP), a constrained optimization algorithm, was used to minimize the sum of squared errors between the predicted and measured force. The constrained optimization successfully converged to the solution as shown in Figure 5.7.

5.3.3 Measured and predicted forces

Figure 5.8 presents a case of 3D machining using a ball end mill. The cutting geometry is shown in part (a) of the figure. Part (b) shows the corresponding cutter-part immersion. Figure 5.9 shows a comparison of the instantaneous cutting forces predicted using the current system and the corresponding experimental measurements. The slight difference can be attributed to: 1) Errors in the measurement of the cutting edge and rake face; 2) Interpolation tolerances in the geometric fitting of the cutting edge and rake face; 3) Radial run-out on the cutting edge; and 4) Other errors related to the modeling of the cutting process mechanics.

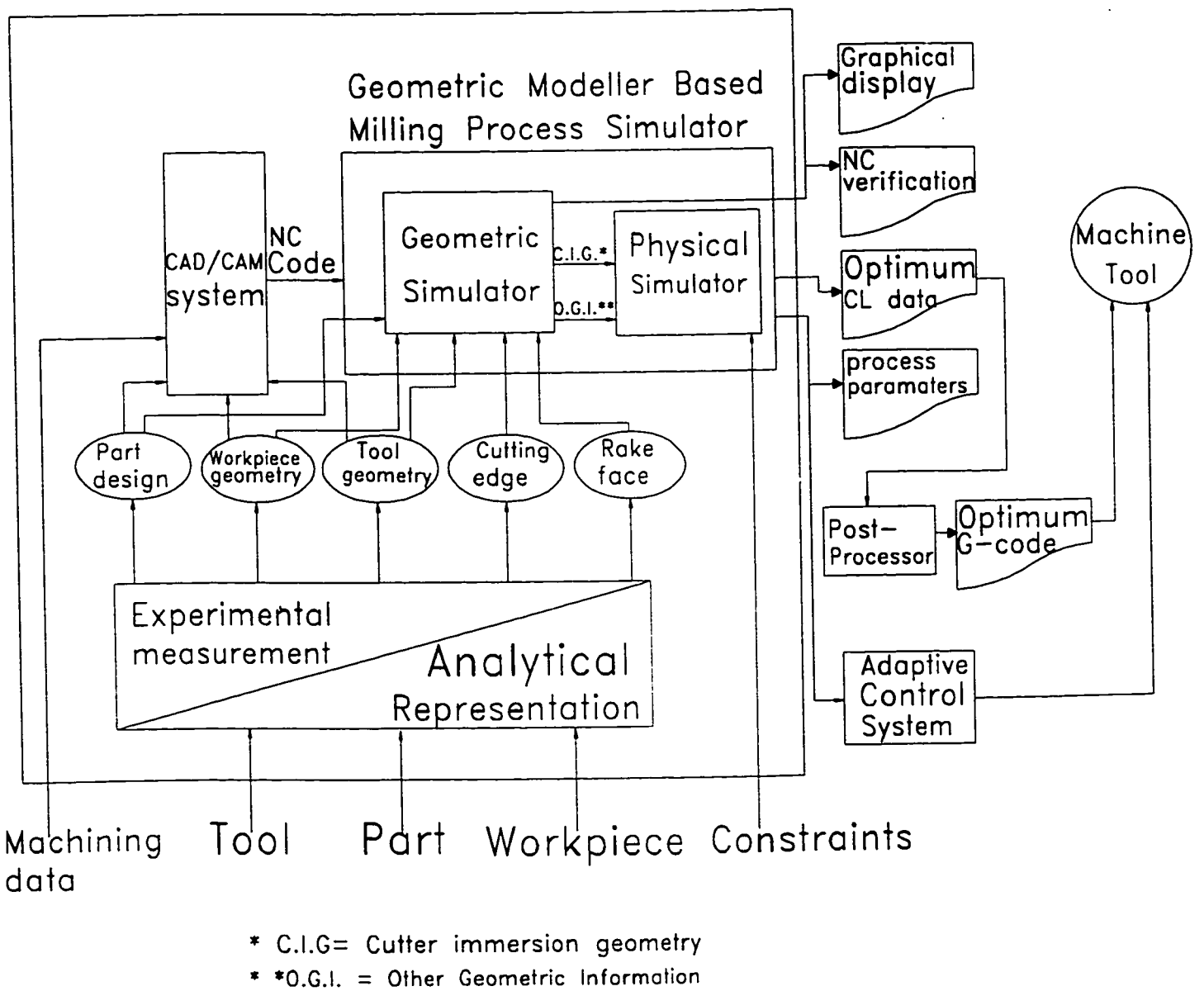
5.3.4 Observations

Lower cutting forces and powers are experienced if cutting with the vicinity of the tool tip is avoided. Also, as the up-hill angle increases, the ball nose tip engagement decreases which in turn significantly affects the magnitude of the resultant forces.

Figure 5.10 shows the extracted immersion geometry for different levels of tool engagement. They vary in terms of the extent to which the tool tip is close to the

engaged part of the edge. These different cutting scenarios were achieved by using parts with different inclinations of the top surface. The tool was then moved to cut along a path (3D linear interpolation) that is parallel to the top surface, Figure 5.11. The corresponding measured and predicted forces for four different "up-hill" angles (10^o, 15^o, 20^o, and 25^o) are shown in Figure 5.12. The agreement is very good. The higher values of the predicted forces can be attributed to the absence of rubbing force modeling in the physical simulation. Indeed, the calibration constants were extracted from the half-immersion case where the tool tip is engaged and thus involving rubbing effect. On the other hand, in the cases shown in Figure 5.12, the cutting is away from the tool tip.

In 3-axis milling, the ball nose tip engagement is directly dependent on the surface normal because the tool can not be tilted. On the other hand, if 5-axis milling is used, then the tool can be oriented to control this engagement and consequently the force magnitude. This is an important advantage of 5-axis milling over 3-axis milling using ball end mills, an operation widely used nowadays in die/mold machining.



* C.I.G= Cutter immersion geometry

* *O.G.I. = Other Geometric Information

Figure 5.1: Enhanced CAD/CAM system for machining process simulation

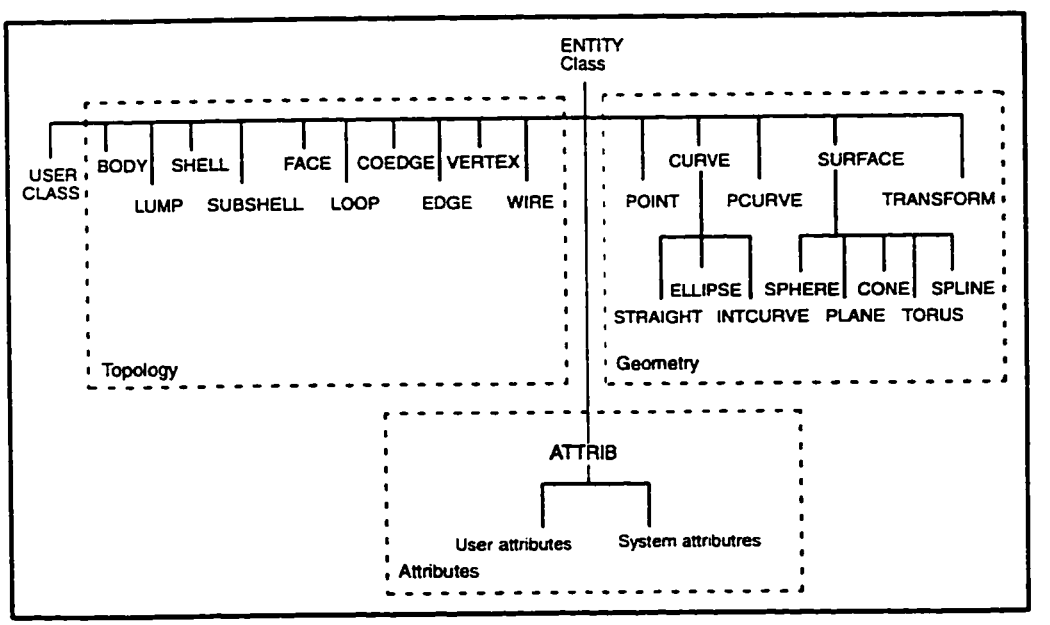
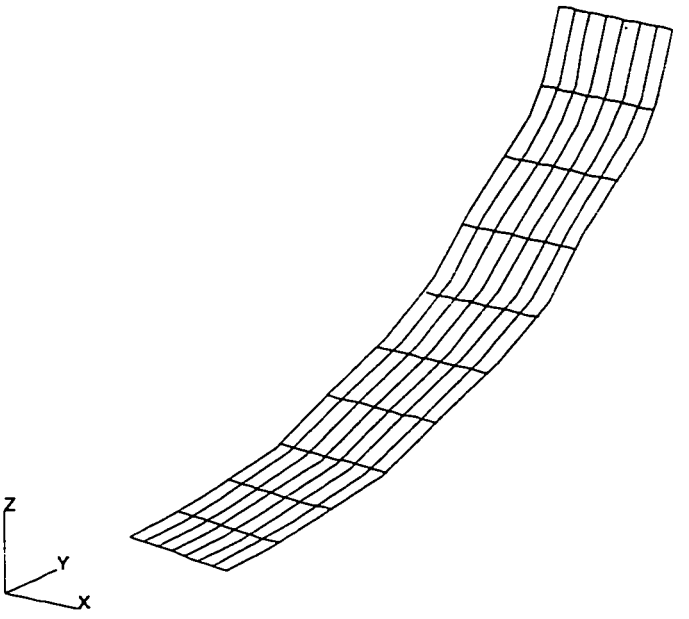


Figure 5.2: Solid model topology, Geometry and attached Attributes used in ACIS



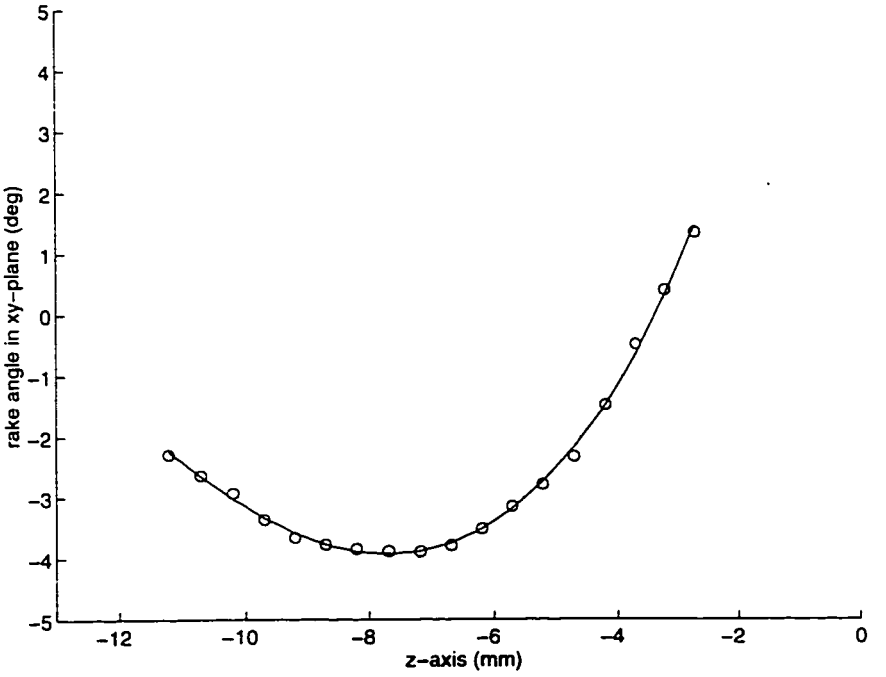


Figure 5.3: (a) B-spline surface representing the rake face; (b) Extracted rake angle distribution

Cutting edge on:	Equation
Flat end mill or cylindrical part of ball end mill	$x(\theta_1) = R\cdot\cos(\theta_1)$ $y(\theta_1) = R\cdot\sin(\theta_1)$ $z(\theta_1) = R\cdot\theta_1/\tan(\beta)$
Ball nose of ball end mill [17]	$x(\theta_2)=R\cdot\cos(\alpha_n)\cdot\sin(\theta_2)$ $y(\theta_2)=-1/2\cdot R\cdot\sin(2\cdot\alpha_n)\cdot$ $(\cos(\theta_2)-1)$ $z(\theta_2)=-R\cdot(\cos^2(\alpha_n)\cdot\cos(\theta_2)+$ $\sin^2(\alpha_n))$
Ball nose of ball end mill [18]	$\theta_i(\theta_3,z) = \theta_3-(z/R)\cdot\tan(\beta)$ $-(i-1)\cdot(2\pi/n)$

Figure 5.4: Analytical cutting edge representations

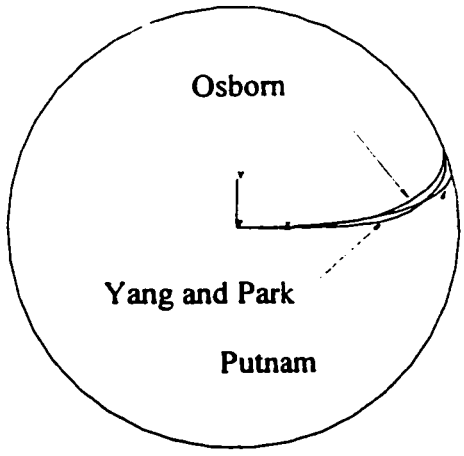


Figure 5.5: Comparison of different cutting edge designs

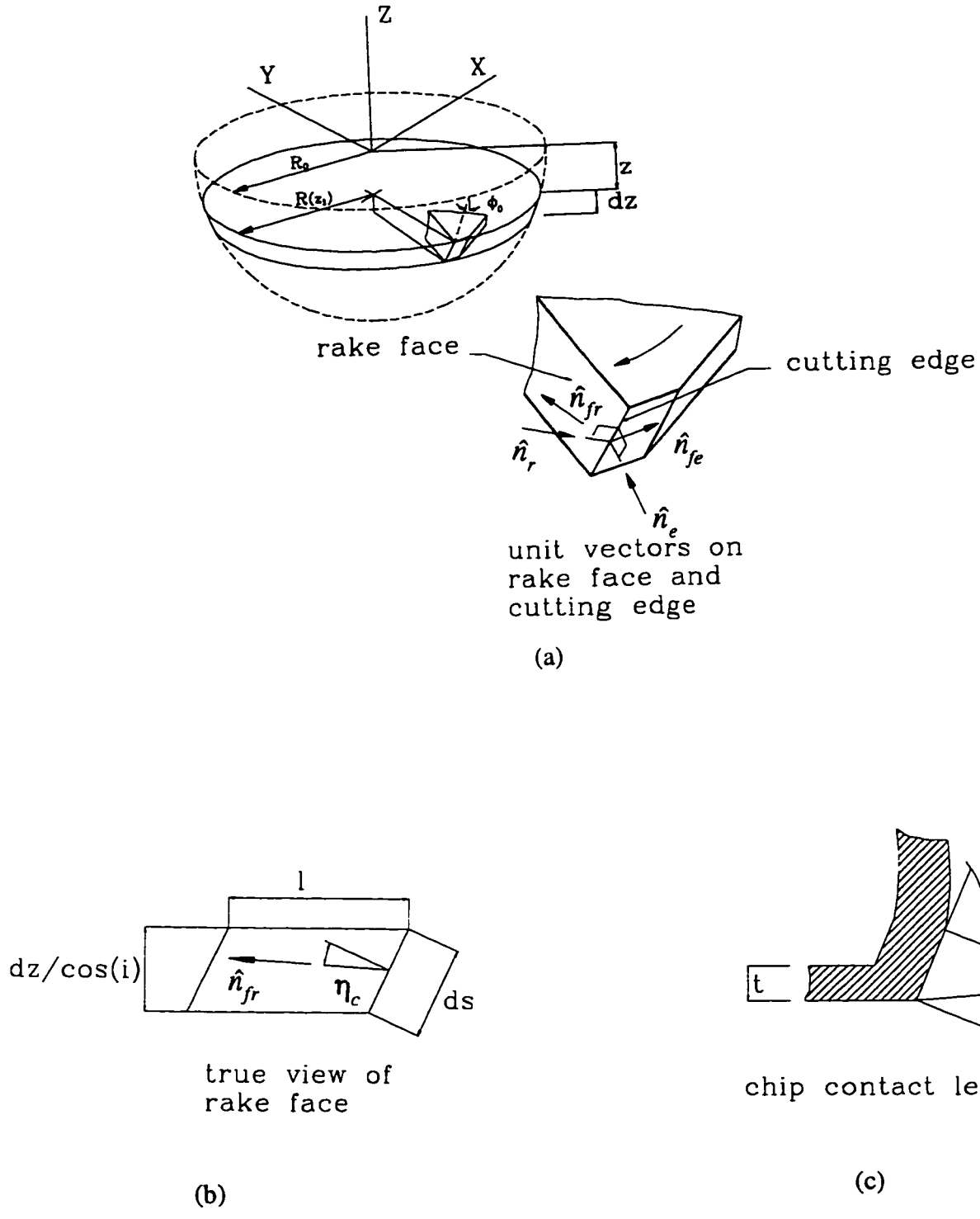


Figure 5.6: (a) Unit vectors on rake face and cutting edge; (b) True view of rake face showing the chip flow angle; (c) Uncut chip thickness and chip contact length

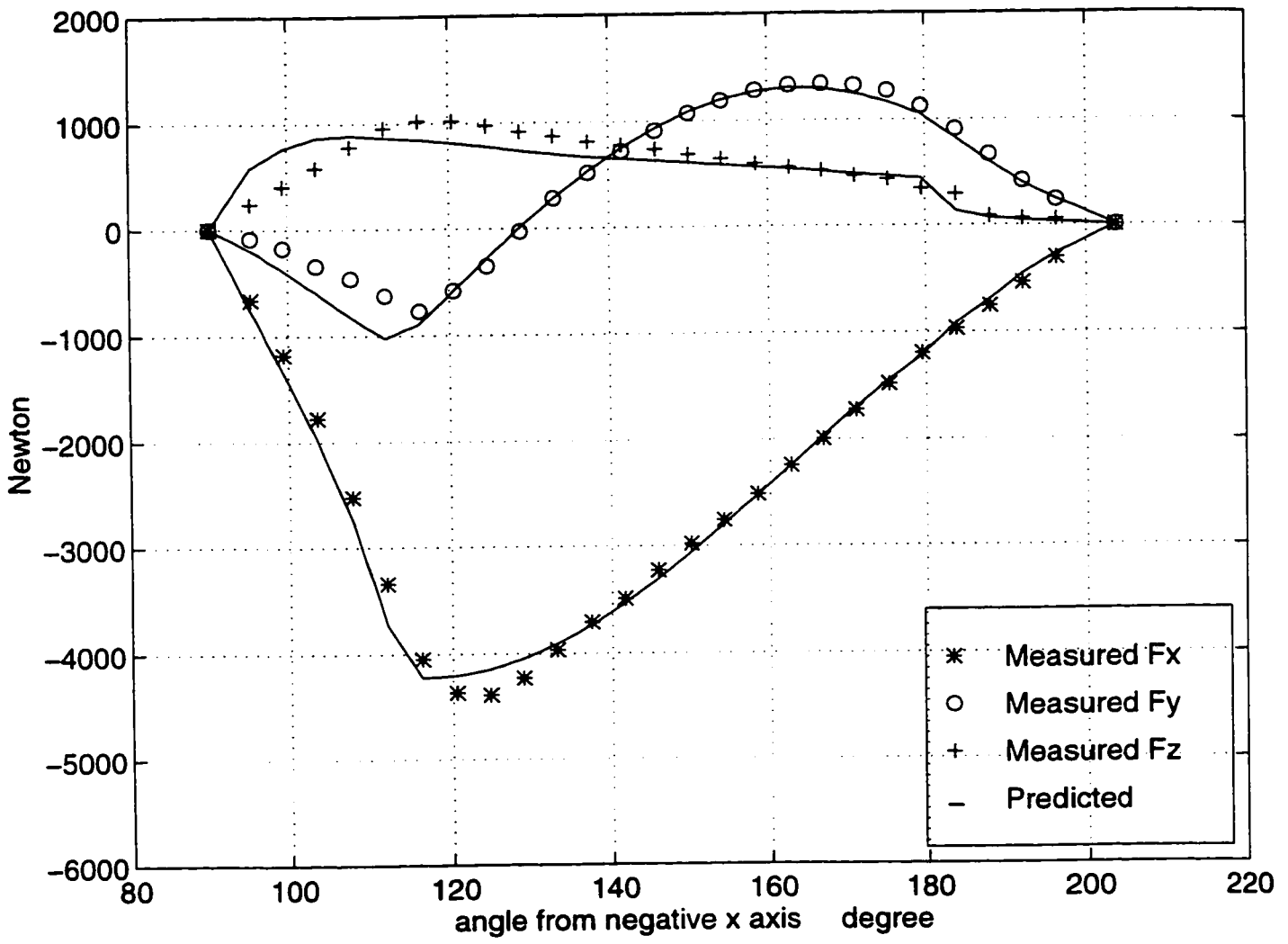
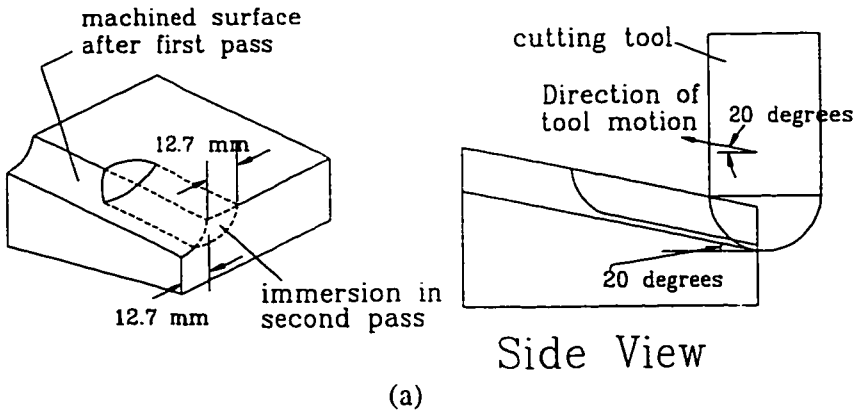
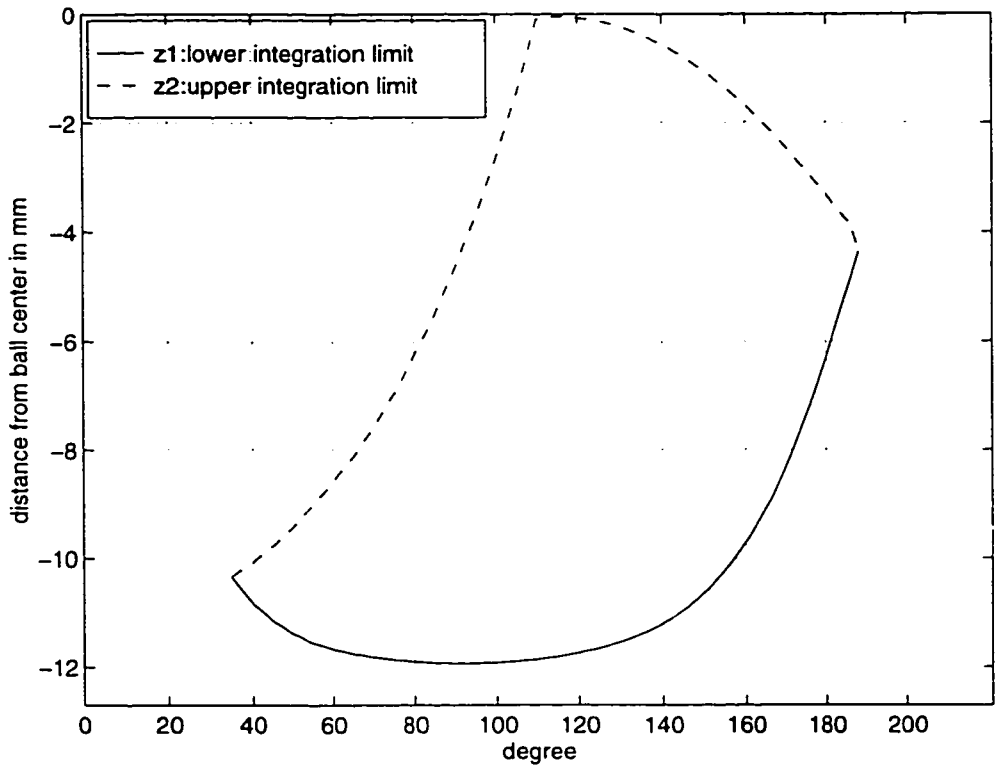


Figure 5.7: Force model calibration using the case of half-immersion



(a)



(b)

Figure 5.8: Geometric simulation of a 3D machining case using ball end mill

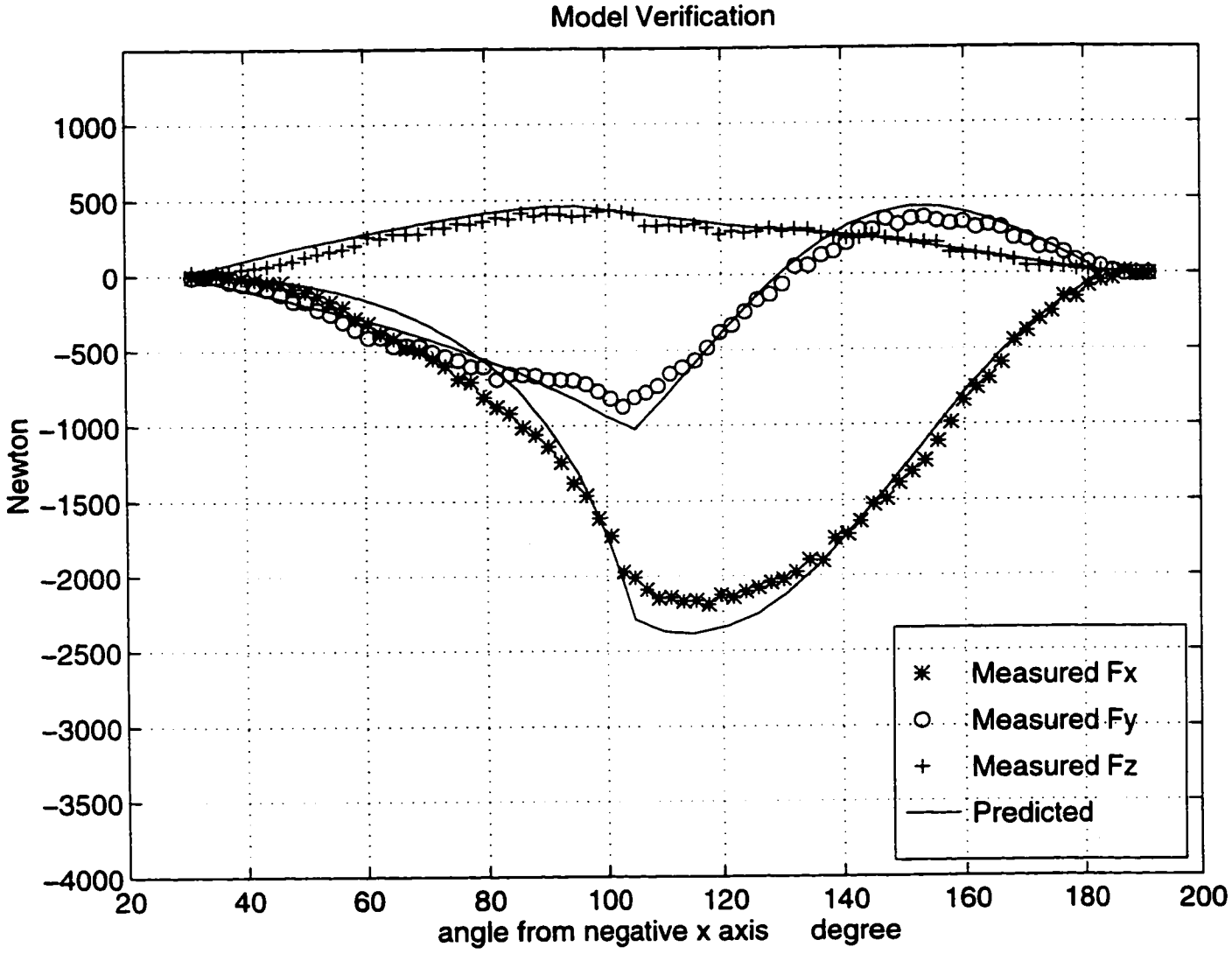


Figure 5.9: Physical simulation of a 3D machining case using ball end mill: Comparison of experimental and predicted cutting forces

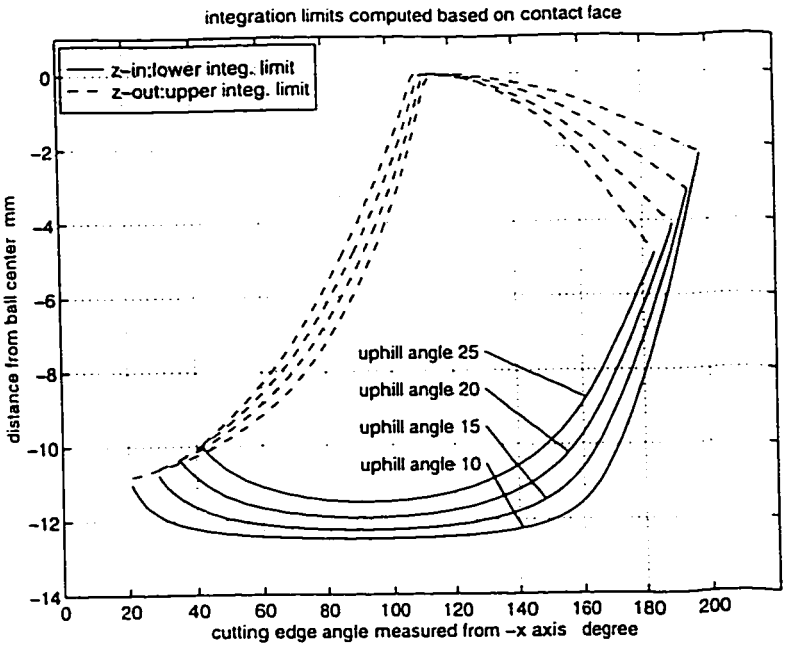


Figure 5.10: Effect of the up-hill angle on the immersion geometry

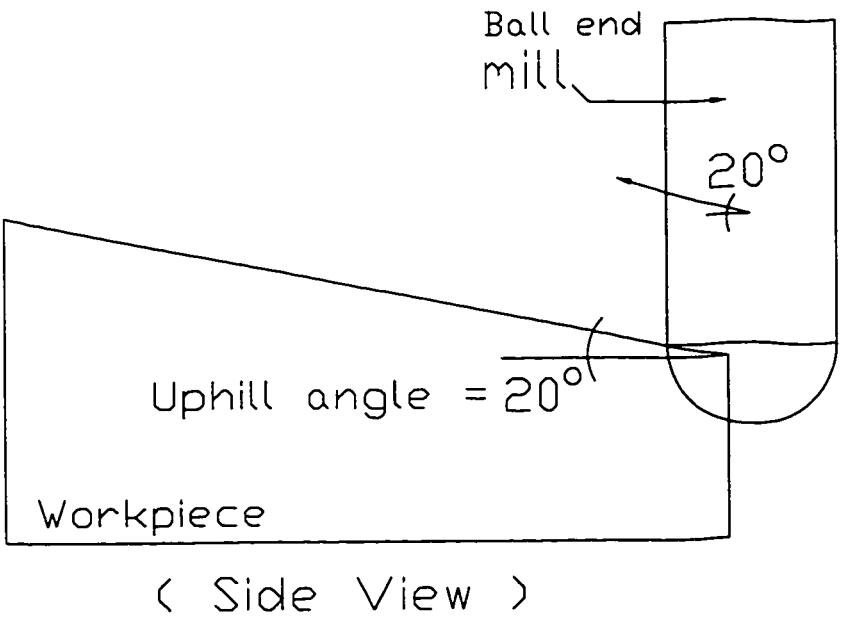


Figure 5.11: Definition of Uphill angle and direction of cutting

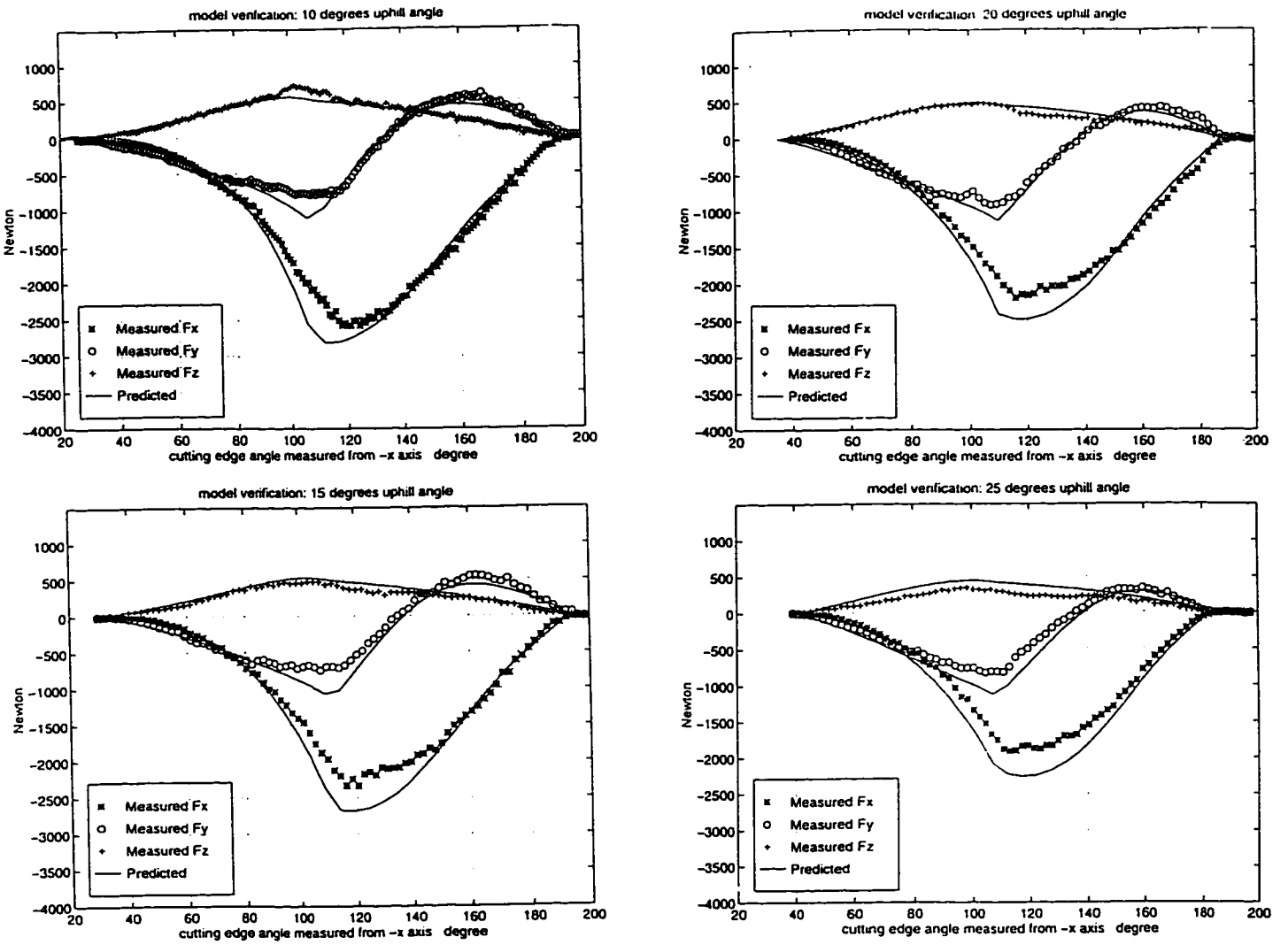


Figure 5.12: Comparison of simulated and predicted forces for different 3D immersion cases

Chapter 6

Summary and Future Research

6.1 Summary

Though solid modeling has established itself as the geometric representation in many of today's commercial CAD/CAM systems, it still does not support directly some of the manufacturing applications, including machining process simulation. Also, CAD/CAM systems capabilities are still limited to tool path planning which nominally achieves a specified part shape. They do not model the physical parameters of the machining process and thus fail to guarantee a safe/reliable operation during actual machining or to improve the productivity by optimizing the cutting conditions subject to physical constraints. Most of the previous research has focussed on the geometric aspect of the machining process (e.g. NC geometric simulation and verification). Those research work which included the physical aspect of the process in the machining simulation, proposed strategies for extracting the geometric information

that are not based purely on solid modeling. Such techniques could not carry informationally complete representations for some or all of the geometric entities involved in the machining process. This has limited the accuracy of the resulting simulation as well as its capabilities (e.g. The simulator can only compute average forces). It has also produced geometric techniques which are process specific (e.g. only suitable for the case of milling, not for turning or drilling). This thesis has dealt with those limitations. More specifically, the major contributions can be summarized as follows:

- A generic and accurate geometric modeling approach for simulating the general machining operation (i.e. any machining process, general part and tool shapes, and any cutting edge geometry) was proposed and implemented. As opposed to previous work, the current one was based on a careful consideration of the different geometric entities involved in the machining process (part shape, removed material volume, cutting edge, and rake face) and aimed at developing a methodical approach which gives priority to generality and accuracy. This has led to a general procedure that serves as a paradigm for all machining processes.

- The cutter-part immersion geometry, in the form of in-cut segments of the engaged edges, was identified as being the most basic geometric information required to accurately predict the different process parameters. Other geometric information that might need to be extracted using the geometric simulator consist of chip thickness variation and rake angle distribution. Expressions for chip thickness variation for flat and ball end mills can be used to compute the chip load with a very good accuracy.

- It was found that more accurate representations for cutting edges of existing end mills can be extracted by measuring points and fitting them using B-spline curves. It was also shown that the rake angle distribution on the rake face can be extracted from the representation of the rake surface. The latter can be generated by fitting a B-spline surface to a grid of points measured on the rake face. These procedures were implemented herein for the case of ball end mills.

- The developed system can model and simulate full 3-axis milling of 3D parts. The enhanced force model (which includes the variation of the cutting constants along the cutting edge, the rake angle, and the chip flow angle) predicts very accurate force components. Although the model needs to be calibrated for each tool/workpiece combination and cutting conditions, the calibration is valid for any immersion geometry.

- Experiments were conducted to verify the accuracy of the simulator and show its applicability. First, instantaneous cutting forces and torque, for a typical 2 1/2-axis die semi-finishing using ball end mills, were measured and compared with simulation results. Next, 3-axis ball end milling of 3D part was simulated and instantaneous cutting forces computed. Experiments were run to verify the simulation. In both cases the agreement between experimental measurements and simulation results was very good.

- It was possible to compute dynamic forces and predict the onset of chatter by including the process dynamics in the physical simulation. This required the extraction of the modal parameters of the machine tool used in the experiments and the use of dynamic regenerative force model.

- A strategy for developing an enhanced CAD/CAM system for machining process simulation has been proposed and the basic requirements identified and implemented. The development which combined solid, surface, and curve modeling was based on CAD/CAM industry standards and thus facilitates the rapid commercial application of the technology.

6.2 Future Research

Possible continued research would extend the same geometric approach to 4- and 5-axis end milling. This requires the generation of swept volumes of tools moving in 4- and 5- axis motions. The difference between the latter and 3-axis motion when modeling the swept volume lies in the shape of the driving edge (or critical curve) along the tool path. In 3-axis machining, the shape is invariant and thus the generation of the swept volume requires only the driving edge at the initial tool position and the instantaneous tool orientations along the path. On the other hand, in 5-axis machining, both the instantaneous tool orientation and driving edge are required at each increment along the motion. One way of producing the swept volume in the latter case is by direct application of the envelope equations to calculate discriminate curves at incremental positions along the tool path, then create faces in between. Performing the geometric simulation with other tool shapes requires the addition of the corresponding swept volumes, basically, an upgrading of the library of tool swept volumes. The first step consists of determining the geometric equivalent of the cutter in question. For example, the geometric equivalent of a flat end mill is a cylinder and that of a ball end mill is the union of a cylinder and a hemisphere). Next, the

same procedure described in Chapter 4 is used in case of 3-axis machining or the one suggested above for higher number of axes.

Another possible extension consists of incorporating the process dynamics in the enhanced mechanistic force model to compute the dynamic forces and predicts dynamic deflection as well as the onset of chatter. The approach would be similar to what has been presented and implemented in Chapter 4 to extend the static force model of ball end milling to a dynamic regenerative one. However, the dynamic modeling has to be extended beyond the tool and tool holder in the case of 4- and 5-axis machining to account for possible vibration in the rotary table. The workpiece dynamics also has to be modeled if flexible parts are machined. Finally, an expression of the dynamic chip thickness in 4-5 axis machining is required.

References

[1] F. Abrari. A regenerative dynamic force model for ball end milling. Master's thesis, McMaster University, Hamilton, ON, April 1994.

[2] F. Abrari, M. A. Elbestawi, and A. D. Spence. On the dynamics of ball end milling: Modeling of cutting forces and stability analysis. to be submitted to International Journal of Machine Tool Design and Research.

[3] Y Altintaş and A. Spence. End milling force algorithms for cad systems. *CIRP Annals*, 40(1):31–34, 1991.

[4] R. O. Anderson. Detecting and eliminating collisions in nc machining. *Computer Aided Design*, 10(4):231–237, 1987.

[5] C. Andrew and S. A. Tobias. Vibration in horizontal milling. *International Journal of Machine Tool Design and Research*, 2:369–378, 1962.

[6] E. J. A. Armarago and N. P. Deshpande. Computerized predictive cutting models or forces in end milling including eccentricity effects. *CIRP Annals*, 38(1):45–49, 1989.

[7] P. Atherton, C. Earl, and C. Fred. A graphical simulation system for dynamic

five-axis nc verification. In *Autofact Show of the society of Manufacturing Eng.*, pages 2–1–2–12, Nov 1987.

[8] A. Baer, C. Eastman, and M. Henrion. Geometric modelling: A survey. *Computer-Aided Design and Manufacture*, 11(5):253–272, September 1979.

[9] B. Baumgart. A polyhedron representation for computer vision. In *National Computer Conference, AFIPS Conf. Proc.*, pages 589–596, 1975.

[10] E. Budak, Y. Altintas, and E. J. A. Armarego. Prediction of milling force coefficients from orthogonal cutting data. *ASME Journal of Engineering for Industry*, 1996. in press.

[11] I. T. Chappel. The use of vectors to simulate material removed by numerically controlled milling. *Computer Aided Design*, 15(3):156–158, May 1983.

[12] R. E. Devor, W. A. Kline, and W. J. Zdeblick. A mechanistic model for the force system in end milling with application to machining airframe structures. In *Eighth North American Manufacturing Research Conference Proceedings*, pages 297–303, May 1980.

[13] R. E. Devor, J. W. Sutherland, and W. A. Kline. Control of surface error in end milling. In *Eleventh North American Manufacturing Research Conference Proceedings*, pages 356–362, May 1983.

[14] M. Dewaele and G. L. Kinzel. Speed improvements for nc program verification using solid modeling. In *Proceedings ASME Int'l Computers in Engineering Conference and Exposition*, volume 1, pages 601–607, Anaheim, C. A., 1989. ASME.

[15] H. El-Mounayri, M. A. Imani, M. A. Elbestawi, and A. D. Spence. Closing the gap between CAD/CAM and machining process simulation: A generic solution. submitted to the 1997 ASME International Mechanical Engineering Congress and Exposition, Symposium on Predictive Modeling in Metal Cutting as Means of Bridging Gap between Theory and Practice, Dallas, Texas.

[16] 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. In *13th Symposium on Engineering Application of Mechanics, 1996 CSME Forum*, pages 394–401, McMaster University, ON, Canada, 1996. CSME.

[17] H. El-Mounayri, A. D. Spence, and M. A. Elbestawi. Milling process simulator-a generic solid modeller based paradigm. 1996. accepted for publication in the Journal of Engineering for Industry.

[18] H. A. Elmaragy. Computer modelling of machining process. In *NAMRC proceedings*, pages 271–276, 1982.

[19] D. J. Ewins. *Modal Testing: Theory and Practice.* Research studies Press Ltd., England, 1986.

[20] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design—A Practical Guide.* Academic Press Inc., 1250 Sixth Avenue, San Diego, CA, USA 92101, third edition, 1993.

[21] H. Feng and C. Menq. The prediction of cutting forces in the ball-end milling process - i. model formulation and model building procedure. *International Journal of Machine Tools and Manufacture*, 34(5):697–710, 1994.

[22] B. K. Fussel and K. Srinivasan. An investigation of the end milling process under varying machining conditions. *Journal of Engineering for Industry*, ...:27–36, February 1989.

[23] B. K. Fussell, C. Ersoy, and R. B. Jerard. Computer generated cnc machining feed rates. In *Proceedings 92 Japan USA Symposium on Flexible Automation*, pages 377–384, New York, NY, USA, 1992. ASME.

[24] Lang G. Understanding vibration measurement, Dec. 1978. Application note 9.

[25] C.M. Hoffman. *Geometric and Solid Modeling–An Introduction.* Morgan Kaufmann, San Mateo, CA, 1989.

[26] B. M. Imani, M. H. Sadeghi, and M. A. Elbestawi. An improved process simulation system for ball end milling of sculptured surfaces. 1997. to be submitted to the International Journal of Machine Tool Design and Research.

[27] Spatial Technology Inc. *ACIS Geometric Modeler: Application Guide.* Colorado, USA, 1996.

[28] F. Ismail, M.A. Elbestawi, R.X. Du, and K. Urbasik. Generation of milled surfaces including tool dynamics and wear. *ASME Journal of Engineering for Industry*, 115:245–252, August 1993.

[29] R. B. Jerrard, R. L. Drysdale, and K. Hauck. Geometric simulation for numerical control machining. In *Proceedings ASME Int'l Computers in Engineering Conference*, volume 2, pages 129–136, San Fransisco, July 31- Aug 3.

[30] M. Karasick. *On the Representation and Manipulation of Rigid Solids.* PhD thesis, McGill University, Montreal, Quebec, Canada, 1988. Comp. Sci.

[31] L. A. Kendall, S. Arora, and G. Cross. Intelligent supervisory control prototype for machining systems. In *Proceedings of the North American Manufacturing Research Conference*, volume 16, pages 309–315, Dearborn, MI, 1988. Society of Manufacturing Engineers.

[32] W. A. Kline and R. E. Devor. The effect of runout on cutting geometry and forces in end milling. *International Journal of Machine Tool Design and Research*, 23:123–140, 1983.

[33] W. A. Kline, R. E. Devor, and J. R. Lindberg. The prediction of cutting forces in end milling with application to cornering cuts. *International Journal of Machine Tool Design and Research*, 22:7–22, 1983.

[34] W.A. Kline, R.E. DeVor, and R. Lindberg. The prediction of cutting forces in end milling with application to cornering cuts. *International Journal of Machine Tool Design and Research*, 22(1):7–22, 1982.

[35] W.A. Kline, R.E. DeVor, and I.A. Shareef. The prediction of surface accuracy in end milling. *ASME Journal of Engineering for Industry*, 104:272–278, 1982.

[36] F. Koenigsberger and A.J.P. Sabberwal. An investigation into the cutting force pulsations during milling operations. *International Journal of Machine Tool Design and Research*, 1:15–33, 1961.

[37] F. Koenisberger and J. Tlusty. *Machine Tool Structures.* Pergamon Press, 1970.

[38] M. E. Lim, H. Feng, and C. Menq. The prediction of dimensional errors for machining sculptured surfaces using ball-end milling. *Manufacturing Science and Engineering*, 64:149–156, 1993.

[39] M. Mäntylä. *An Introduction to Solid Modelling.* Computer Science press, Rockville, Maryland.

[40] M.E. Martellotti. An analysis of the milling process. *Transactions of the ASME*, 63:677–700, 1941.

[41] M.E. Merchant. Mechanics of the cutting process I. *Journal of Applied Physics*, 16:267–275, 1945.

[42] A. P. Narvekar, Y. Huang, and J. H. Oliver. Intersection of rays with parametric envelope surfaces representing five-axis nc milling tool swept volumes. *Advances in Design Automation*, 2:223–230, 1992.

[43] J. H. Oliver and Goodman. E. D. Color graphic verification of nc milling programs for sculptured surfaces. In *10th Annual Automotive Computer Graphics Conference and Exposition*, Detroit, 1985. Engineering Society of Detroit.

[44] S. Park, M. Yang, and C. Lee. Simulation of nc machining using ball end mill. In *Computer Modelling and Simulation of Manufacturing Processes*, volume 20, pages 67–76, New York, NY, USA. ASME, Materials Division (Publication).

[45] L. Piegl. On nurbs: A survey. *IEEE - Computer Graphics and Applications*, 11(1):55–71, 1991.

[46] A. Requicha and H. Voelcker. Boolean operation in solid modelling: Boundary evaluation and merging algorithms. In *Proceedings of the IEEE*, volume 73, pages 30–44, 1985.

[47] A.A.G. Requicha. Representations for rigid solids: Theory, methods, and systems. *ACM Computing Surveys*, 12(4):437–464, December 1980.

[48] P. Rigby. High speed milling in the mold and die making industries. In *Industrial Diamond Association, Diamond and CBN Ultrahard Materials Symposium' 93*, pages 29–30, Windsor, ON, Canada, Sept 1993.

[49] S.D. Roth. Ray casting for modeling solids. *Computer Graphics and Image Processing*, 18:109–144, 1982.

[50] S. Smith and J. Tlusty. An overview of modeling and simulation of the milling process. *ASME Journal of Engineering for Industry*, 113:169–175, 1991.

[51] A. Spence, Y. Altintas, and D. Kirkpatrick. Direct calculation of machining parameters from a solid model. *Computers in Industry*, 14:271–280, 1990.

[52] A.D. Spence. Solid modeller based milling process simulation. Report CICSR-TR92-010, The University of British Columbia, Centre for Integrated Computer Systems Research, Vancouver, BC, March 1992.

[53] A.D. Spence and Y. Altintas. A solid modular based milling process simulation and planning system. *ASME Journal of Engineering for Industry*, 116:61–69, February 1994.

[54] R. Sridhar, R. E. Hohn, and G. W. Long. A general formulation of the milling process equation. contribution to machine tool chatter research-5. *Transactions of ASME*, 90:317–324, 1968.

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

[56] R. Tilove. Set membership classification: A unified approach to geometric intersection problems. *IEEE Transactions on Computer*, C-29(1):847–883, 1980.

[57] J. Tlusty and F. Ismail. Special aspects of chatter in milling. *ASME Journal of Vibration, Acoustics, Stress and Reliability in Design*, 105:24–32, 1983.

[58] J. Tlusty and P. MacNeil. Dynamics of cutting forces in end milling. *CIRP Annals*, 24:21–25, 1975.

[59] J. Tlusty and S. Smith. Forced vibration, chatter, accuracy in high speed milling. In *Proceedings of the 13th North American Manufacturing Research Conference*, volume Berkeley, California, pages 221–229, 1985.

[60] B. C. Ullagaddi. Modeling and monitoring of cutting dynamics in end milling. Master's thesis, McMaster University, Hamilton, ON, 1991.

[61] T. Van Hook. Real-time shaded nc milling display. In *ACM SIGGRAPH*, volume 20, pages 15–20, 1986.

[62] H. B. Voelcker and W. A. Hunt. The role of solid modelling in machining-process modelling and nc verification. In *Int'l Congress and Exposition*, pages 1–8, Detroit, Michigan, Feb. 23-27 1981.

[63] W.P. Wang. Solid modeling for optimizing metal removal of three-dimensional NC end milling. *Journal of Manufacturing Systems*, 7(1):57–65, 1988.

[64] K. Yamazaki, N. Kojima, C. Sakamoto, and T. Saito. Real-time model reference adaptive control of 3-D sculptured surface machining. *CIRP Annals*, 40(1):479–482, 1991.

[65] M. Yang and H. Park. The prediction of cutting force in ball end milling. *International Journal of Machine Tools and Manufacture*, 31(1):45–54, 1991.

[66] G. Yucesan and Y. Altintas. Mechanisms of ball end milling process. *ASME Journal of Manufacturing Science and Engineering*, 64:543–551, 1993.

[67] I. Zeid. *CAD/CAM Theory and Practice.* McGraw-Hill, Inc., 1991.

Appendix A

Coordinate Measuring Tool

The coordinate measuring tool is used to measure points on a cutting edge that does not have an available analytical representation or a rake face for which the distribution of the rake angle is not known. It can also be used to measure points on the machined surface for comparison with the design (i.e. the required part). It consists of a Renishaw touch probe with a 1 mm diameter ruby ball stylus and a 5-axis machining center. The probe is used to measure points on the entity of interest (i.e. edge, rake face or part surface) as decribed below.

A.1 Measuring procedure

A.1.1 Cutting edge

The first step in measuring points on a cutting edge on the ball nose of a ball end mill consists of locating the center of the ball nose, point T of the cutting edge in Figure A.1. This point is first identified visually. This initial estimation is then improved

as follows. While the probe is touching at that point, the tool is rotated; if contact is lost, then the X-coordinate of the machine is changed to restore contact. This is repeated until contact prevails for the complete 360° rotation of the tool. The next step is to measure other points on the edge. The Y-coordinate is kept constant and equal to zero. The Z-coordinate is decreased incrementally. For each decrement, the probe starts at a certain X-value: x_i, where $0 \leq x_i \leq R$ (tool radius). The tool is then rotated. If no touching is encountered, then x is decreased; otherwise, it is increased with small increments until contact is never experienced. The last value of x which involved contact is the proper one. This X-value and the corresponding z and θ (angular position of the rotating table) are then used to compute x_p and y_p, the X- and Y- coordinates of this particular point on the edge. The above is repeated until sufficient points are measured. A B-spline curve is then fitted to these points ([16]).

A.1.2 Rake face

The same set-up described above is used. For representing the rake face of the cutting edge on the ball end mill used here, a 11 × 11 grid of points was measured (only the ball nose was considered). The tool was positioned horizontally and the points were measured by approaching with the probe vertically.

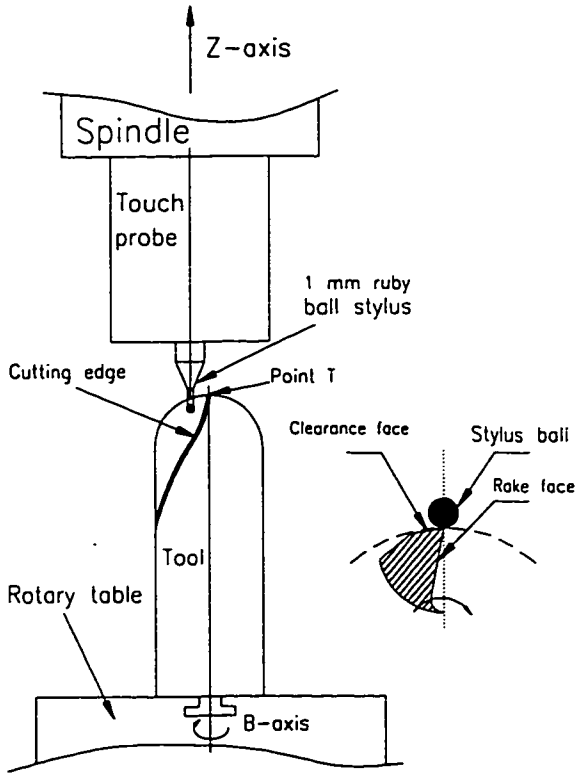


Figure A.1: Probing cutting edge on ball nose of ball end mill

Appendix B

Points measured on Cutting edge and Rake face

B.1 Points measured on cutting edges of ball nose

Points measured on three different edges (E1, E2 and E3):

E1	x	θ	z	E2	x	θ	z	E3	x	θ	z
	0	0	0		0	0	0		0	0	0
	4.828	1.847	-1.0		3.713	357.06	-0.5		3.86	348.84	-0.5
	6.753	3.847	-2.0		5.135	357.35	-1.0		5.257	347.5	-1.0
	8.135	5.862	-3.0		7.071	357.701	-2.0		6.35	346.28	-1.5
	9.217	7.45	-4.0		7.782	358.62	-2.5		7.239	346.21	-2.0
	10.113	8.904	-5.0		8.442	358.771	-3.0		8.572	345.958	-3.0
	10.838	10.5	-6.0		9.53	0.589	-4.0		9.639	345.838	-4.0
	11.422	12.274	-7.0		10.005	1.799	-4.5		10.502	346.048	-5.0

Points measured on three different edges (continued):

E1	x	θ	z	E2	x	θ	z	E3	x	θ	z
	11.904	14.184	-8.0		10.411	3.219	-5.0		11.196	346.268	-6.0
	12.285	16.195	-9.0		11.148	4.829	-6.0		11.8	346.359	-7.0
	12.618	16.674	-10.0		11.783	6.089	-7.0		12.258	346.937	-8.0
	12.821	18.714	-11.0		12.265	8.169	-8.0		12.613	347.896	-9.0
	12.948	20.804	-12.0		12.672	9.609	-9.0		12.893	348.986	-10.0
	12.974	22.088	-12.5		12.976	12.429	-10.0		13.121	349.576	-11.0
	13.20	22.556	-12.7		13.205	14.539	-11.0		13.2	351.216	-12.0
					13.357	17.039	-12.0		13.2	352.366	-12.5
					13.2	17.899	-12.5				
					13.2	18.579	-12.7				

B.2 Points measured on rake face of cutting of ball nose

The following 121 points (forming a grid of 11 by 11 points) were measured on the rake face:

-71.0	-25.001	-211.013	-72.001	-25.001	-210.639	-75.001	-26.200	-209.567
-71.0	-25.201	-211.014	-72.001	-25.201	-210.632	-75.001	-26.400	-209.567
-71.0	-25.402	-211.018	-72.001	-25.400	-210.632	-75.001	-26.601	-209.565
-71.0	-25.603	-211.010	-72.001	-25.600	-210.626	-75.001	-26.802	-209.567
-71.0	-25.803	-211.010	-72.001	-25.801	-210.619	-75.001	-27.000	-209.567
-71.0	-26.001	-211.015	-72.001	-26.001	-210.616	-75.001	-27.200	-209.575
-71.0	-26.202	-211.018	-72.001	-26.200	-210.611	-75.001	-27.40	-209.577
-71.0	-26.403	-211.015	-72.001	-26.400	-210.606	-75.001	-27.602	-209.580
-71.0	-26.601	-211.015	-72.001	-26.601	-210.601	-75.001	-27.800	-209.585
-71.0	-26.802	-211.007	-72.001	-26.801	-210.599	-75.001	-28.000	-209.590
-71.0	-27.002	-211.002	-72.001	-27.000	-210.599	-75.001	-28.201	-209.595
-71.501	-25.001	-210.842	-74.000	-25.801	-209.892	-76.001	-26.901	-209.301
-71.501	-25.201	-210.842	-74.000	-26.001	-209.885	-76.001	-27.101	-209.303
-71.501	-25.400	-210.835	-74.000	-26.200	-209.882	-76.001	-27.302	-209.311
-71.501	-25.600	-210.830	-74.000	-26.400	-209.880	-76.001	-27.500	-209.318
-71.501	-25.801	-210.822	-74.000	-26.601	-209.880	-76.001	-27.701	-209.323
-71.501	-26.001	-210.817	-74.000	-26.802	-209.880	-76.001	-27.901	-209.331
-71.501	-26.200	-210.812	-74.000	-27.000	-209.877	-76.001	-28.100	-209.341
-71.501	-26.400	-210.807	-74.000	-27.200	-209.877	-76.001	-28.300	-209.349
-71.501	-26.601	-210.807	-74.000	-27.401	-209.877	-76.001	-28.501	-209.364
-71.501	-26.802	-210.807	-74.000	-27.602	-209.877	-76.001	-28.702	-209.374
-71.501	-27.002	-210.799	-74.000	-27.800	-209.88	-76.001	-28.900	-209.384

-77.000	-27.800	-209.102	-79.001	-30.002	-208.950	-80.002	-31.501	-209.064
-77.000	-28.000	-209.115	-79.001	-30.200	-208.981	-80.002	-31.700	-209.108
-77.000	-28.201	-209.128	-79.001	-30.401	-209.008	-80.002	-31.902	-209.146
-77.000	-28.402	-209.135	-79.001	-30.601	-209.036	-80.002	-32.100	-209.184
-77.000	-28.600	-209.151	-79.001	-30.800	-209.069	-80.002	-32.301	-209.227
-77.000	-28.801	-209.163	-79.001	-31.000	-209.097	-80.002	-32.501	-209.270
-77.000	-29.001	-209.179	-79.001	-31.201	-209.133	-80.002	-32.702	-209.311
-77.000	-29.202	-209.199	-79.001	-31.402	-209.166	-80.002	-32.900	-209.351
-77.000	-29.400	-209.214	-79.001	-31.600	-209.202	-80.002	-33.101	-209.395
-77.000	-29.601	-209.232	-79.001	-31.800	-209.235	-80.002	-33.301	-209.443
-77.000	-29.801	-209.252	-79.001	-32.001	-209.273	-80.002	-33.500	-209.489
-78.000	-28.702	-208.968	-79.502	-30.700	-208.991			
-78.000	-28.900	-208.988	-79.502	-30.901	-209.021			
-78.000	-29.100	-209.002	-79.502	-31.102	-209.054			
-78.000	-29.301	-209.026	-79.502	-31.300	-209.087			
-78.000	-29.502	-209.044	-79.502	-31.501	-209.123			
-78.000	-29.700	-209.067	-79.502	-31.701	-209.158			
-78.000	-29.900	-209.090	-79.502	-31.901	-209.199			
-78.000	-30.101	-209.115	-79.502	-32.100	-209.242			
-78.000	-30.302	-209.141	-79.502	-32.301	-209.278			
-78.000	-30.500	-209.169	-79.502	-32.501	-209.318			
-78.000	-30.700	-209.191	-79.502	-32.702	-209.359			

Appendix C

C definition of the Half-edge data structure

The following is the C definition of the Half-edge data structure used in GWB.

```
typedef float          vector[4];
typedef float          matrix[4][4];
typedef short          Id;
typedef struct solid   Solid;
typedef struct face    Face;
typedef struct loop    Loop;
typedef struct halfedge   Halfedge;
typedef struct vertex  VErtex;
typedef struct edge    Edge;
typedef union niodes   Node;
```

```
struct solid
{
    Id      solidno;      /* solid identifier */
    Face    *sfaces;      /* pointer to list of faces */
    Edge    *sedges;      /* pointer to list of edges */
    Vertex  *sverts;      /* pointer to next solid */
    Solid   *nexts;       /* pointer to previous solid */
};
struct face
{
    Id      faceno;       /* face identifier */
    Solid   *fsolid;      /* back pointer to solid */
    Loop    *flout;       /* pointer to outer loop */
    Loop    *floops;      /* pointer to list of loops */
    vector   feq;         /* face equation */
    Face    *nextf;       /* pointer to next face */
    Face    *prevf;       /* pointer to previous face */
};
struct loop
{
    HalfEdge    *ledg;    /* pointer to ring of halfedges */
    Face    *lface;       /* back pointer to face */
    Loop     *nextl;      /* pointer to next loop */
    Loop     *prevl;      /* pointer to previous loop */
```

```
}
struct edge
{
    HalfEdge    *he1;     /* pointer to right halfedge */
    HalfEdge    *he2;     /* pointer to left halfedge */
    Edge     *nexte;     /* pointer to next edge */
    Edge     *preve;     /* pointer to previous edge */
}
struct halfedge
{
    Edge     *edg;     /* pointer to parent edge */
    Vertex    *vtx;     /* pointer to starting vertex */
    Loop     *wloop;    /* back pointer to loop */
    HalfEdge    *nxt;     /* pointer to next halfedge */
    HalfEdge    *prv;     /* pointer to previous halfedge */
}
struct vertex
{
    Id     vertexno;    /* vertex identifier */
    HalfEdge    *vedge;     /* pointer to a halfedge */
    vector     vcoord;     /* vertex coordinates */
    Vertex     *nextv;     /* pointer to next vertex */
    Vertex     *prevv;     /* pointer to previous vertex */
}
```

```
union nodes
{
    Solid       s;
    Face        f;
    Loop        l;
    HalfEdge    h;
    Vertex      v;
    Edge        e;
};
```

Appendix D

Analytical equations for edge

The geometry of the cutting edge on the ball nose of a ball end mill with constant helix angle can be given as:

$$\frac{D \cdot \pi}{tan(\phi)} = L \tag{D.1}$$

where L is the lead and ϕ is the helix angle. The above can be rewritten as follows:

$$tan(\phi) = \frac{\pi \cdot 2 \cdot R}{L} \tag{D.2}$$

For a given z, R is computed using:

$$R(z) = \sqrt{R_0^2 - z^2} \tag{D.3}$$

Next, ψ, the angle defining a point on the edge (measured from +ve x-axis), is determined from:

$$\psi = -\frac{z}{R_0 \cdot cotan(\phi_0)} \tag{D.4}$$

Finally, x and y coordinates of the point on the edge are computed using the following:

$$x = \sqrt{R_0^2 - z^2} \cdot cos(\psi) \tag{D.5}$$

$$y = \sqrt{R_0^2 - z^2} \cdot sin(\psi)$$

Appendix E

Extraction of modal parameters

Experimentally exciting a structure with a measured force and simultaneously measuring its response motion permits a force/motion transfer function to be evaluated. This measurement can be used to determine the structure resonant frequencies, stiffness, damping, and inertia. On the other hand, a series of such measurements can be used to perform modal analysis, defining the deformation pattern associated with each natural vibration. Although the frequency response functions may be construed as definitions of the sinusoidal input/output relationship of a structure, they can be measured using non-sinusoidal excitation forms (Lang [24]). Three different techniques are broadly used, of which the impulsive testing is by far the most rapid technique that can be applied to the study of a complex structure. Its fundamental advantage is that no shaker system needs to be mounted to the test object. Impulsive testing utilizes short duration transient force inputs (with corresponding broadband spectra) to excite all frequencies in the structure simultaneously. Impulsive testing is normally conducted using some form of an instrumented hammer. A force transducer

is mounted to the head of the hammer, measuring the force input to the structure. An accelerometer is used to measure the response of the structure. The data is analyzed with a dual-channel FFT analyzer. The two transient time histories are Fourier transformed to yield the input and output spectra. The resultant ratio of the output and input spectra is the desired frequency response function. Any of the six transfer functions which have become classics in the study of mechanical structures can be used. The choice depends on the available instrumentation. For this reason, inertance (acceleration/force) is measured here. Mobility (velocity/force) and compliance (displacement/force) can simply be derived from inertance by integration.

Mounting the accelerometer at the tip of the tool and hitting with impact hammer at the same level yields the driving point structural response at that location (it defines the interrelationships between two points in the structure as a function of frequency). A cartesian display is used, which consists of a real and imaginary description. In such a case, a resonant frequency is identified when the real component has a zero value while the imaginary component experiences a local minimum.

Plotting any of the frequency response functions with logarithmic amplitude versus logarithmic frequency provides a means of directly estimating mass, stiffness and damping parameters in systems characterized by far coupled modes (more details can be found in Lang [24] and Ewins [19]). One can also use the cartesian description of the compliance function to compute the modal parameters for the same case (i.e. well-separated modes). The equations giving the modal parameters (K, C, and M) for the case of single degree of freedom system based on the real and imaginary parts

of the compliance function are as follows:

$$\xi = \frac{\omega_{max} - \omega_{min}}{2 \cdot \omega_n} \tag{E.1}$$

where ω_{max} is the frequency above resonance where the real part of compliance reaches a peak, ω_{min} is the frequency below resonance where the real part of compliance reaches a peak (of opposite sign), and ω_n is the natural frequency.

$$K = \frac{1}{4 \cdot \xi \cdot (1 - \xi) \cdot G_{max}} \tag{E.2}$$

or

$$K = \frac{1}{4 \cdot \xi \cdot (1 + \xi) \cdot G_{min}} \tag{E.3}$$

where G_{max} is the real part of the compliance (transfer function) at ω_{max} and G_{min} is the value at ω_{min}.

$$M = \frac{K}{4 \cdot \pi^2 \cdot f_n^2} \tag{E.4}$$

and

$$C = 2 \cdot \xi \cdot (K \cdot M)^{1/2} \tag{E.5}$$

where

$$f_n = \frac{\omega_n}{2 \cdot \pi} \tag{E.6}$$

The set-up used here to perform the hammer test is shown in Figure E.1. A Hewlet Packard 5663 frequency analyzer with a frequency resolution of 2 Hz was used. The method of excitation was impact using a 806.6 gm Kistler hammer type 9726A20000. The response of the spindle and overhang at the level of the tool was measured using an accelerometer. An average of 10 samples was used and the coherence was in most cases between 0.9 and 1.0. An oscilloscope (Nicolet 3091) was used to monitor the input and output during the tests.

The frequency response (inertance) was measured for the spindle and overhang. Next, the compliance function was derived. The results of the measurement performed in $x-$ and $y-$ direction were shown in Figure 4.15. Within the frequency range of interest, two dominant modes are identified in the $y-$ direction (at 456 Hz and 553 Hz). On the other hand, four modes are clearly distinguishable in the $x-$ direction (at 313 Hz, 437 Hz, 572 Hz and 996 Hz). However, only the two most dominant ones are considered for that direction.

The approach used to extract the individuals modes in $x-$ and $y-$ directions is based on the following steps. First, fit (to the best) a single degree of freedom frequency response curve to each mode. For modes which are heavily distorted due to the coupling with another more flexible mode, such a fitting is a very crude approximation. Next, by superposing the individual modes, the resulting curve can be considered an initial representation of the measured transfer function. Finally, the parameters are modified by trial and error to improve the fitting. This can be done by using the superposed single degree of freedom transfer functions or the transfer function of a two-degree coupled system.

The first approach actually is based on the assumption that a two d.o.f system is equivalent to the superposition of two single d.o.f systems. It was used previously in the context of machine tools and produced very reliable results. Figure E.2 shows the frequency response measured for one such case. Table E.1 shows the corresponding extracted modal parameters (Ullagaddi [60]). As noticed in the figure, there are only two very well-separated modes in each direction. However, this is not the case when it comes to the structure analyzed here which exhibits coupled modes. For this reason, the second approach has to be used. Actually, the accuracy of the first approach

depends mainly on the difference between the stiffnesses of the two modes, as shown below.

A two degree of freedom system has two natural frequencies. For each one, there corresponds a natural state of vibration with a displacement configuration known as the normal modes. The latter are free vibrations that depend only on the mass and stiffness of the system (i.e. M_1, M_2, K_1, and K_2). If the system has no dynamic coupling, then the two natural frequencies are given by:

$$\omega_1^2 = \frac{B+C}{A} \tag{E.7}$$

$$\omega_2^2 = \frac{B-C}{A} \tag{E.8}$$

where

$$A = 2 \cdot M_1 \cdot M_2$$

$$B = M_2 \cdot K_1 + M_1 \cdot (K_1 + K_2)$$

$$C = ((M_2 \cdot K_1 + M_1 \cdot (K_1 + K_2))^2 - 4 \cdot M_1 \cdot M_2 \cdot K_1 \cdot K_2)^{1/2}$$

The use of the following equations

$$\omega_1^2 = \frac{K_1}{M_1} \tag{E.9}$$

$$\omega_2^2 = \frac{K_2}{M_2} \tag{E.10}$$

to determine the modal parameters for the case of a two degree of freedom system is only an approximation. The amount of error introduced is directly related to the level of coupling between the two modes of the system. It can be shown that equations E.7 and E.8 and equations E.9 and E.10 are almost equivalent for the particular case of a two d.o.f system characterized by $K_2 \gg K_1$. In fact, C (defined above) can be

rewritten as follows:

$$
\begin{aligned}
C^2 &= (M_2 \cdot K_1 + M_1 \cdot (K_1 + K_2))^2 - 4 \cdot M_1 \cdot M_2 \cdot K_1 \cdot K_2 \\
C^2 &= M_2^2 \cdot K_1^2 + M_1^2 \cdot (K_1 + K_2)^2 + 2 \cdot M_1 \cdot M_2 \cdot K_1 \cdot (K_1 + K_2) \\
&\quad -4 \cdot M_1 \cdot M_2 \cdot K_1 \cdot K_2 \\
C^2 &= M_2^2 \cdot K_1^2 + M_1^2 \cdot (K_1 + K_2)^2 + 2 \cdot M_1 \cdot M_2 \cdot K_1^2 + 2 \cdot M_1 \cdot M_2 \cdot K_1 \cdot K_2 \\
&\quad -4 \cdot M_1 \cdot M_2 \cdot K_1 \cdot K_2 \\
C^2 &= M_2^2 \cdot K_1^2 + M_1^2 \cdot (K_1 + K_2)^2 + 2 \cdot M_1 \cdot M_2 \cdot K_1^2 - 2 \cdot M_1 \cdot M_2 \cdot K_1 \cdot K_2 \\
C^2 &= M_2^2 \cdot K_1^2 + M_1^2 \cdot (K_1 + K_2)^2 + (-2 \cdot M_1 \cdot M_2 \cdot K_1^2 + 4 \cdot M_1 \cdot M_2 \cdot K_1^2) \\
&\quad -2 \cdot M_1 \cdot M_2 \cdot K_1 \cdot K_2 \\
C^2 &= (M_2^2 \cdot K_1^2 + M_1^2 \cdot (K_1 + K_2)^2 - 2 \cdot M_1 \cdot M_2 \cdot K_1^2) + 2 \cdot M_1 \cdot M_2 \cdot K_1 \cdot (2 \cdot K_1 - K_2)
\end{aligned}
$$

for the case where $K_2 \gg K_1$, we can set $2 \cdot K_1 - K_2 \approx -K_2$. Then,

$$
\begin{aligned}
C^2 &= M_2^2 \cdot K_1^2 + M_1^2 \cdot (K_1 + K_2)^2 - 2 \cdot M_1 \cdot M_2 \cdot K_1^2 - 2 \cdot M_1 \cdot M_2 \cdot K_1 \cdot K_2 \\
C^2 &= M_2^2 \cdot K_1^2 + M_1^2 \cdot (K_1 + K_2)^2 - 2 \cdot M_1 \cdot M_2 \cdot K_1 \cdot (K_1 + K_2) \\
C^2 &= (M_2 \cdot K_1 - M_1 \cdot (K_1 + K_2))^2
\end{aligned}
$$

Consequently, the natural frequencies of a two-degree coupled system (equations E.7 and E.8) become:

$$
\begin{aligned}
\omega_1^2 &= \frac{M_2 \cdot K_1 + M_1 \cdot (K_1 + K_2) + M_2 \cdot K_1 - M_1 \cdot (K_1 + K_2)}{2 \cdot M_1 \cdot M_2} \qquad \text{(E.11)} \\
\omega_1^2 &= \frac{2 \cdot M_2 \cdot K_1}{2 \cdot M_2 \cdot M_1} \\
\omega_1^2 &= \frac{K_1}{M_1}
\end{aligned}
$$

$$
\omega_2^2 = \frac{M_2 \cdot K_1 + M_1 \cdot (K_1 + K_2) - M_2 \cdot K_1 - M_1 \cdot (K_1 + K_2)}{2 \cdot M_1 \cdot M_2} \qquad \text{(E.12)}
$$

$$\omega_2^2 = \frac{2 \cdot M_1 \cdot (K_1 + K_2)}{2 \cdot M_1 \cdot M_2}$$
$$\omega_2^2 \approx \frac{K_2}{M_2}$$

Physically what the above means is that when the difference in stiffness between the two modes is very large, we have at the mode whose natural frequency is very close to the natural frequency of the stiffer element of the system a tendency of domination of that stiffer element. The other element will be vibrating as if it is an integral part of the first one. On the other hand, when the system is vibrating at the other mode, the stiffer element will have very little vibration compared to the other less stiff element (whose natural frequency is very close to the natural frequency of the system at that mode), as if only one element is vibrating. If, in addition, $\omega_2 \gg \omega_1$, then the above trend is even more accentuated. In that particular case, we can even refer to the first mode as mode of the first element (with modal parameters M_1, K_1 and C_1) and the second mode as mode of the second element (with modal parameters M_2, K_2 and C_2). This was the case in Figure E.2 where $\frac{K_2}{K_1} = 21$ (i.e. $K_2 \gg K_1$) and $\omega_2 \gg \omega_1$; and consequently the system could be treated as a two single d.o.f system. The difference between the natural frequencies computed using equations E.7 and E.8 and those using equations E.9 and E.10 is less than 1.2 %. In such a case, we can refer to the first mode as the Spindle mode and the second one as the Tool mode.

The above doesn't apply to the tool and tool holder analyzed here. One of the reasons is the presence of the rotating dynamometer (mounted to the spindle) and the tool adaptor (mounted between the dynamometer and the tool). This has actually altered the dynamics of the structure significantly. Instead of having two dominant well-separated modes in each direction, the structure now exhibits more modes and

the level of coupling is much higher. To extract the different modes in this case the second approach is required. It consists of the following. After extracting the modal paramaters using superposed single d.o.f modes, the values are used as initial input to the more exact representation of the system (i.e. as a two d.o.f coupled system). For such a system, the frequency response is given by the following equation:

$$\frac{X_2}{F} = \frac{A}{B} \tag{E.13}$$

where

$$\begin{aligned} A &= M_1 \cdot S^2 + (C_1 + C_2) \cdot S + (K_1 + K_2) \\ B &= (M_2 \cdot S^2 + C_2 \cdot S + K_2) \cdot (M_1 \cdot S^2 + (C_1 + C_2) \cdot S + (K_1 + K_2)) - (C_2 \cdot S + K_2)^2 \end{aligned}$$

The above can be used to obtain a close fit to the measured frequency response in the region of the two modes of interest. This can be done by trial and error.

Additional measurements were made to get a better insight on the dynamics of the system. A hammer test was performed for each of the following cases:

- Spindle only
- Spindle and dynamometer
- Spindle, dynamometer and tool adaptor

It was not possible to get good coherence in the tests made in the $y-$ direction. On the other hand, the tests conducted in the $x-$ direction were more reliable and yielded the frequency response curves shown in Figure E.3. Part (d) shows the measurements for the complete structure of the tool and tool holder (i.e. spindle, dynamometer, tool adaptor, and tool). It is clear from part (a) of the figure that the spindle exhibits a

mode with a natural frequency about 336 Hz in the $x-$ direction. In part (d), we have mode at 312 Hz, which implies that the main contribution to this mode comes from the spindle. Part (b) shows two additional modes at 540 Hz and 608 Hz. These can be attributed mainly to the dynamometer. The corresponding modes in part (d) are at 440 Hz and 576 Hz. Next, if we compare parts (c) and (b) of the figure, we can notice the effect of introducing the tool adaptor. In fact, now the second and third modes have frequencies *shifted* to 456 Hz and 576 Hz. An additional mode which becomes quite significant when the tool is mounted is at 1000 Hz. This mode is much less flexible than the other two and consequently can be neglected when modeling the dynamic of the system.

Finally, it should be noted that if the investigated frequency span is increased to 2 KHz, then another very significant mode is captured (as was depicted in Figure 4.15). Its natural frequency is at 1408 Hz. Apparently, the major contribution to this mode comes from the tool.

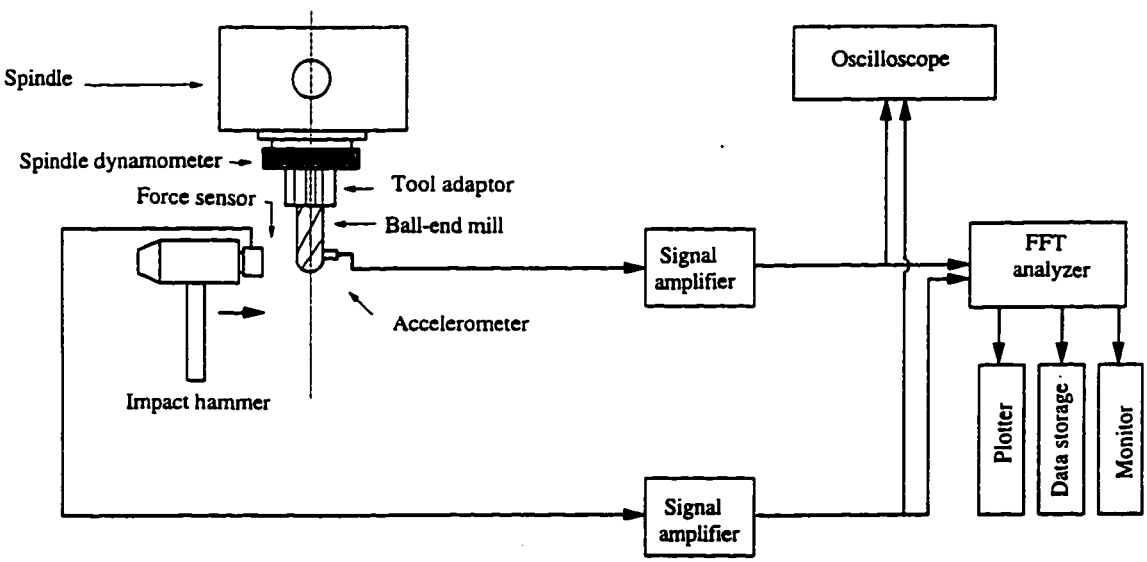
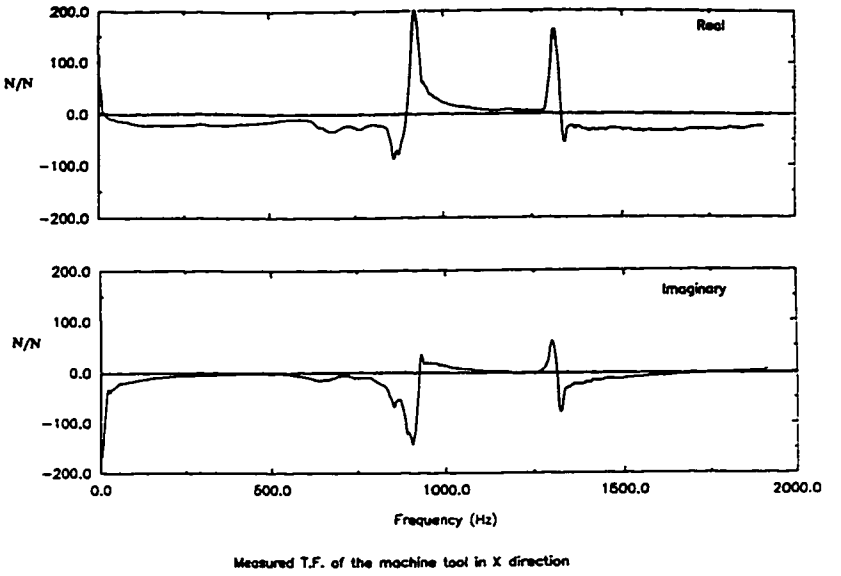


Figure E.1: Experimental set-up for conducting hammer test



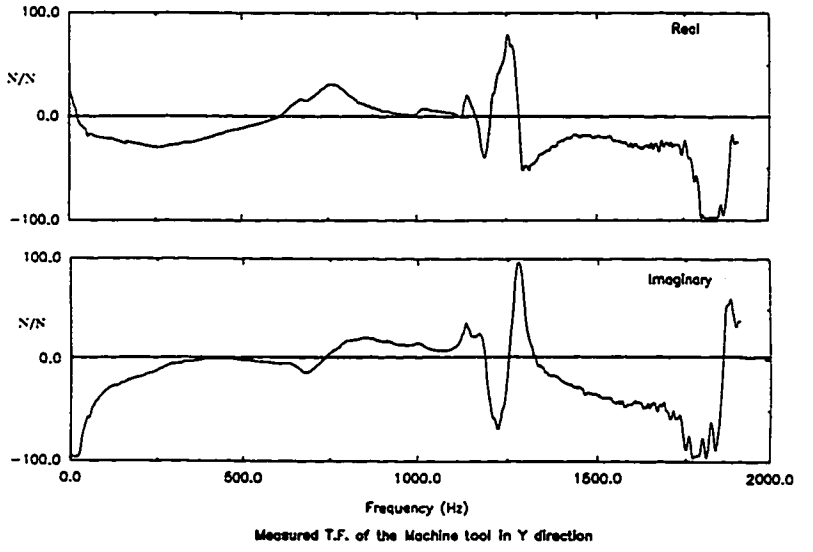
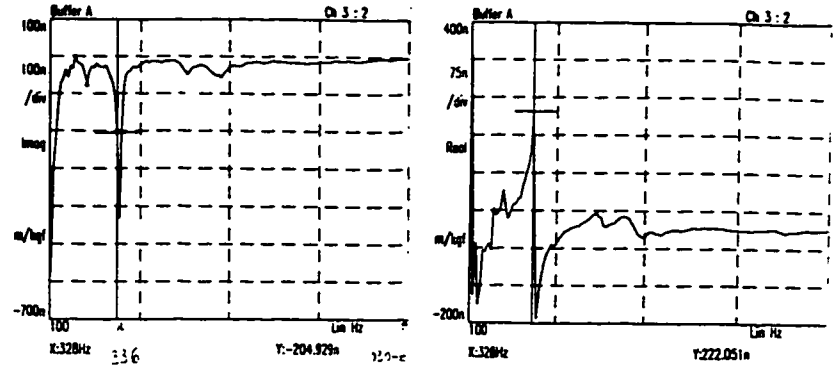


Figure E.2: Frequency response for a case of two well-separated modes

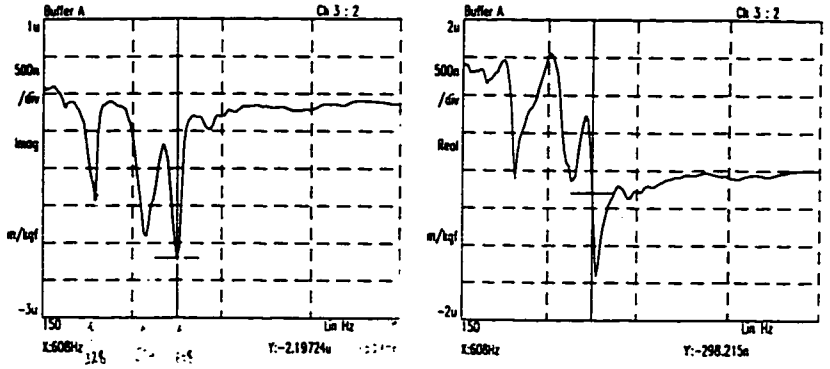
mode	Nat.Freq. ω_n (Hz)	damping ratio(%)	Mass (Kg)	Damping (kg/sec)	Stiffness (N/m)
M_x	660	0.087	10.0	7112	1.7E8
T_x	1370	0.019	0.11	35.32	8.1E6
M_y	710	0.096	8.0	6848	1.6E8
T_y	1330	0.019	0.11	34.91	7.7E6

The Modal Parameters of the machine tool

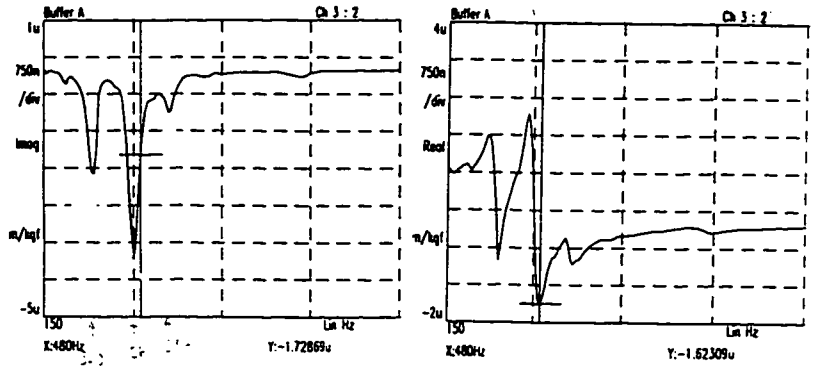
Table E.1: Extracted modal parameters for a case of two well-separated modes



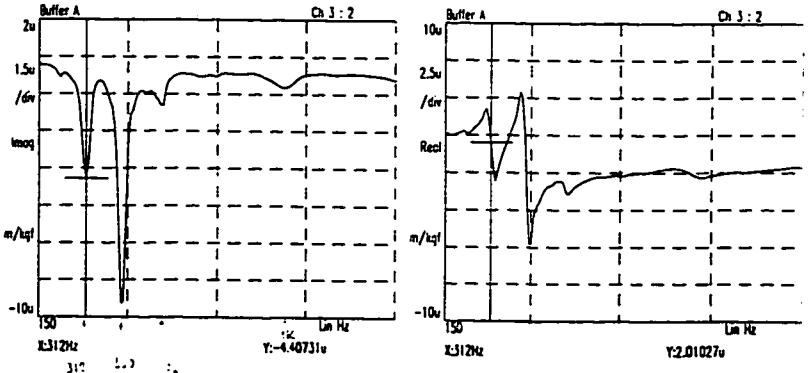
(a)



(b)



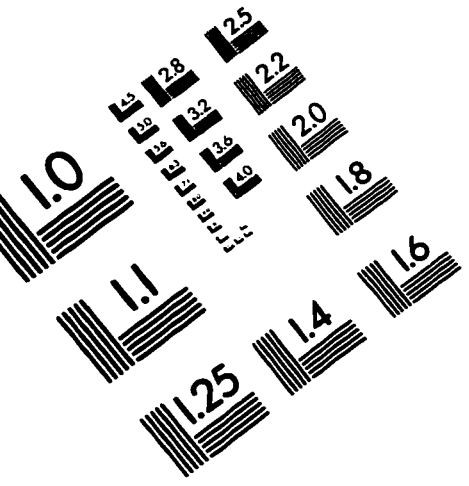
(c)

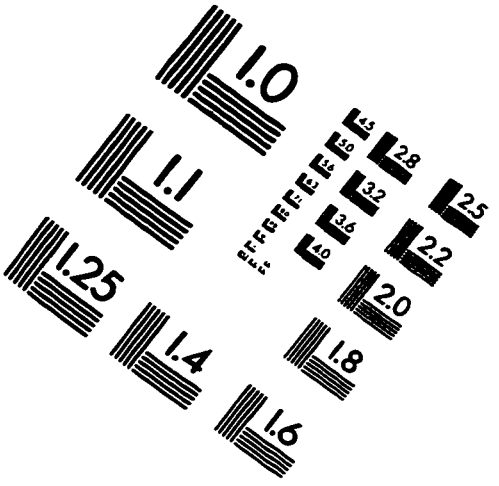


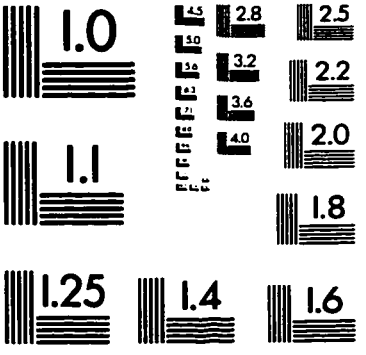
(d)

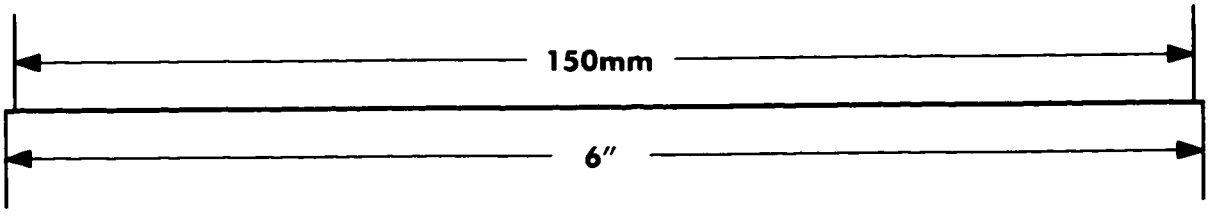
Figure E.3: Frequency response in x-direction for different spindle overhang conditions

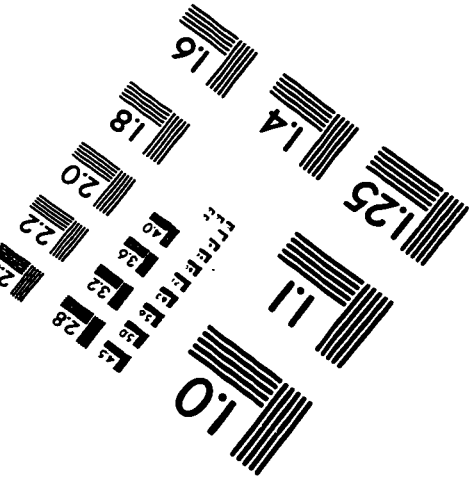
IMAGE EVALUATION TEST TARGET (QA-3)

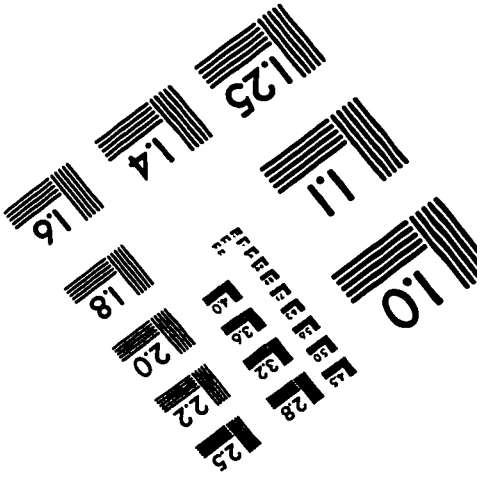












APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989