CHATTER AVOIDANCE IN MILLING

By

Philip K. F. Chan

B.A.Sc (Mechanical Engineering)

University of British Columbia

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

MECHANICAL ENGINEERING

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

August 1990

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Mechanical Engineering
The University of British Columbia
2075 Wesbrook Place
Vancouver, Canada
V6T 1W5

Date: SEPT 28, 1990

Abstract

One of the major limitations on productivity in metal cutting is chatter. Chatter is a form of unstable self-excited vibration which causes poor surface finish, as well as cutter and machine tool damage. The investigation of chatter suppression in milling using continuously variable spindle speed is presented in this thesis.

The fundamental mechanism in regenerative chatter is due to favorable phasing between the inner and outer modulations on the chip thickness. In this thesis, the spindle speed is sinusoidally varied to prevent the dynamic cutting process from locking on to a constant phase shift and causing unstable cutting, or chatter. Because of the nonlinearities and complexities of the process, time domain simulation of the dynamic cutting process has been modelled. The influence of various parameters, such as axial depth of cut, process damping from flank interference, and amplitude and frequency of speed variation have been investigated using the simulation model. The trends predicted by simulation results have been experimentally verified using cutting tests on a milling machine.

It has been concluded from simulation and milling tests that a variable spindle speed can partially increase the chatter limit, but can never totally prevent chatter. The variable spindle speed strategy is incorporated into a proposed in-process chatter detection and avoidance algorithm. The milling process is monitored using the sound pressure signal measured by a microphone. When the amplitude of the sound spectrum near the natural frequency exceeds a threshold value, chatter has been detected and the spindle speed is oscillated until stability is regained. The proposed algorithm is implemented on line and experimental results are presented.

Table of Contents

List of Tables

List of Figures

Nomenclature

a : axial depth of cut (milling) or width of cut (general) [mm]

a_{crit} : minimum depth of cut at which chatter develops [mm]

a_{lim} : depth of cut at which chatter develops [mm]

A_{par} : ploughing contact area parallel to velocity [mm^2]

A_{pen} : ploughing contact area perpendicular to velocity [mm^2]

b : width of cut in milling [mm]

c : feed per tooth [mm/tooth-revolution]

dn : amplitude of sinusoidal spindle speed variation

dt : discrete intervals in digital simulations

f : feed rate [mm/sec]

F : total cutting force [N]

F_m : mean component of cutting force [N]

F_{par} : ploughing force parallel to velocity [N]

F_{pen} : ploughing force perpendicular to velocity [N]

F_r : radial cutting force [N]

$F_{r,po}$: radial ploughing force [N]

$F_{r,s}$: radial shearing force [N]

F_t : tangential cutting force [N]

$F_{t,po}$: tangential ploughing force [N]

$F_{t,s}$: tangential shearing force [N]

F_v : variable component of cutting force [N]

F_x : cutting force in (X) direction [N]

$F_{x'}$: cutting force in orthogonal (X') direction [N]

F_y: cutting force in (Y) direction [N]

$F_{y'}$: cutting force in orthogonal (Y') direction [N]

G: system transfer function

h : uncut chip thickness [mm]

h_m: mean uncut chip thickness [mm]

h_v: variable component of uncut chip thickness [mm]

K_s : specific cutting pressure [N/mm^2]

l_{pl} : primary flank face length [mm]

n : spindle speed [rps]

n_f : frequency of sinusoidal spindle speed variation [Hz]

n_o : nominal spindle speed [rps]

N : number of full waves between inner and outer modulation

r_1 : cutting force ratio

$r_{o,i}$: runout of tooth (i) [mm]

$\overline{VB}$: length of flank wear land [mm]

v : relative velocity between cutter and workpiece [mm/sec]

x : deflection in feed (X) direction [mm]

x' : deflection in orthogonal (X') direction [mm]

X : feed direction axis

X' : modal axis of tool-workpiece vibration

y : deflection in normal (Y) direction [mm]

y' : deflection in orthogonal (Y') direction [mm]

Y : normal direction axis

Y' : modal axis of tool-workpiece vibration

y : present vibration (inner modulation)

y_o : previous vibration (outer modulation)

Y : amplitude of inner (present) modulation

Y_o : amplitude of outer (previous) modulation

$z_{e,i}$: radial deflection minus runout for tooth (i) [mm]

z_t : number of teeth on the cutter

z_i : deflection in radial direction of tooth (i) [mm]

z_{min} : minimum previous deflection in radial direction

Z : radial direction axis (stationary or rotational)

α : angle between (X) and (X') axes

β : angle between (Y) axis and cutting force direction in turning

$d\phi$: angle increment between iteration steps

ε : phase shift between inner and outer modulation

γ : clearance angle of primary flank face

γ_e : angle of instantaneous tooth velocity

ϕ : immersion angle in milling

ϕ_i : immersion angle of tooth (i)

τ : tooth period [sec]

ω: vibration (chatter) frequency [rad/sec]

Acknowledgement

I would like to thank my supervisor Dr. Yusuf Altintas for his support and instruction during this work for which I am grateful. I would also like to thank Darcy Montgomery, all the other graduate students, research engineers, and technicians for their valuable technical assistance during my many years at UBC. I am indebted to my family, whose support and patience have made this work possible.

This work was supported by NSERC of Canada under Grant Number OGP-0006164, the Centre for Integrated Computer Systems Research (CICSR) UBC, and the BC Advanced Systems Institute.

Chapter 1

Introduction

Milling is a process which is widely used in the aerospace, automotive, and defense industries. Wing sections for commercial and high performance aircrafts, and stamping dies for automotive parts are all machined by milling. Large amounts of metal must be removed by milling in many instances. The percentage of the rough workpiece that is converted into metal chips can be as high as 90%. Thus the metal removal rate in these operations is an important factor in its economic efficiency. Much effort in metal cutting research is devoted to increasing the metal removal rate.

The maximum metal removal rate in present machines is often not limited by spindle motor power, but by the development of unstable self-excited vibrations, or what is called *chatter*. Chatter is a consequence of the relative compliance between the tool and the workpiece. The development of chatter produces large vibrations which can leave undesired deep marks on the workpiece surface as well as irreparable damage to cutter and machine tool. In conventional machining, metal removal rates are lowered to avoid chatter.

In light of this limitation to the metal removal rate, chatter has become a significant research topic. Some researchers study the physics of the dynamic cutting process, the fundamental basis behind chatter development. Other researchers, including the author, investigate different machining strategies to avoid or suppress chatter.

In this thesis, the investigation of spindle speed oscillation as a means of suppressing chatter in milling is presented. The following chapter reviews the fundamental theories

and models relevant to the work performed in this research. Basic metal cutting models, chatter models, and present chatter avoidance strategies are discussed.

Chapter 3 describes the models used to investigate the strategy of chatter suppression by continuously variable spindle speed. A time domain simulation of turning develops an understanding of how variable spindle speed increases cutting stability, while a time domain milling simulation is presented to determine the effects of speed oscillation on milling.

Chapter 4 discusses milling results from both simulation and experiment. Comparison between fixed and variable speed cutting simulations shows trends that are verified by experimental cutting tests.

Chapter 5 presents experimental results from the complete automatic chatter detection and suppression system. Experience from tests shown in Chapter 4 is used to develop this system.

Conclusions drawn from the research are summarized in Chapter 6. Future work relevant to the development of this research topic is also presented.

Chapter 2

Literature Review

2.1 Introduction

Metal cutting is a complex process where research efforts are concentrated in different areas. This chapter briefly reviews the areas which are relevant for the proper understanding of the work described in the rest of this thesis. Metal removal is discussed in terms of a static process, dynamic process, orthogonal turning, and milling. Self-excited vibrations are introduced and the necessary stability criteria are reviewed. Finally, the present strategies in chatter suppression are explained in light of the objective of this thesis.

2.2 Static Cutting Process

The metal cutting process is defined as the removal of thin layers of metal by a wedge-shaped tool. Static analysis of this operation considers both the tool and workpiece as rigid and nonvibratory. Research is still very active in the area of static cutting. Thus this section can only review the basics of the process.

The case of two dimensional orthogonal cutting is often used to explain the mechanisms of metal removal. Here, a tool with a straight edge perpendicular to the cutting velocity removes chips from a workpiece (see Figure 2.1). In practice, all cutting operations are three dimensional, otherwise known as oblique cutting. In this case, the cutting edge may not be straight and the cutting velocity is not necessarily perpendicular to it.

Due to the complexities of oblique cutting, only orthogonal cutting is considered in this thesis. In review, two dimensional orthogonal cutting assumes,

- the cutting edge is straight
- the cutting velocity is perpendicular to the cutting edge
- deformation is uniform along the cutting edge
- there is no side spreading of work material.

The orthogonal process considered is shown in Figure 2.1. As the workpiece moves towards the tool at relative velocity (v), the metal chip with a width of cut (a) and an uncut chip thickness (h) is sheared from the workpiece along the region from A to B. Different models have been used to describe the mechanisms in the shear zone; most researchers model it as either a thin plane zone such as Merchant [9], or a thick deformation zone such as Palmer and Oxley [11] and Lee and Shaffer [7]. From these models, shear angle predictions can be made and consequently, cutting force predictions. Without recapitulating all the equations necessary in deriving the force equations, it is suffice to note here that these models yield cutting force equations which are dependant on the yield shear stress of the workpiece, the shear angle, the rake angle, and the friction coefficient between the tool and the workpiece, as well as the cutting parameters of cutting speed, uncut chip thickness, and width of cut. Due to difficulties in predicting the deformation mechanism in the shear zone, the cutting forces are often represented by a simpler set of mechanistic equations,

$$F_t = K_s a h \tag{2.1}$$

$$F_r = r_1 K_s a h \tag{2.2}$$

where F_t = cutting force in the tangential direction

F_r = cutting force in the radial direction

K_s = specific cutting pressure

a = depth of cut

h = uncut chip thickness

r_1 = cutting force ratio.

The two coefficients, (r_1) and (K_s), are experimentally determined for a given tool-work material combination. These are the force equations used in this thesis for the shearing process.

2.3 Turning Geometry

Most research in metal cutting stability has been centered on orthogonal turning; thus it is worthwhile to quickly review the important aspects of turning geometry. Since orthogonal turning is usually plunge cutting, the geometry of plunge cutting is shown in Figure 2.2, where the tool is facing the workpiece. The width of cut (a) in Equation 2.1 and Equation 2.2 is measured in the longitudinal direction. The uncut chip thickness (h) measured in the radial direction in plunge turning is equal to the feed per revolution (c) measured in length per revolution. Turning is a continuous, single-point metal removal process where the relative cutting velocity is a consequence of the workpiece radius and spindle speed (n). The tool, plunging at a feed rate of (f) meters per second, is responsible for maintaining a certain chip thickness ($h = c = \frac{f}{n}$). If the spindle speed and feed rate are constant, it follows from Equations 2.1 and 2.2 that the cutting forces in turning are constant in magnitude and resultant direction.

2.4 Milling Geometry

Milling is geometrically different from turning. Figure 2.3 shows that the cutting tool, which typically has multiple cutting edges, rotates instead of the workpiece. The depth of cut (a) is measured in the axial direction while the uncut chip thickness (h) is measured normal to the cut surface. As the cutter encounters the workpiece with radial width of cut (b), it only cuts in its front half, leading to engagement and disengagement of the cutting edges and an interrupted cutting process. As a result of the relative feed (f) of the workpiece towards the rotating cutter, the uncut chip thickness varies continuously. The instantaneous uncut chip thickness for small feed rates can be closely approximated by,

$$\begin{aligned} h(\phi) &= c\sin(\phi) \qquad (2.3)\\ \text{where } h(\phi) &= \text{instantaneous chip thickness}\\ c &= \text{nominal feed per tooth period}\\ \phi &= \text{immersion angle.} \end{aligned}$$

Therefore, the cutting force magnitude and directions also change with time. The radial and tangential forces are thus given by,

$$F_t = K_s a c \sin(\phi) \qquad (2.4)$$

$$F_r = r_1 K_s a c \sin(\phi). \qquad (2.5)$$

The force components in the stationary feed (X) and normal (Y) directions are then given by,

$$F_x = F_r \sin(\phi) + F_t \cos(\phi) \qquad (2.6)$$

$$F_y = F_r \cos(\phi) - F_t \sin(\phi). \qquad (2.7)$$

2.5 Dynamic Cutting

The static cutting model presented in previous sections assumes no relative vibration between tool and workpiece. All relative motion is a consequence of the prescribed cutting conditions of cutting speed and feed rate. However, vibration develops due to the relative compliance between the workpiece and the tool in practical cutting. Different theories exist for force predictions in dynamic cutting, two of which are reviewed here.

Tobias [19] looks at the dynamic cutting process in terms of small variations about a nominal steady-state condition. Consider an orthogonal cutting process where the workpiece is rigid, and the tool has single degree of freedom dynamics in the feed direction. The dynamics of the system during cutting is represented by,

$$m\ddot{y}(t) + c_d\dot{y}(t) + ky(t) = -dF,$$

where (m), (c_d), and (k) are the structural parameters of the system and (dF) is the force variation. He expresses the force variation by,

$$\begin{aligned} dF &= k_1 dy + k_2 dc + k_3 dn \qquad (2.8)\\ \text{where } k_1 &= (\frac{\partial F}{\partial y})_{dc=dn=0}\\ k_2 &= (\frac{\partial F}{\partial c})_{dy=dn=0}\\ k_3 &= (\frac{\partial F}{\partial n})_{dc=dy=0}\\ c &= \text{feed rate}\\ n &= \text{spindle speed.} \end{aligned}$$

Closer analysis results in different representations of the last two cutting constants in Equation 2.8,

$$k_2 = (k_s - k_1)\frac{2\pi}{n} \qquad (2.9)$$

$$
\begin{aligned}
k_3 &= k_n - (k_c - k_1)\frac{c_o}{n} \qquad (2.10)\\
\text{where } k_s &= (\frac{\partial F_o}{\partial y_o})_{dn=0}\\
k_n &= (\frac{\partial F_o}{\partial n_o})_{dy=0}\\
F_o &= \text{steady-state cutting force}\\
c_o &= \text{steady-state feed rate}\\
y_o &= \text{steady-state displacement normal}\\
&\quad \text{to the cut surface,}
\end{aligned}
$$

where (k_n) and (k_s) are experimentally determined static cutting coefficients, and (k_1) is the only dynamic constant. The coefficient (k_n) is the static cutting coefficient as a function of cutting speed, while (k_s) is the static cutting coefficient as a function of uncut chip thickness. Tobias terms (k_1) as the dynamic *chip thickness coefficient*, $(k_s - k_1)$ as the *penetration coefficient*, and $(k_n - (k_c - k_1)\frac{c_o}{n})$ as the *cutting speed coefficient*. When the spindle speed (n) is not directly influenced by the vibration (i.e. $dn = 0$), Equation 2.8 reduces to,

$$dF = k_1 dy + (k_s - k_1)\frac{2\pi}{n} dc. \qquad (2.11)$$

The vibration can be represented by,

$$y = Y e^{\delta t}\cos(\omega t),$$

where (ω) is the chatter frequency and (δ) is the total damping ratio in the process. Subsequently, Tobias derives the following equation of motion for the whole system,

$$\ddot{y} + \omega_o^2[\frac{2\zeta}{\omega_o} + F_2\frac{k_1}{k} + \frac{k_s - k_1}{nk}]\dot{y} + \omega_o^2[1 + F_1\frac{k_1}{k}]y = 0. \qquad (2.12)$$

The original structural parameters of the vibration system are the natural frequency (ω_o), and the damping factor (ζ). The variables (F_1) and (F_2) are functions of the total

damping, chatter frequency, and spindle speed,

$$\begin{aligned} F_1 &= 1 - e^{\frac{\delta}{n}}(\cos(\frac{\omega}{n}) + \frac{\delta}{n}\sin(\frac{\omega}{n})) \\ F_2 &= \frac{1}{\omega}e^{\frac{\delta}{n}}\sin(\frac{\omega}{n}). \end{aligned} \tag{2.13}$$

The stability of the system is dependant on the velocity coefficient term in Equation 2.12. The term is equal to (-2δ). The sign of (-2δ) determines the stability,

$$\text{If } -2\delta = [\frac{2\zeta}{\omega_o} + F_2\frac{k_1}{k} + \frac{k_s - k_1}{nk}] = \begin{cases} > 0 & \text{, then stable} \\ < 0 & \text{, then unstable.} \end{cases} \tag{2.14}$$

The computation of the total damping (δ) is difficult due to the number of variables involved (refer to Equations 2.13 and 2.14).

Tlusty [18] looks at the same problem from the standpoint of dynamic cutting force coefficients. Instead of the constant chip thickness generated during steady-state orthogonal cutting, the chip thickness changes continuously due to harmonic vibration of the tool, as shown in Figure 2.4. The dynamic portion of the cutting forces can thus be rewritten as,

$$F_t = K_s a(Y - Y_o) \tag{2.15}$$

$$F_r = r_1 K_s a(Y - Y_o) \tag{2.16}$$

where Y = amplitude of inner (present) modulation

Y_o = amplitude of outer (previous) modulation.

Tlusty replaces the static coefficients, (r_1 and K_s) in Equations 2.15 and 2.16, with dynamic cutting force coefficients $(K_{di}, K_{do}, K_{ci}, K_{co})$,

$$F_t = a(K_{ci}Y + K_{co}Y_o) \tag{2.17}$$

$$F_r = a(K_{di}Y + K_{do}Y_o). \tag{2.18}$$

The subscripts "i" and "o" refer to the effect of the inner and outer modulation, respectively on the cutting forces. Subscript "d" represents the direct effect of radial vibrations on the radial force, while subscript "c" represents the cross effect of radial vibrations on the tangential force. All four coefficients are assumed to be complex so that,

$$K_{ij} = Re[K_{ij}] + jIm[K_{ij}].$$

The chip thickness effect and penetration effect in Tobias' model are accounted for by these coefficients in a different manner. The real part of each coefficient produces a cutting force which is in phase with the respective modulation, i.e. a force due to the contribution of the chip thickness. On the other hand, the imaginary part produces a cutting force which is in phase with the first derivative or velocity of the modulation, or in other words a damping force. The necessity of including damping forces will be discussed in Section 2.8. Even though some simplifying assumptions can be made about these coefficients, much experimental work is still required to obtain the remaining ones. So far, there are wide discrepancies in the measured values of these coefficients from different research groups.

2.6 Chatter

Chatter is a form of unstable self-excited vibration in dynamic metal cutting. A characteristic of self-excited vibration is the dependency of the externally applied forces on the motion of the system. Thus it is unlike forced vibration where the externally applied forces and the motion of the system are independent. There are mainly two types of chatter observed in milling [6] : a) mode coupling and b) wave regeneration.

Mode coupling can only occur when the relative motion between the cutter and workpiece exists in at least two directions. Figure 2.5 shows a vibratory system where the tool vibration is assumed to be in two directions. The workpiece, assumed rigid, traverses

towards the cutter at constant velocity (v). If the cutter simultaneously vibrates in both directions at the same frequency but with a phase difference, an elliptical motion would result. Assuming the tool motion is as shown in the figure, the cutting force acts with the tool motion from B to A and against the motion from A to B. Since the path from B to A is deeper, the larger chip thickness produces a larger force from B to A than from A to B. Thus the energy imparted to sustain the motion is greater than the energy to impede the motion. Over time, this periodic surplus of energy can lead to vibrational instability, or chatter.

Wave regeneration is a phenomenon based on the fact that metal cutting is often a case of a vibratory tool removing a wavy surface left by a previous cut. The same two degree of freedom model from above can be used (see Figure 2.6) to describe this process. If there is relative motion between the tool and workpiece during cutting, then the surface produced would have waves as shown. This wavy surface is removed by a subsequent pass (next revolution in turning, next tooth in milling), generating a periodically variable cutting force on the tool. This force modulation is dependant on the outer surface modulation as well as the tool motion at the time it removes the chip. As a result of the variable cutting force, the tool leaves a wavy surface; thus wavy surfaces are continuously regenerated. Given the proper cutting conditions, wave regeneration can lead to larger and larger waves, i.e. unstable cutting.

2.7 Vibrational Stability

The stability limits for both types of chatter can be derived for the two degree of freedom planing model [6] by making a couple of simplifying assumptions. First, all teeth are assumed to have the same direction of cut, and second, the system is assumed to be completely linear.

The chatter limit for wave regeneration is derived first (refer to Figure 2.6). In this case, the chip thickness (h) is assumed to be modulated by the present vibration (y) as well as the vibration left by the last cut (y_o) (the inner and outer modulation, respectively),

$$\begin{aligned} h &= h_m + y_o - y \\ \text{where } h_m &= \text{mean uncut chip thickness.} \end{aligned}$$

Since the modulations are harmonic, the vibrations are represented by,

$$\begin{aligned} y_o &= Y_o \sin(\omega t) \\ y &= Y \sin(\omega t + 2\pi N + \varepsilon), \end{aligned}$$

that is two sine waves with (N) modulations plus a phase shift (ε) between them. Consequently, the chip thickness can be expressed as,

$$\begin{aligned} h &= h_m + h_v \qquad (2.19) \\ \text{where } h_v &= y_o - y \\ &= \text{variable component of the} \\ &\quad \text{uncut chip thickness.} \end{aligned}$$

From the static cutting model, the total cutting force is given by,

$$\begin{aligned} F &= K_s a h \qquad (2.20) \\ &= K_s a h_m + K_s a h_v \\ &= K_s a h_m + K_s a (y_o - y) \qquad (2.21) \\ &= F_m + F_v. \end{aligned}$$

The total force, like the chip thickness, is the sum of a mean component (F_m) and a variable component (F_v). The amplitude of modulation is given by the product of the

variable component of the force and the system transfer function,

$$y = F_v G(\omega). \tag{2.22}$$

This dynamic cutting model is a closed loop system which can be represented by the block diagram in Figure 2.7. The cutting force (F_v) is a function of the chip thickness (h) which is modulated by the inner and outer modulation (y_o and y). In turn, the inner and outer modulations are dependant on the cutting force (F_v).

At the limit of stability, the amplitude of the inner and outer modulations are equal ($Y_o = Y$), and thus

$$y_o = ye^{-j\omega\tau} = ye^{-j\varepsilon}, \tag{2.23}$$

where ($e^{-j\omega\tau}$) is a time delay term with (τ) representing the time delay between tooth passes, and (ω) is the vibration frequency. The product ($\omega\tau$) is physically equivalent to the phase shift (ε) between the inner and outer modulations. Ignoring the mean components, (h_m) and (F_m), Tlusty derived the axial depth of cut at the stability limit [6],

$$a_{limit} = -\frac{1}{2K_s Re[G(\omega)]}. \tag{2.24}$$

Thus the minimum or critical limit of stability is given by,

$$\begin{aligned} a_{crit} &= -\frac{1}{2K_s Re[G(\omega)]_{min}} \\ \text{where } Re[G(\omega)]_{min} &= \text{minimum signed value of} \\ &\quad \text{the real part of } G(\omega). \end{aligned} \tag{2.25}$$

The important results from this analysis are: 1) the depth of cut is the most limiting machining parameter in chatter generation, and 2) assuming the specific cutting pressure (K_s) is constant, the real receptance ($Re[G(\omega)]$) is the only variable. Despite the simple model used in the derivation, it shows that the structural dynamics of the system (represented by $G(\omega)$) is very important in the determination of stability.

The limit of stability for mode coupling chatter can also be determined using the same equations. Here, the surface to be removed is flat; thus $y_o = 0$ in Equation 2.21, i.e. the outer modulation is equal to zero. It follows that for mode coupling, the limit and critical limit of stability is given by,

$$\begin{aligned} a_{limit} &= -\frac{1}{K_s Re[G(\omega_{lim})]} \\ \text{where} \quad \omega_{lim} &\text{ is such that } Im[G(\omega_{lim})] = 0. \end{aligned} \tag{2.26}$$

In most two degree of freedom systems, $Re[G(\omega_{lim})] \simeq Re[G(\omega)]_{min}$. Thus the critical limit of stability for wave regeneration is about half of that for mode coupling and so in most cases, Equation 2.24 or 2.25 will be used as the limiting depth of cut for chatter.

Equation 2.24 gives the limiting depth of cut for a certain vibration frequency (ω) and consequently, a certain phase shift (ε). The phase shift is geometrically related to the vibration frequency and spindle speed by,

$$\begin{aligned} \frac{\omega}{2\pi n z_t} &= N + \frac{\varepsilon}{2\pi} \\ \text{where } \omega &= \text{vibration frequency} \\ n &= \text{spindle speed} \\ z_t &= \text{number of teeth} \\ N &= \text{largest integer number} \\ &\quad \text{such that } 0 < \varepsilon < 2\pi, \end{aligned} \tag{2.27}$$

where ($z_t = 1$) for turning. With a knowledge of the system structural parameters, stability lobe curves can be generated when Equation 2.27 is taken into consideration. Figure 2.8 shows a typical stability lobe curve. The depth of cut is shown along the ordinate while a variable proportional to the cutting speed is shown on the abscissa, the spindle speed in this case. The curve represents the threshold between stable and

unstable cutting; chatter is predicted above the curve while stable cutting is predicted below the curve.

The horizontal line near the bottom of the diagram marks the critical depth of cut (a_{crit}). It is the lowest depth at which chatter can develop, or equivalently the largest depth at which chatter will never develop. As with all stability lobe curves generated from this basic approach, there are certain spindle speeds or cutting speeds at which stable cutting can be sustained well above the critical depth. At these cutting speeds, the phase shift (ε) is such that the regeneration of waves is minimized. It must be kept in mind that these features of the curve, so clearly defined in Figure 2.8, are blurred in reality due to the factors discussed in the next section.

2.8 Factors Affecting Milling Stability

Stability of the actual milling process is not as clear as the previous analysis shows. Many factors unaccounted for in that analysis combine to make stability predictions difficult.

First, the geometry of the milling process differs from the planing model that is used to derive Equation 2.24. Referring to Figure 2.3, milling is an intermittent cutting process and can therefore, include additional entry and exit dynamics. In addition, the instantaneous chip thickness, cutting force magnitude, cutting force direction, and the direction normal to the cut all change with time. These parameters were all assumed time invariant in deriving the previous limit of stability.

Several common occurrences in practical cutting also affect stability. A phenomenon commonly known as *process damping* has been known to alter stability most drastically. Physically, this damping is a result of interference of the tool flank with the freshly cut surface. Refer to Figure 2.9, where a tool is cutting while vibrating and producing an undulated surface. Due to the changing slope of the cut surface, there will be points in

the tool motion where clearance between the tool flank and the surface disappears. As a result, there are additional forces aside from the typical cutting forces due to the shearing process at the tool tip. Assuming these *rubbing* or *ploughing* forces are proportional to the amount of interference, then they will be partly dependant on the tool vibration velocity between point A and point C.

There are different ways of accounting for flank interference. The imaginary part of Tlusty's [18] dynamic coefficients yield harmonic forces which are proportional to the velocity. Thus the aforementioned process is approximated. In reality, ploughing forces are also strongly related to the cut surface geometry and tool geometry.

Others, such as Sisson and Kegg [13] and Montgomery and Altintas [10], assume ploughing forces are proportional to the contact area in some form. Sisson and Kegg assume,

$$\begin{aligned} F_{plough} &= wlp_o \qquad (2.28) \\ \text{where } w &= \text{contact width} \\ l &= \text{contact length} \\ p_o &= \text{material yield strength.} \end{aligned}$$

Montgomery et al., using friction theory borrowed from Halling [3], assume that the ploughing forces in directions parallel and perpendicular to the instantaneous velocity are proportional to the contact area projected in those respective directions,

$$F_{par} = A_{par}p_o \qquad (2.29)$$

$$F_{pen} = A_{pen}p_o. \qquad (2.30)$$

The third way, used by Wu [22], assumes that ploughing forces in the tangential and radial directions are both proportional to the volume of material displaced by the tool

flank,

$$F_{r_{plough}} = Vp_r \tag{2.31}$$

$$F_{t_{plough}} = Vp_t \tag{2.32}$$

where V = displaced volume

p_r = radial ploughing constant

p_t = tangential ploughing constant.

Since no substantial experimental results have been able to back any of these models, they are all approximate. The use of four constants for dynamic cutting force coefficients is experimentally very time consuming. At the same time, the theoretical basis of this method is too simple to account for such a complex geometric and mechanistic problem. The other three methods, all from similar physical origin, represent a different approach. They are based on geometric considerations as well as experimentally determined ploughing constants. The sum of these two aspects adds validity to these models. In addition, a maximum of two constants need to be determined in these three models, as opposed to four in the case of Tlusty's model.

Another effect which may occur during cutting is when an engaged tooth leaves the cut surface and therefore, the uncut chip thickness reduces to zero. The tool *jumping out of cut* is caused by excessive vibration compared with the instantaneous chip thickness. Since the chip thickness typically begins or ends at zero thickness in milling, the tool can easily jump out of cut.

Tool wear, an inescapable reality in metal removal, also plays an important role in cutting stability. The metal to metal contact between the tool and workpiece surface leads to wear by mechanisms of adhesion, abrasion, and diffusion [1]. A finite tool tip radius or a finite flank wearland alters the geometry of the cutting edge and thus influences the

stability by changing the amount of process damping. It is generally accepted that a small amount of wear increases stability, while excessive wear decreases stability.

Finally, the problem of runout also complicates the prediction of chatter. Runout is common in milling cutters, especially with high speed steel end mills; it is the inequality of radii between the tool rotation center and its various cutting edges. In addition to runout of the cutter, there may be out of roundness in the spindle as well. The runout and spindle out of roundness alter the amplitude and phase in wave regeneration. The importance of runout depends on its relative magnitude to the chip thickness.

2.9 Chatter Avoidance

Due to the detrimental nature of chatter, there is on going research to avoid or suppress this kind of instability. Research can be classified into three areas,

- machine tool design
- cutter design
- on-line manipulation of machining variables.

The analysis in Section 2.7 showed that the limiting depth of cut is critically dependant on the structural dynamics of the system, i.e. $G(\omega)$. Although Equation 2.24 is derived from a simpler model, this dependency still holds true for milling. Analytically, decreasing the magnitude of the negative part of $Re[G(\omega)]$ will increase the stability of the system. Physically, increasing the dynamic stiffness of the machine tool across the frequency spectrum will increase stability.

Milling cutter design is primarily concentrated on the development of nonuniform tooth pitch tools. Properly unevenly spaced teeth are known to increase stability by disturbing the wave regeneration process of chatter. However, as noted by Lin et al. [8],

the vibration reduction is critically dependant on the cutting conditions and workpiece geometry. If either of the two changes during cutting, then the stabilizing influence of the uneven chip load would diminish. Thus optimal performance can only be achieved from cutters with adjustable tooth spacing or with many different cutters.

The third area of research in chatter avoidance is the manipulation of available machining parameters. Changing the spindle speed is currently the most successful method. Optimization of the spindle speed falls into two categories,

- set point changes
- continuously varying changes.

The reason behind making set point changes to the cutting speed has been well documented by Smith [14]. Refer to Figure 2.8, where a typical stability lobe curve is shown. At high spindle speeds or tooth frequencies, the number of waves between subsequent cutting edges is small. Thus the spindle speed can be changed such that it corresponds to a nearby *stability peak.* As a result, the stability at a certain depth of cut is maximized. Smith reports that this method works well when the cutting speed is high enough so that there are only one or two wave modulations between subsequent tooth passage. This strategy, though, is not beneficial at moderate and low cutting speeds. Due to the geometric constraint of Equation 2.27, the stability peaks are not prominent at low cutting speeds. More importantly, process damping becomes a more dominating factor at low speeds. Subsequently, these two effects blur stability peaks.

The use of a continuously changing spindle speed is the strategy investigated in this thesis. Although the concept of variable speed cutting is not new, little work has been concentrated in this area. Less work has been done on variable speed milling, and even less has included experimental work. The most relevant results are published by Lin et

al. [8], and they deal with vibration control of face milling using spindle speed oscillations. Simulations show significant reduction in vibration amplitude, but unfortunately, minimal experimental results are given.

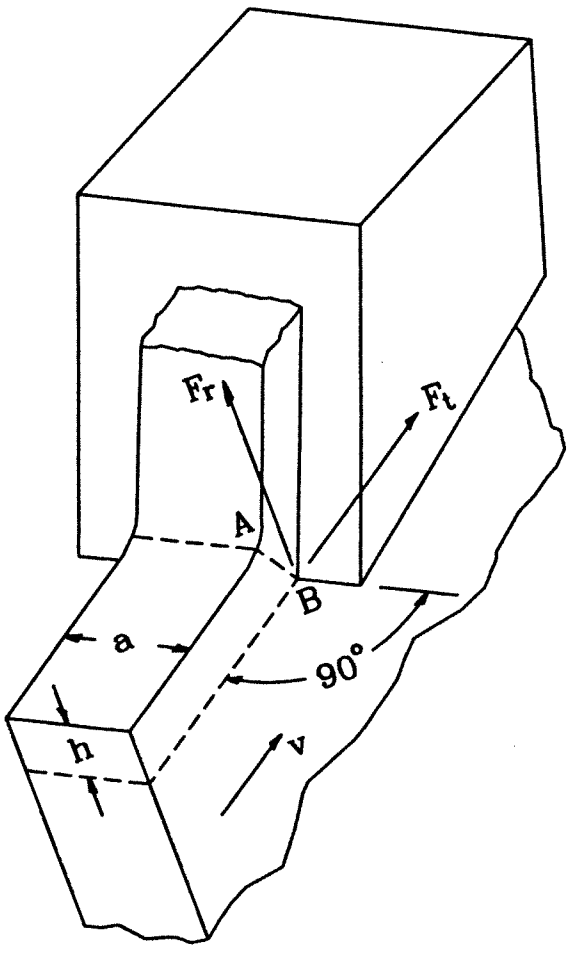


Figure 2.1: Orthogonal cutting process.

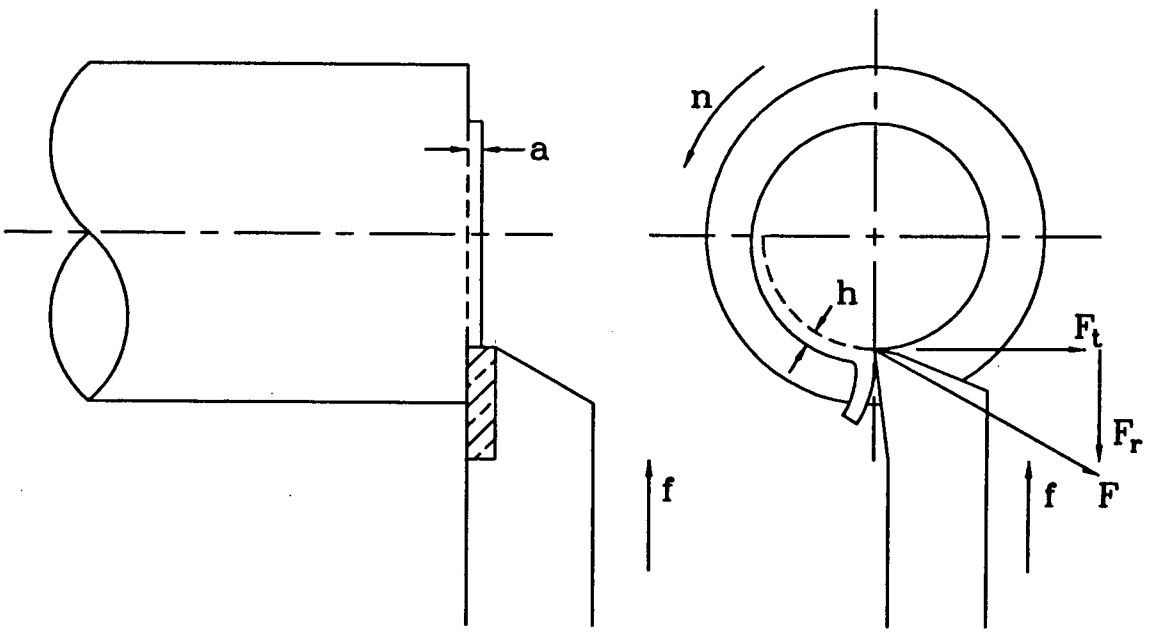


Figure 2.2: Orthogonal turning geometry.

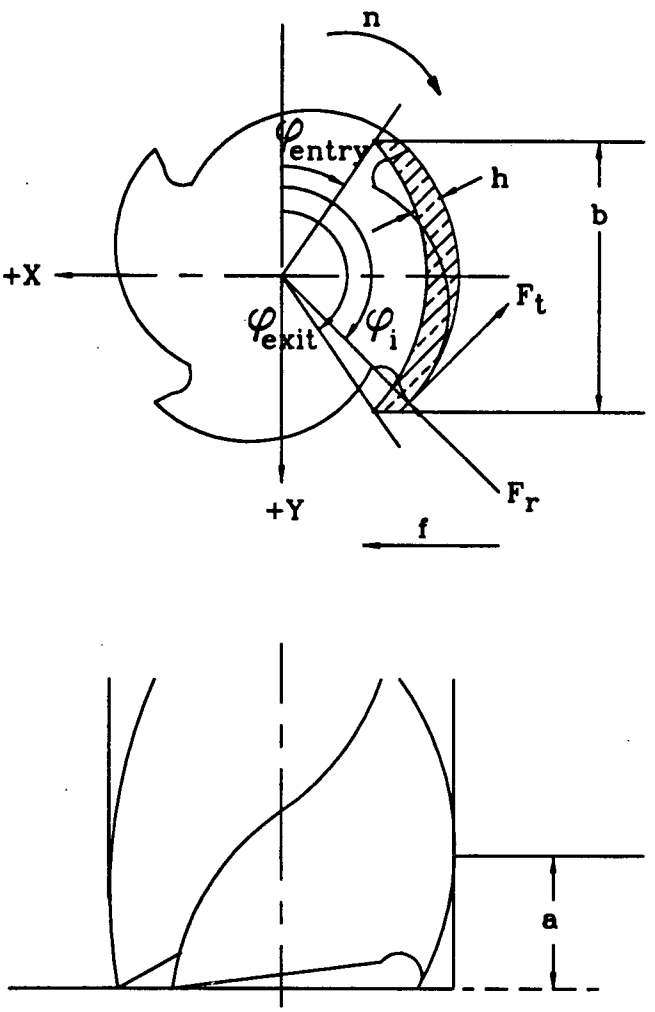


Figure 2.3: End milling geometry.

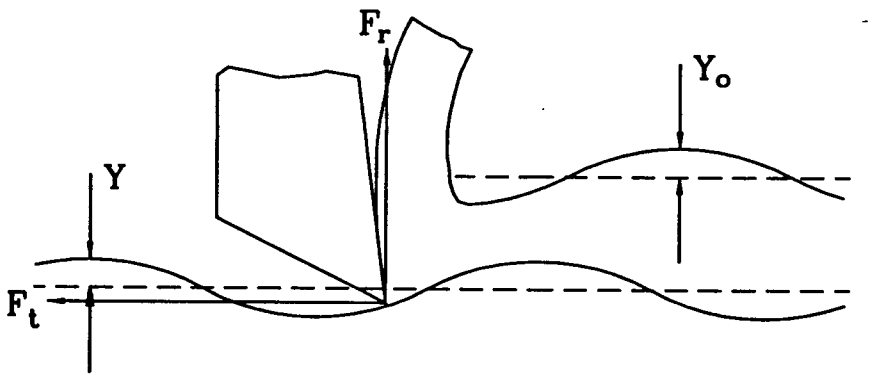


Figure 2.4: Inner and outer modulation of surface.

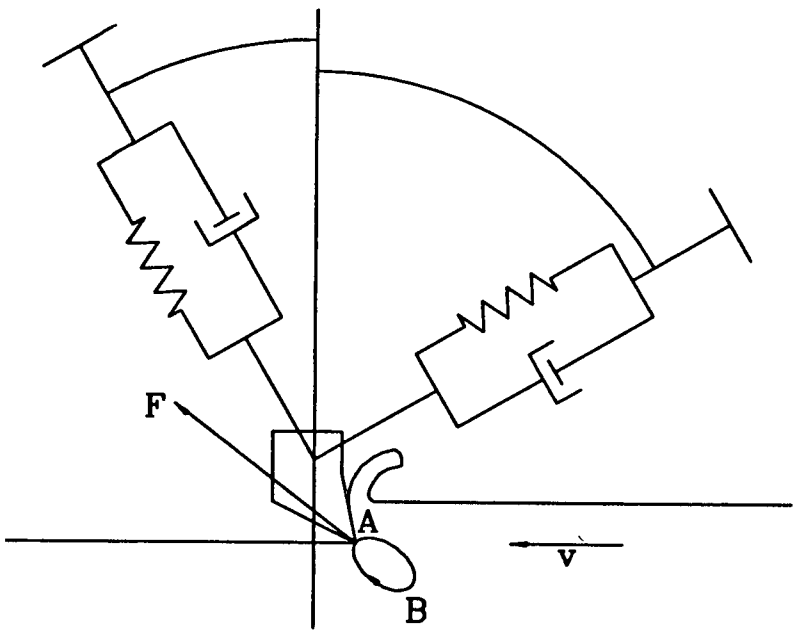


Figure 2.5: Mechanism of mode coupling.

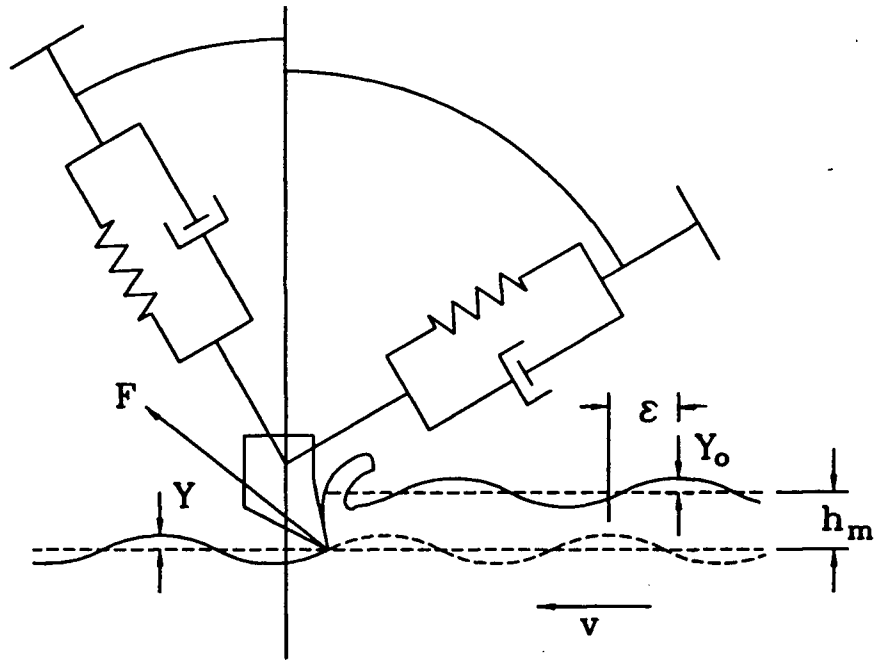


Figure 2.6: Mechanism of wave regeneration.

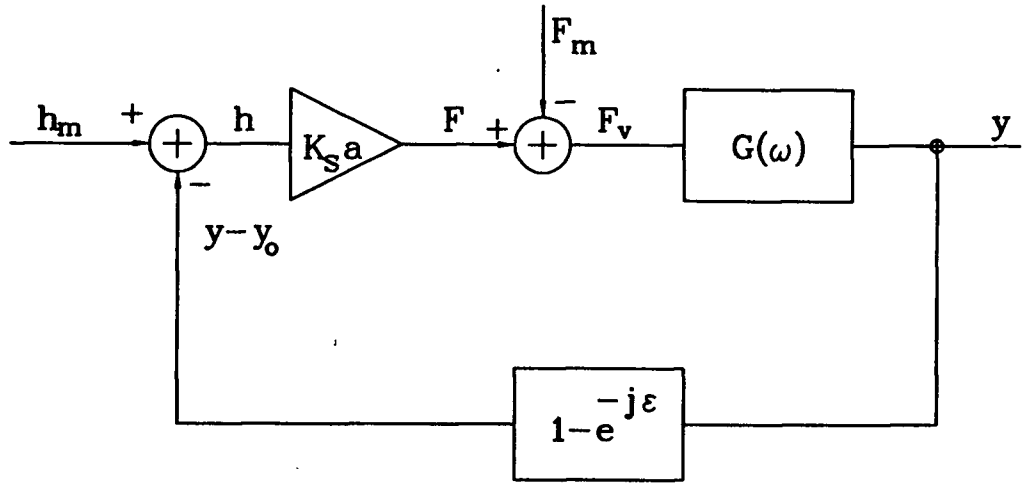


Figure 2.7: Block diagram of wave regneration.

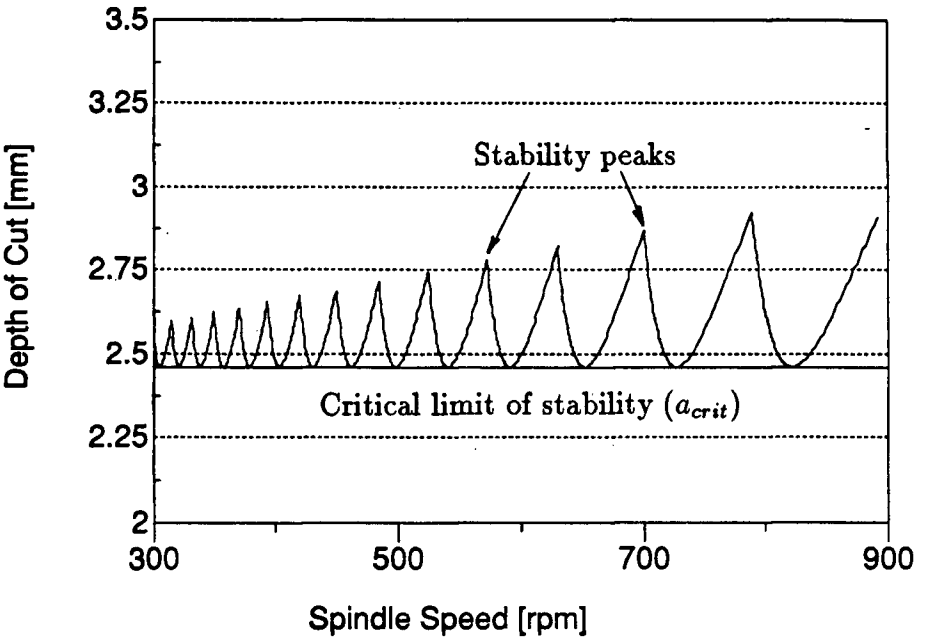


Figure 2.8: Typical stability lobe curve.

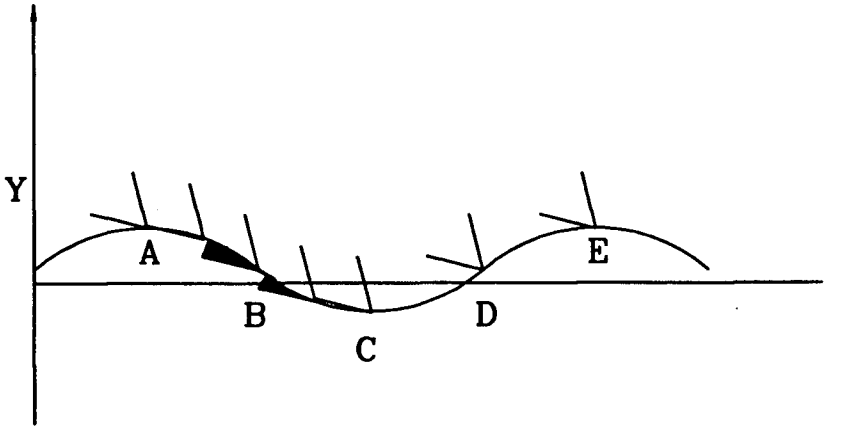


Figure 2.9: Process damping mechanism.

Chapter 3

Dynamic Cutting Models

3.1 Introduction

The various models used to analyze the strategy of variable speed milling are presented in this chapter. A plunge turning time simulation serves to explain the underlying basis of stabilization from speed oscillations, while a milling simulation shows the effectiveness of variable speed cutting in actual milling conditions. Time simulations are necessary because of the analytical difficulties inherent in the variable speed turning and milling problems.

3.2 Algorithm of Milling Simulation

The time domain simulation developed for this research is a flexible program that can model a variety of milling situations. The final objective of the simulation is the evaluation of variable speed milling in terms of increasing stability, and this purpose must be kept in mind. Since stability is critically dependant on the amount of damping during cutting, the phenomenon of ploughing or flank interference must be included in some form. In addition, a simple tool wear model also improves the accuracy of the simulation.

The geometry of milling is once again shown in Figure 3.1. Milling cutter dynamics are approximated by two orthogonal modes (X') and (Y') which are rotated (α) degrees from the feed direction (X) and normal direction (Y), respectively. At a given time, the

cutting force from shearing due to a single tooth in cut (from Equations 2.4 and 2.5) are,

$$F_{t,s}(\phi_i) = K_s a c \sin(\phi_i) \tag{3.33}$$

$$F_{r,s}(\phi_i) = r_1 K_s a c \sin(\phi_i) \tag{3.34}$$

where $F_{t,s}(\phi_i)$ = tangential cutting force of tooth (i) due to shearing

$F_{r,s}(\phi_i)$ = radial cutting force of tooth (i) due to shearing

ϕ_i = immersion angle of tooth (i).

Henceforth, the notation is simplified and the variable for tooth (i) at immersion angle (ϕ_i) will be denoted by,

$$X_y(\phi_i) = X_{y,i}$$

or even more simply as

$$X_{y,z}(\phi_i) = X_{y,z}.$$

Depending on the tool geometry and velocity, there may also be ploughing forces so that the total force on tooth (i) is,

$$F_{t,i} = F_{t,s} + F_{t,p_o} \tag{3.35}$$

$$F_{r,i} = F_{r,s} + F_{r,p_o} \tag{3.36}$$

where F_{t,p_o} = tangential cutting force of tooth (i) due to ploughing

F_{r,p_o} = radial cutting force of tooth (i) due to ploughing.

The evaluation of ploughing forces is described later in this section. The total force on

the cutter is equal to the vectorial sum of the forces due to all teeth in cut,

$$\begin{aligned} \vec{F} &= \sum_{i=1}^{z_t} \vec{F}_{t,i} + \sum_{i=1}^{z_t} \vec{F}_{r,i} \\ \text{where } z_t &= \text{the total number of teeth in cut.} \end{aligned} \tag{3.37}$$

The total force can be more useful if it is decomposed into components in the feed and normal directions, (X) and (Y),

$$F_x = \sum_{i=1}^{z_t} F_{t,i} \cos(\phi_i) + \sum_{i=1}^{z_t} F_{r,i} \sin(\phi_i) \tag{3.38}$$

$$F_y = -\sum_{i=1}^{z_t} F_{t,i} \sin(\phi_i) + \sum_{i=1}^{z_t} F_{r,i} \cos(\phi_i), \tag{3.39}$$

or subsequently into components in the directions of the two uncoupled degrees of freedom, (X') and (Y'),

$$F_{x'} = F_x \cos(\alpha) - F_y \sin(\alpha) \tag{3.40}$$

$$F_{y'} = F_x \sin(\alpha) + F_y \cos(\alpha). \tag{3.41}$$

The equations of motion for the system is,

$$m_{x'}\ddot{x}(t) + c_{x'}\dot{x}(t) + k_{x'}x(t) = F_{x'} \tag{3.42}$$

$$m_{y'}\ddot{y}(t) + c_{y'}\dot{y}(t) + k_{y'}y(t) = F_{y'}, \tag{3.43}$$

where $(m_{x'})$, $(c_{x'})$, and $(k_{x'})$ are the structural parameters for the mode in the (X') direction, and $(m_{y'})$, $(c_{y'})$, and $(k_{y'})$ are the structural parameters for the mode in the (Y') direction. Using a digital integration method such as Euler's approximation, the present tool deflection in the (X') and (Y') directions (x'_t and y'_t) can be found from the previous deflections (x'_{t-1} and y'_{t-1}), their first and second derivatives, and the structural parameters,

$$\ddot{x}'_t = (F_{x'} - c_{x'}\dot{x}'_{t-1} - k_{x'}x'_{t-1})/m_{x'}$$

$$\begin{aligned}
\dot{x}'_t &= \dot{x}'_{t-1} + \ddot{x}'_t dt \\
x'_t &= x'_{t-1} + \dot{x}'_t dt \\
\text{and,} & \\
\ddot{y}'_t &= (F_{y'} - c_{y'}\dot{y}'_{t-1} - k_{y'}y'_{t-1})/m_{y'} \\
\dot{y}'_t &= \dot{y}'_{t-1} + \ddot{y}'_t dt \\
y'_t &= y'_{t-1} + \dot{y}'_t dt.
\end{aligned} \tag{3.44}$$

The deflections are used to determine the uncut chip thickness for each tooth in cut (in the direction normal to the cut surface, i.e. the radial direction (Z) in Figure 3.1). After transforming the deflections from the modal directions (x' and y') to the feed and normal directions (x and y), the radial deflection is given by,

$$z_i = x\sin(\phi_i) + y\cos(\phi_i).$$

The chip thickness for tooth (i) is equal to the feed per tooth at immersion (ϕ_i) plus the previous minimum radial deflection ($z_{min,i}$) minus the present radial deflection all at immersion angle (ϕ_i),

$$h(\phi_i) = \begin{cases} c\sin(\phi_i) - z_i + z_{min,i} & \text{if } c\sin(\phi_i) - z_i < z_{min,i} \\ 0 & \text{if } c\sin(\phi_i) - z_i > z_{min,i}. \end{cases} \tag{3.45}$$

The minimum values of the surface deflection (z_{min}) at different immersions are kept in an array where they can be updated with each tooth pass as necessary. When the present deflection subtracted from the feed per tooth is less than the previous minimum deflection (z_{min}), there is a positive chip thickness (see Figure 3.2(a)), otherwise the tool has jumped out of cut and the uncut chip thickness is equal to zero (see Figure 3.2(b)), leading to zero cutting forces. Once the uncut chip thickness is determined, its value is used to determine the instantaneous cutting forces, and the algorithm begins again at the next immersion angle.

The above paragraphs described the basis of the simulation which is in essence similar to the algorithm presented by Smith [15] and others [4]. However, as mentioned earlier in the section, this model includes many qualities important in altering the stability characteristics of dynamic milling, such as:

- runout
- variable spindle speed
- flank interference
- tool wear.

Runout from the end mill can be thought of as additional constant *negative* deflections associated with each tooth. Therefore, the uncut chip thickness equation from 3.45 is replaced by,

$$h(\phi_i) = \begin{cases} c\sin(\phi_i) - z_{e,i} + z_{min,i} & \text{if } c\sin(\phi_i) - z_{e,i} < z_{min,i} \\ 0 & \text{if } c\sin(\phi_i) - z_{e,i} > z_{min,i} \end{cases} \tag{3.46}$$

$$\text{where } z_{e,i} = z_i - r_{o,i}$$

$$= \text{effective vibration}$$

$$r_{o,i} = \text{runout for tooth } (i).$$

The inclusion of runout is useful for correlation with experimental findings.

A sinusoidal spindle speed variation imposed on top of the nominal speed (n_o) is easily incorporated into the simulation by choosing two user inputs, the amplitude of speed variation (dn) and the frequency of speed variation (n_f). The spindle speed as a function of time is thus given by,

$$n = n_o + dn\sin(2\pi n_f t). \tag{3.47}$$

Since the immersion angle increment ($d\phi$) is fixed in order to properly keep track of previous surface modulations, continuous speed changes are achieved by changing the time increment (dt) between iterations, where it is related to the spindle speed by,

$$dt = \frac{d\phi}{2\pi n} = \frac{d\phi}{2\pi(n_o + dn\sin(2\pi n_f t))}. \tag{3.48}$$

The most important addition to the simulation is the option to include the mechanism of flank interference. A model based on friction theory, which was used by Montgomery et al. [10] for metal cutting, provides the process damping necessary for proper stability analysis, especially at low cutting speeds. Figure 3.3(a) shows the ploughing geometry for an infinitely sharp tool where the workpiece surface is assumed flat in the tangential direction. The necessary input parameters are the clearance angle (γ) and the length of the primary flank face (l_{fl}). When the angle of the instantaneous tooth velocity (γ_e) is larger than the clearance angle, then ploughing forces will come into effect. The contact areas parallel and perpendicular to the velocity are given by,

$$A_{par} = a l_{pl}\cos(\gamma_e - \gamma) \tag{3.49}$$

$$A_{pen} = a l_{pl}\sin(\gamma_e - \gamma) \tag{3.50}$$

$$\text{where } l_{pl} = \text{minimum of } (l_{fl}, \frac{h}{\sin(\gamma)}).$$

The length of contact along the flank face is assumed to be limited to (l_{fl}), since the clearance angle typically increases sharply along the secondary flank face. As mentioned before, the ploughing forces are proportional to the material yield strength,

$$F_{par} = A_{par} p_o \tag{3.51}$$

$$F_{pen} = A_{pen} p_o. \tag{3.52}$$

Thus the ploughing forces in the tangential and radial directions in Equations 3.35 and 3.36 are finally given by,

$$F_{t,po} = F_{par}\sin(\gamma_e) + F_{pen}\cos(\gamma_e) \tag{3.53}$$

$$F_{r,po} = F_{par}\cos(\gamma_e) - F_{pen}\sin(\gamma_e). \tag{3.54}$$

Tool wear is a complicated problem; typical wear results in a finite radius tool tip, a flank wear flat, and crater wear on the rake face. For the case where there is a flank wear flat, as shown in Figure 3.3(b), the steps in determining interference forces are similar. Due to the zero clearance along the flank wearland ($\overline{VB}$), ploughing is present whenever the tooth velocity direction is into the workpiece, so that Equations 3.49 and 3.50 are replaced by,

$$A_{par} = \begin{cases} a(\overline{VB}\cos(\gamma)) & \text{if } 0 < \gamma_e < \gamma \\ a(l_{pl}\cos(\gamma_e - \gamma) + \overline{VB}\cos(\gamma)) & \text{if } \gamma_e > \gamma \end{cases} \tag{3.55}$$

$$A_{per} = \begin{cases} a(\overline{VB}\sin(\gamma)) & \text{if } 0 < \gamma_e < \gamma \\ a(l_{pl}\sin(\gamma_e - \gamma) + \overline{VB}\sin(\gamma)) & \text{if } \gamma_e > \gamma \end{cases} \tag{3.56}$$

where $\overline{VB}$ = flank wearland length.

When the instantaneous tooth velocity angle is positive (i.e. into the workpiece), interference of the flank wearland occurs. If the angle is greater than the clearance angle, rubbing of both the wear flat and flank face is assumed. Since the limiting velocity angle necessary for interference is zero for this model as opposed to the clearance angle for an infinitely sharp tool, a tool with wear always results in more ploughing.

The high ploughing stiffness introduces the problem of large physically improbable impact forces when the tool starts to plough. With a digitization frequency which is just large enough to track the vibrations of the tool (i.e. approximately two times the tool natural frequency according to the Nyquist criterion), the ploughing forces initially imposed on the tool are often erroneously large. To avoid this problem, the size of the time step (or equivalently, the angle step) must be decreased, perhaps by as much as two orders of magnitude. The necessary computation time, unfortunately, also increases by

two orders of magnitude. As with any type of computer simulation, the efficiency and runtime of the program must be a major factor of concern. In this case, a digitization frequency of several hundred times the tool natural frequency makes the simulation runtime unacceptably long. Improvement to this situation lies in the fact that the cutter only ploughs a fraction of the time, in fact, as little as 5% if there is no flank wear flat. A simple solution is to use two values of angle increments where the small increment comes into effect when flank interference occurs. In actual implementation, once ploughing is detected, the algorithm repeats the original time step with a series of very small steps as shown in Figure 3.4. Since the main purpose of especially small time steps is to avoid large impact ploughing forces and not to produce a surface profile of the workpiece, the radial deflection produced by these small steps are not recorded. Only the radial vibrations at nominal immersion steps are updated in the surface deflection array.

3.3 Turning Simulation

This section regresses from the previous section by outlining a turning simulation. A time domain simulation of turning is necessary for the analysis of variable speed cutting in the succeeding sections. The algorithm is almost entirely duplicated from other researchers [6], and is presented here for the sake of clarity. The turning simulation is in essence a simplified version of the milling simulation.

The model used for turning is shown in Figure 3.5. A two degree of freedom system with orthogonal modes (X' and Y') approximates the dynamics between the tool and the workpiece. Orthogonal modes are rotated by (α) degrees from the feed (X) and normal (Y) directions. The resultant cutting force is in the direction (β) degrees from the normal. The basic digital step in the algorithm is the rotation angle step ($d\phi$), where

there are (m) steps per revolution, or

$$d\phi = \frac{2\pi}{m}.$$

The cutting force at a step (i) and rotation angle (ϕ) is given by,

$$F_i = \begin{cases} K_s a h & \text{if } h > 0 \\ 0 & \text{if } h \leq 0. \end{cases} \tag{3.57}$$

The uncut chip thickness, as in the milling case, depends on the present normal vibration (z_i) as well as the minimum surface left by previous passes at that rotation angle $(z_{\phi,min})$,

$$h = h_m + z_{\phi,min} - z_i.$$

Components of the cutting force that excite the orthogonal modes are given by,

$$F_{x'} = F_i \cos(\frac{\pi}{2} + \alpha - \beta) \tag{3.58}$$

$$F_{y'} = F_i \cos(\beta - \alpha). \tag{3.59}$$

Digital integration is then used to obtain the vibration of the two modes (x' and y'), and the vibration in the normal direction which modulates the chip thickness is given by,

$$z_i = y' \cos(\alpha) - x' \sin(\alpha).$$

Once the normal vibration is determined, the uncut chip thickness is computed. Consequently, the angle step can be incremented, and the algorithm starts once again.

The digital turning simulation is a simple program that accurately models the system shown in Figure 3.5, which is also the model used for stability lobe generation. Thus simulation results can be used to study variable speed cutting as well as to compare with stability lobe diagrams.

3.4 Variable Speed Cutting

The theory of variable speed cutting is best analyzed in terms of the turning process. Milling adds complicated geometric considerations which are unnecessary for the proper analysis of the ways variable speed stabilize cutting. Before discussing speed oscillations, it is important to get a better understanding of fixed speed cutting.

3.4.1 Fixed Speed Cutting

The model presented in Section 2.7 together with the geometric constraint of Equation 2.27 represents a fixed speed plunging operation in turning. The important stability equations from Section 2.7 were,

$$\begin{aligned} a_{limit} &= -\frac{1}{2K_s Re[G(\omega)]} \\ \frac{\omega}{2\pi n} &= N + \frac{\varepsilon}{2\pi}. \end{aligned}$$

These equations predict the stability of the cut given the system transfer function and the spindle speed. Additional information of chatter frequency (ω) and phase shift (ε) can also be found.

At first, it may seem that there are too many unknowns to be determined, but there is an additional relationship that comes into play, relating the phase shift with the transfer function. To fully understand this relationship, it is necessary to derive Equation 2.24 in some detail as Tlusty [6] has. The substitution of Equation 2.20 into Equation 2.22results in,

$$y = F_v G(\omega) = K_s a(y_o - y)G(\omega).$$

However, at the limit of stability ($Y_o = Y$) and,

$$y_o = ye^{-j\varepsilon},$$

and that leads to

$$y = K_s a(ye^{-j\varepsilon} - y)G(\omega),$$

or alternately,

$$y = -K_s a_{limit} y(1 - e^{-j\varepsilon})G(\omega).$$

Thus the limiting depth of cut is given by,

$$a_{limit} = -\frac{1}{K_s G(\omega)(1 - e^{-j\varepsilon})}.$$

Since the depth of cut is always a real number, and the specific cutting pressure is assumed real in this case, the difference,

$$G(\omega)(1 - e^{-j\varepsilon}),$$

must be real as well. As $(e^{-j\varepsilon})$ is an unit vector, the phase plane plot at the limit of stability is shown in Figure 3.6. Therefore,

$$K_s G(\omega)(1 - e^{-j\varepsilon}) = 2K_s Re[G(\omega)],$$

and the final form of Equation 2.24 is apparent. There is also a direct relationship between $G(\omega)$ and the phase shift (ε). From Figure 3.6, the phase shift as a function of $G(\omega)$ is

$$\varepsilon = 2\pi - 2tan^{-1}(\frac{Re[G(\omega)]}{Im[G(\omega)]}). \tag{3.60}$$

Using Equations 2.24, 2.27, and 3.60, a particular value of chatter frequency, chatter limit, and phase shift can be found for a certain spindle speed. Although the relationship is transcendental, simple computer iterations are sufficient to produce stability lobe curves such as the one in Figure 3.7. Figure 3.8 shows the additional variables of chatter frequency, integer number of surface waves, and phase shift for the dynamic system in Figure 2.6 which has the cutting and structural parameters as listed in Table 3.1. The

peaks of the limiting cutting depth curve correspond to the transition of the number of wave modulations from N_o to $N_o + 1$ in Equation 2.27. For a certain wave, the chatter frequency tends to slowly increase with speed, while the phase shift decreases correspondingly. For this case, the chatter limit decreases monotonically until the critical depth of cut (a_{crit}), and then increase until the intersection of the curve for the next wave, producing stability peaks. It is important to emphasize that stability at a certain depth of cut changes with spindle speed. Thus the process can move from chatter to stability with just a small shift in speed. The proper speed adjustment alters the phase shift so that wave regeneration activity is reduced.

The optimal value of the phase shift is zero degrees, which results in no dynamic uncut chip thickness variation. However, the phase shift (ε) between the inner and outer modulations at the limit of stability always lies between 180 and 360 degrees. This follows from the restriction on the chatter frequency during cutting. Refer to Figure 3.9 which shows the relationship between the cutting force (F), the present vibration (y), and the previous vibration (y_o). The variable component of the uncut chip thickness ($y_o - y$) is parallel with the real axis because it is in phase with the force (F). The phase angle between the present vibration and the vector ($y_o - y$) is denoted as (ψ). It is clear from the diagram that the angle (ψ) can take any value between -90 and -270 degrees,

$$-270^0 < \psi < -90^0. \tag{3.61}$$

The vibration (y), however, also lags the force (F) by the angle (φ). The value for (φ) for a single degree of freedom system is limited to between 0 and -180 degrees,

$$-180^0 < \varphi < 0^0. \tag{3.62}$$

The phase shift is zero for static loads, and approaches -180 degrees for high frequency excitations. Since the force (F) is in phase with the uncut chip thickness, it is in phase

with $(y_o - y)$; in order for this relationship to be true, the two angles (ψ) and (φ) must be equal, as it is shown in the diagram,

$$\psi = \varphi. \tag{3.63}$$

The angles (ψ) and (φ) may, therefore, only take a value which satisfies both Equation 3.61 and Equation 3.62, or

$$-180^0 < \psi = \varphi < -90^0. \tag{3.64}$$

Therefore, the chatter frequency is always greater than the natural frequency of the structural system since the phase lag between the vibration and force has to be greater than 90 degrees. The phase shift between the present and previous vibrations (form Figure 3.9) is related to (φ) by,

$$\varepsilon = 360^0 + 2(90^0 + \varphi). \tag{3.65}$$

Since the value of (φ) satisfies Equation 3.64, the value of (ε) is limited to between 180 and 360 degrees,

$$180^0 < \varepsilon < 360^0. \tag{3.66}$$

From this derivation, it is clear that there is always a phase shift between the present and previous vibrations. The phase shift approaches 360 degrees (or equivalently, zero degrees) as the vibration frequency approaches the natural frequency of the system. This is the main reason behind chatter prevention when the tooth frequency coincides with the natural frequency of the cutting system [14].

The results of Figure 3.8 imply that a vibratory system with real receptance $(G(\omega))$ cutting at speed (n) will converge on a particular value of phase shift (ε) and chatter frequency (ω). Time histories of these two variables from time domain simulations show this fact more convincingly. Plunging simulation results of the above system with a

spindle speed of 600 [rpm] and a depth of cut of 2.7 [mm] are shown next. First, Figure 3.10 shows the time history of the normal vibration. It is apparent that the maximum amplitude continues to increase after about four revolutions (t = 0.4 [sec]). Corresponding time histories of phase shift (computed after 0.2 [sec]) and vibration frequency (computed after 0.1 [sec]) in Figure 3.11 show that both variables oscillate transiently for a short time before settling to fixed values. Apparent oscillations in the steady-state values are a result of numerical round off error and not due to system instability. Wave regeneration works to destabilize the vibrations after the first revolution, but the process becomes more prominent once the phase shift reaches a steady-state value. The overall stability of the system depends on the cutting depth relative to the limiting depth of cut. If the depth of cut is greater than the limit for the given spindle speed, then there is enough energy to increase the vibration with each passing revolution. On the other hand, if the limit is not exceeded, wave regeneration does not impart sufficient energy to sustain vibrations, despite the fact that the phase shift settles to a fixed value. Figure 3.12 and Figure 3.13 illustrate this when the depth of cut is reduced to 2.5 [mm] and the speed is adjusted to 636 [rpm].

3.4.2 Stability of Variable Speed Cutting

Due to difficulties of the problem, there have been few attempts to investigate variable speed cutting in a theoretical manner. Jemielniak and Widota [5], however, have presented a method of analysis based on the influence of the chatter frequency. Their analysis serves as a good starting point for discussion and is summarized below.

For simplicity, the governing equation for a single degree of freedom system is used,

$$m\ddot{y}(t) + c\dot{y}(t) + ky(t) = F(t).$$

The cutting force (as explained in Section 2.7) is given by,

$$F(t) = K_s a[h_m - y(t) + y(t-\tau)],$$

where (h_m) is the mean chip thickness and (τ) is the time delay between tooth passes, or the tooth period. In the case of turning, the tooth period is equivalent to the spindle period. Thus the dynamic system becomes,

$$m\ddot{y}(t) + c\dot{y}(t) + ky(t) = K_s a(h_m - y(t) + y(t-\tau))$$

or equivalently,

$$m\ddot{y}(t) + c\dot{y}(t) + Ky(t) = K_y(h_m + y(t-\tau)), \tag{3.67}$$

where the stiffness of the system $(K = k + K_s a)$ comprises of the structural stiffness (k) and the cutting stiffness $(K_y = K_s a)$. The static component of the force $(K_y h_m)$ is ignored, and Equation 3.67 is considered as a forced vibration problem with forcing function $(K_y y(t-\tau))$. Assuming vibrations are always sinusoidal in shape, the amplitude magnification factor (λ) between the present vibration (Y_t) and the vibration one revolution previous (Y_τ) is given by,

$$\lambda = \frac{Y_t}{Y_\tau} = \frac{K_y/K}{\sqrt{[1-(\frac{\omega}{\omega_o})^2]^2 + 4\zeta^2(\frac{\omega}{\omega_o})^2}}. \tag{3.68}$$

The natural frequency of the dynamic system is $(\omega_o = \sqrt{\frac{K}{m}})$, while the chatter frequency is (ω). The dynamic system is unstable if $(\lambda > 1)$ and stable if $(\lambda < 1)$. The amplitude magnification factor as a function of frequency for three depths of cut is shown in Figure 3.14.

This analysis is based on the amplitude magnification factor's dependency on the chatter frequency. The stability of the system for a given depth of cut depends on the frequency it is vibrating at. Referring to Figure 3.14 for depth (a_3), vibrations are

unstable for $\omega_a < \omega < \omega_b$. In turn, stability is also related to the spindle speed since Jemielniak and Widota assume speed and chatter frequency are directly proportional,

$$\omega_t = (\frac{n_t}{n_\tau})\omega_\tau. \tag{3.69}$$

The variables (ω_t and n_t) are the present frequency and speed, and (ω_τ and n_τ) are the frequency and speed during the previous revolution. The frequency of self-excited vibration in fixed speed turning is assumed to be always equal to the natural frequency of the system (ω_o), nearly the most unstable frequency. Thus a change in spindle speed, producing a proportional change in the chatter frequency, always decreases the value of (λ). However, following this change, the system tends to adjust its vibration such that the new vibration frequency approximates (ω_o) again, where the magnification factor is larger than one. Thus continuous spindle speed oscillation is necessary to prevent the system from developing a new, unstable vibration. The chatter frequency for succeeding revolutions (M) is given by,

$$\omega_1 = \frac{n_1}{n_0}\omega_0,\ \omega_2 = \frac{n_2}{n_1}\omega_1 = \frac{n_2}{n_0}\omega_0,\ \cdots \omega_M = \frac{n_M}{n_{M-1}}\omega_{M-1} = \frac{n_M}{n_0}\omega_0.$$

From Equation 3.68, each frequency corresponds to a particular amplitude magnification factor (λ). It follows that the vibration magnitudes of succeeding revolutions are given by,

$$\begin{aligned} Y_1 &= Y_0\lambda_1(\omega_1) \\ Y_2 &= Y_1\lambda_2(\omega_2) = Y_0\lambda_1(\omega_1)\lambda_2(\omega_2) \\ &\vdots \\ Y_M &= Y_{M-1}\lambda_M(\omega_M) = Y_0\lambda_1(\omega_1)\cdots\lambda_M(\omega_M). \end{aligned} \tag{3.70}$$

In order to achieve permanent stability the total amplitude ratio at any revolution (M) must be less than one,

$$\frac{Y_M}{Y_0} = \lambda_1(\omega_1)\cdots\lambda_M(\omega_M) < 1.$$

The amplitude ratio at each spindle revolution can be computed from Equations 3.68, 3.69, and 3.70 for speed oscillation governed by $n = n_o + dn \sin(2\pi n_f t)$. At the nominal spindle speed (n_o), the vibration frequency is assumed to be the natural frequency of the system (ω_o). The computation of (Y_M/Y_o) is graphically shown in Figure 3.15. The left curve plots the amplitude magnification factor as a function of the frequency. To the right is the imposed spindle speed variation as a function of time. The resulting amplitude ratios are shown below the speed curve. Two oscillations are shown, one for ($dn = 6\%n_o$) and (n_f =6% n_o) and the other for ($dn = 10\%n_o$) and ($n_f = 6\%n_o$). In the case of 6% variation amplitude, speed oscillations are unable to stabilize the system and the amplitude ratio increases to approximately 1.6 after 10 revolutions. When the variation amplitude is increased to 10%, amplitude ratios decrease and stability is achieved.

Jemielniak and Widota point out a couple of important observations in their analysis. The logic behind these observations are correct, though there are drawbacks in their execution. The first point is the link between spindle speed and chatter frequency with stability during steady-state cutting. They postulate that there is a proportional relationship between the speed and frequency which leads to stability being a function of spindle speed. Thus changes in speed can directly alter the amplitude magnification factor (λ). Only the final statement is correct. The relationship they used between the speed and frequency (Equation 3.69) and the frequency and magnification factor (Equation 3.68) are erroneous. Analyses by Tlusty and Ismail have established that these two relationships are more complex since they are based on three interrelated equations (Equations 2.24, 2.27, and 3.60). Theoretically, steady-state stability depends on the negative real receptance (Equation 2.24) rather than the magnitude of the receptance of the dynamic system (Equation 3.68). Nevertheless, it is valuable to note that a change in the speed can increase stability.

The second important, but inaccurate observation is the affinity of the dynamic system to vibrate at the natural frequency unless speed is varied continuously. The inaccurate aspect is the assumption that the steady-state vibration frequency is the natural frequency, regardless of the speed. As just mentioned in the previous section, the frequency of vibration is governed by the dynamic system as well as the speed, and in fact is practically always different from the natural frequency. The important aspect is the attraction of the dynamic system to a particular chatter frequency and associated magnification factor, and hence the necessity to vary the spindle speed continuously for maximum stability.

Finally, the assumption Jemielniak and Widota employ to solve the equation of motion has to be commented on. The equation is considered as a forced vibration problem, where the forcing results from the waviness of the workpiece left by the previous revolution. The solution is thus exact only for steady-state analysis, when the forcing frequency is fixed and is therefore, equal to the vibration frequency. When the spindle speed changes continuously, the *forcing frequency* and chatter frequency also change continuously. In this way, the solution is only accurate when speed variations are slow enough such that the forcing frequency always approximates the chatter frequency. Transient effects dominate when variations are fast.

Another analysis of chatter suppression from spindle speed variation was published by Takemura et al. [17]. On the surface, their approach is dissimilar to the analysis of Jemielniak and Widota. Takemura uses an energy balance method to determine stability. Stable cutting is achieved when the energy supplied to vibration is less than the energy dissipated by the vibration. He explains that for a certain cutting depth where there is chatter, there are also stable cuts that can take place at different spindle speeds and vibration frequencies. He gives no relationships between stability, speed, and frequency, but maintains that it is not possible to fix to a certain energy state with just speed

control. Thus it is necessary to vary the spindle speed continuously so that over a given time period, more cutting occurs in the stable region than in the unstable region. In reality, the basis of this analysis is similar to the one presented by Jemielniak and Widota. Both rely on the fact that by changing speed continuously, it is possible to spend more time in stable cutting than unstable cutting. Neither explanations, however, account for the transient nature of speed oscillations.

To better understand the problem, it is worthwhile to reexamine the equation of motion. As discussed above, the equation of motion governing the cutting process of a single degree of freedom machine tool is given by,

$$m\ddot{y}(t) + c\dot{y}(t) + (k + K_s a)y(t) = K_s a(h_m + y(t - \tau)). \tag{3.71}$$

Solving Equation 3.71 in the frequency domain for fixed speeds (as done earlier in Section 3.4.1) gives a particular set of frequency and phase shift for a certain spindle speed and depth of cut. That solution only applies when the spindle speed is fixed. When the spindle speed is the sum of a nominal value and a sinusoidal oscillation, the time delay is given by,

$$\tau = \frac{1}{n_o + dn\sin(2\pi n_f t)},$$

so that the time delay in Equation 3.71 is a function of time. In addition, the mean chip thickness is also time variant since it is also a function of speed,

$$h_m = \frac{f}{n_o + dn\sin(2\pi n_f t)}.$$

Consequently, the frequency domain analysis of Equation 3.71 is no longer valid. The new equation that must be solved is,

$$\begin{aligned} m\ddot{y}(t) + c\dot{y}(t) \quad &+ \quad (k + K_s a)y(t) = K_s a(h_m(t) + y(t - \tau(t))) \\ \text{where } \tau(t) \quad &= \quad \frac{1}{n_o + dn\sin(2\pi n_f t)} \end{aligned} \tag{3.72}$$

$$h_m(t) = \frac{f}{n_o + dn\sin(2\pi n_f t)}$$

and unfortunately, a simple solution does not exist. The best alternative is to use digital simulations to investigate the influence of variable cutting speed.

The next several plots illustrate the influence of speed oscillations on the vibration, frequency, and phase shift. In the case described below, variable speed is initiated after the twelfth spindle revolution (at $t = 1.2$ [sec]). The three plots (in Figures 3.16 and 3.17), of the normal vibration, phase shift, and chatter frequency, are for a nominal speed of 600 [rpm], depth of 2.7 [mm], variation amplitude of 50 [rpm], and variation frequency of 5 [Hz]. Before the twelfth revolution, all three plots are the same as the corresponding fixed speed plots (Figures 3.10 and 3.11). While the chatter frequency and phase shift converge to fixed values, the vibration increases in magnitude. Once variable speed begins, the vibration magnitude starts to descend and the other two variables fluctuate continuously, not settling to steady values. Speed oscillation improves stability because the resulting fluctuation of vibration frequency and phase shift prohibits the full mechanism of wave regeneration to occur. The system is not allowed to settle to the states of greatest instability, which is what happens during fixed speed cutting. A discussion of the factors affecting the stabilizing ability of variable spindle speed will be presented later, but the next three figures emphasize the fact that stabilization is brought upon by phase shift fluctuation.

Simulations show that the amount of reduction in vibration is directly reflected in the distribution of the phase shift over a fixed time period. A large increase in stability relates to a widely spread phase shift distribution. Figure 3.18 shows the normalized phase shift distribution before and after speed oscillation begins for the case illustrated in Figure 3.17(a). As expected in fixed speed cutting, the phase shift is narrowly distributed, with approximately 90% of the phase shift lying between 225 and 260 degrees. On the other

hand, the phase shift for variable speed cutting is significantly more evenly distributed. A measure of the wideness of the phase shift distribution is the variance. The variance of the phase shift is given by,

$$s^2(\varepsilon) = \frac{\sum_{i=1}^{P}(\varepsilon_i - \bar{\varepsilon})^2}{P-1},$$

where $(\bar{\varepsilon})$ is the mean value of the phase shift, and (ε_i) represents the phase shifts from a set of (P) samples. For example, the variance of the fixed speed phase distribution in Figure 3.18 is approximately 10 degrees, while the variance of the variable speed distribution is approximately 50 degrees. The next two plots show the trend of decreasing vibration amplitude with increasing phase shift variance. Figure 3.19(a) shows the maximum amplitude and the variance for different values of variation frequency when the nominal speed is 600 [rpm] and the variation amplitude is 60 [rpm]. Conversely, Figure 3.19(b) has the same two variables with variation amplitude changing and the variation frequency fixed at 6 [Hz].

Time simulations also give some indications of how the speed variation parameters (dn and n_f) influence stability. Although the plunging model does not realistically represent milling, its relative simplicity and fast simulation time prove to be convenient in the determination of trends in variable speed cutting. Figure 3.20 illustrates the maximum vibration amplitude during the 15th revolution for various frequencies and amplitudes of variation for nominal speeds of 500 [rpm] and 600 [rpm]. Only general trends can be observed from these plots. Large values of variation amplitude are always more beneficial than small values. However, increases in variation amplitude yield diminishing returns. For example, the increase in stability when variation amplitude changes from 10 [rpm] to 20 [rpm] is usually more than when the variation amplitude changes from 20 [rpm] to 30 [rpm]. The vibration amplitude also decreases with increasing variation frequency. This trend continues until the variation frequency approaches the nominal spindle speed

(8.33 [Hz] for Figure 3.20(a) and 10.0 [Hz] for Figure 3.20(b)).

3.4.3 Discussion of Variable Speed Cutting

Variable speed oscillation stabilizes the cutting process in two ways : 1) oscillation causes the process to move to more stable speeds such that over a period of time, the average stability covered by the range of speed is more stable than the original speed, and 2) resulting transient fluctuation of the phase shift between inner and outer modulation prevents the full wave regeneration process to develop. Having illustrated these observations in the previous sections, a discussion of the determinants of variable speed cutting success is presented next.

First, the stability lobe diagram must be reexamined here since it forms an integral part of this discussion. The stability lobe curve represents all points on the diagram where the amplitude magnification factor is equal to one, i.e the points of critical stability. Every point on the line, and in fact every point on the diagram, has a chatter frequency, phase shift, and magnification factor associated with it. In other words, for a given depth of cut and speed, the steady-state dynamic process is characterized by a *state* which can be described by the chatter frequency, phase shift, and most importantly amplitude magnification factor. This state represents the most unstable combination of frequency and phase shift pertaining to wave regeneration for the given speed and depth of cut. The overall stability of the process depends on the resulting steady-state amplitude magnification factor. If the depth of cut is lower than the limiting depth for the given speed ($\lambda < 1$), then the system is stable.

Although the cutting process does not necessarily adhere to these states during variable speed cutting, the stability of these states certainly has an influence on the overall stability of the process. These equilibrium conditions are nevertheless, what the dynamic

system is approaching as the speed varies. Thus stabilization is aided when speed oscillations span steady states that have lower magnification factors than the original fixed speed system. This aspect is what Jemielniak and Widota discussed in their analysis. The same reason results in variable speed cutting being less successful at large depths of cuts than at small depths of cut. The amplitude magnification factor is greater at large depths for all spindle speeds.

The most effective way speed oscillation stabilizes is through transient fluctuation of the phase shift away from the steady states dictated by the stability lobe analysis. Effective stabilization does not permit the phase shift to settle to steady values to develop wave regeneration. In light of this, the second influential factor is the speed variation amplitude and frequency. These parameters determine the rate at which steady states are changing. Generally speaking, if variations are *fast enough*, then the process may not have an opportunity to set up wave regeneration as dictated by the equilibrium condition, be it stable or unstable.

The last determinant forms a logical link with the other two described so far. The first element of the three is the stability of the states that the system is trying to converge on. The second element is the speed at which these states are changing. The final element is how fast the system can respond to changes in cutting speed to settle to an equilibrium state. The faster the process can settle to a steady state, the more likely it is to set up wave regeneration and destabilize the system.

The main reason behind variable cutting speed stabilization is the avoidance of the steady states which develop chatter by wave regeneration. Therefore, it is justified to reason that the above three factors are most influential in determining effectiveness. The three together dictate how close the dynamic system can approach to the steady states, and how unstable those state are.

3.5 Conclusions

Various models are introduced to investigate the ability of speed oscillation to stabilize milling. Time domain turning simulations and frequency domain analysis established the reasons why variable speed can stabilize cutting. The dependency of stability on speed allows for greater stability when oscillation results in a range of speeds that are, on the average, more stable than the original fixed speed. More importantly, variable cutting results in continuously fluctuating phase shift between the inner and outer modulation, preventing the most powerful wave regeneration mechanism to develop. A digital milling simulation is developed in the following section to predict the effects of speed oscillation under actual milling conditions.

Turning Simulation Parameters					
Nominal Spindle Speed - 600 rpm					
Feed rate - 1.5 mm/sec					
Specific Cutting Pressure - 2000 MPa					
Modal Direction - $\alpha = 30^0$					
Cutting Force Direction - $\beta = 70^0$					
X' - Direction			Y' - Direction		
$f_{x'}$	$K_{x'}$	$\zeta_{x'}$	$f_{y'}$	$K_{y'}$	$\zeta_{y'}$
[Hz]	[KN/m]		[Hz]	[KN/m]	
100.0	40000.0	0.05	120.0	56000.0	0.05

Table 3.1: Turning simulation parameters.

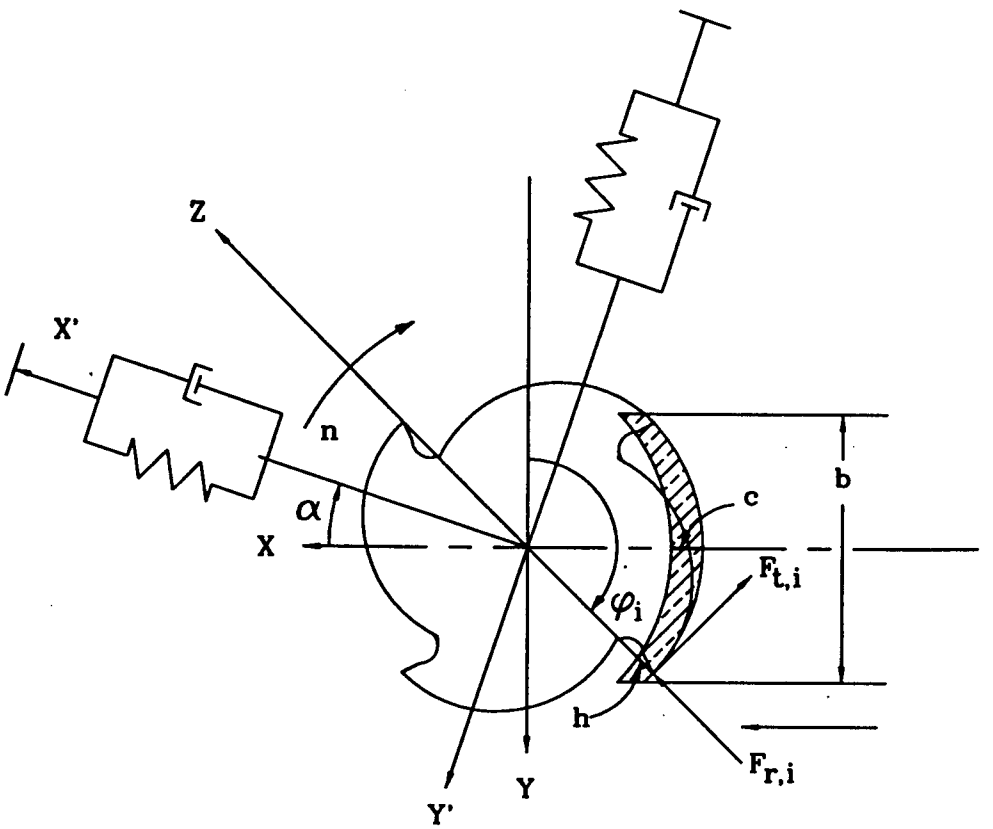
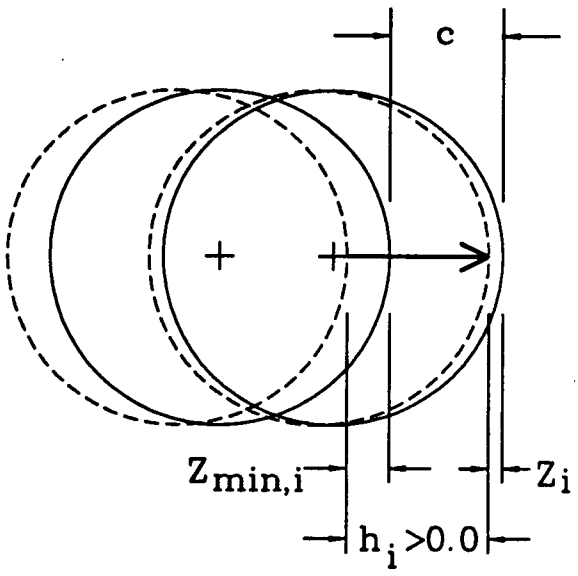
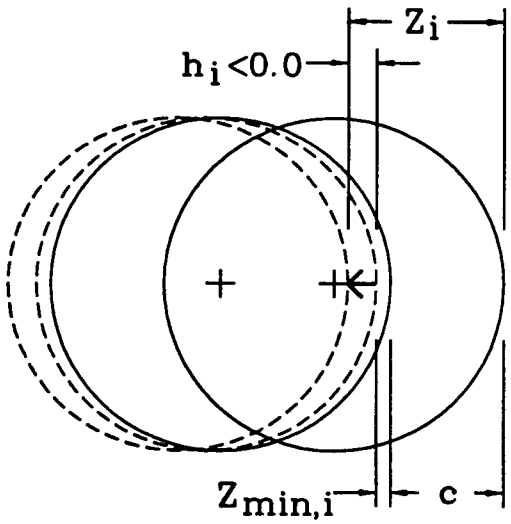


Figure 3.1: Milling geometry for simulation.



(a)



(b)

Figure 3.2: Positive and *negative* uncut chip thickness.

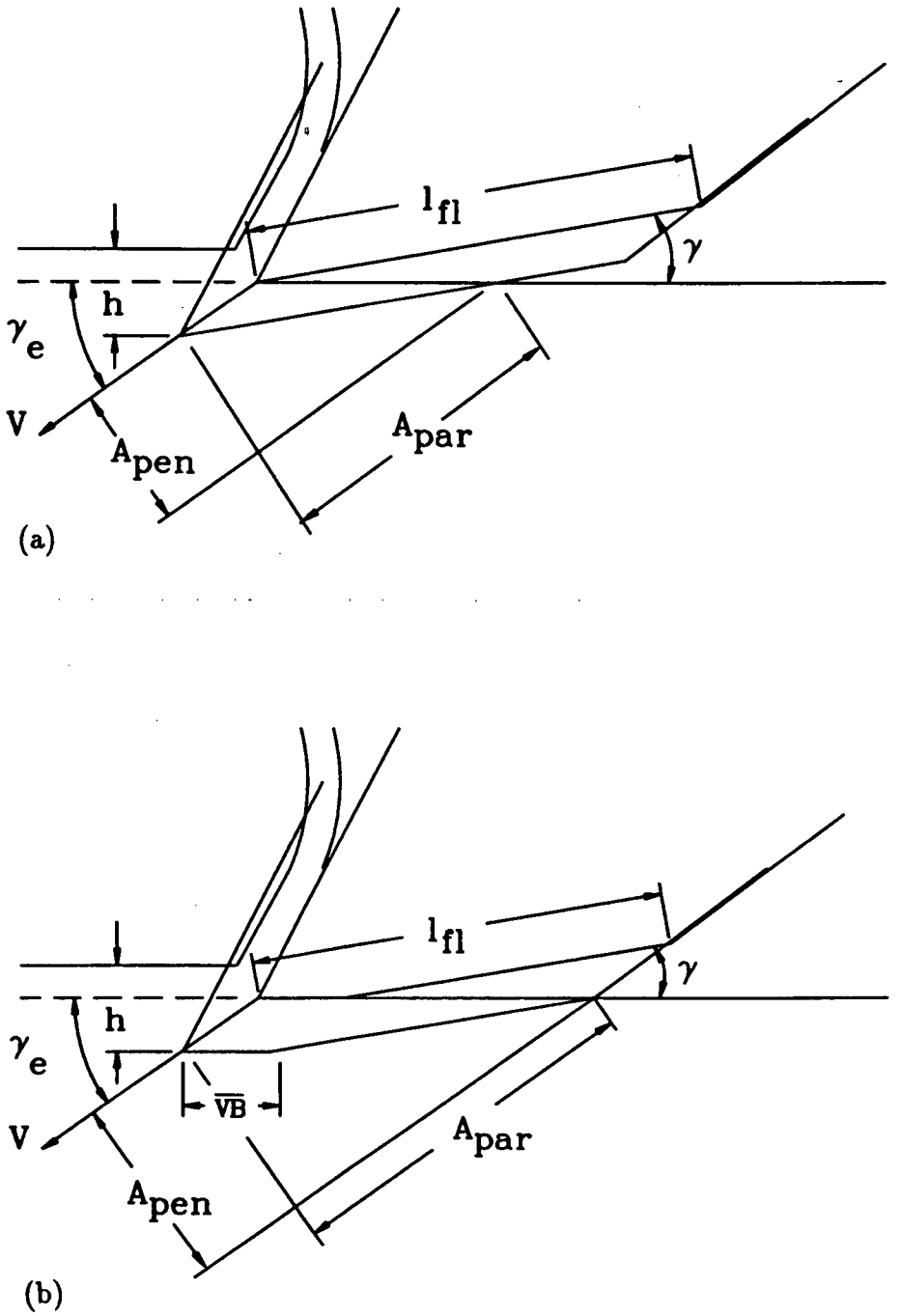


Figure 3.3: Ploughing geometry for sharp and worn tool.

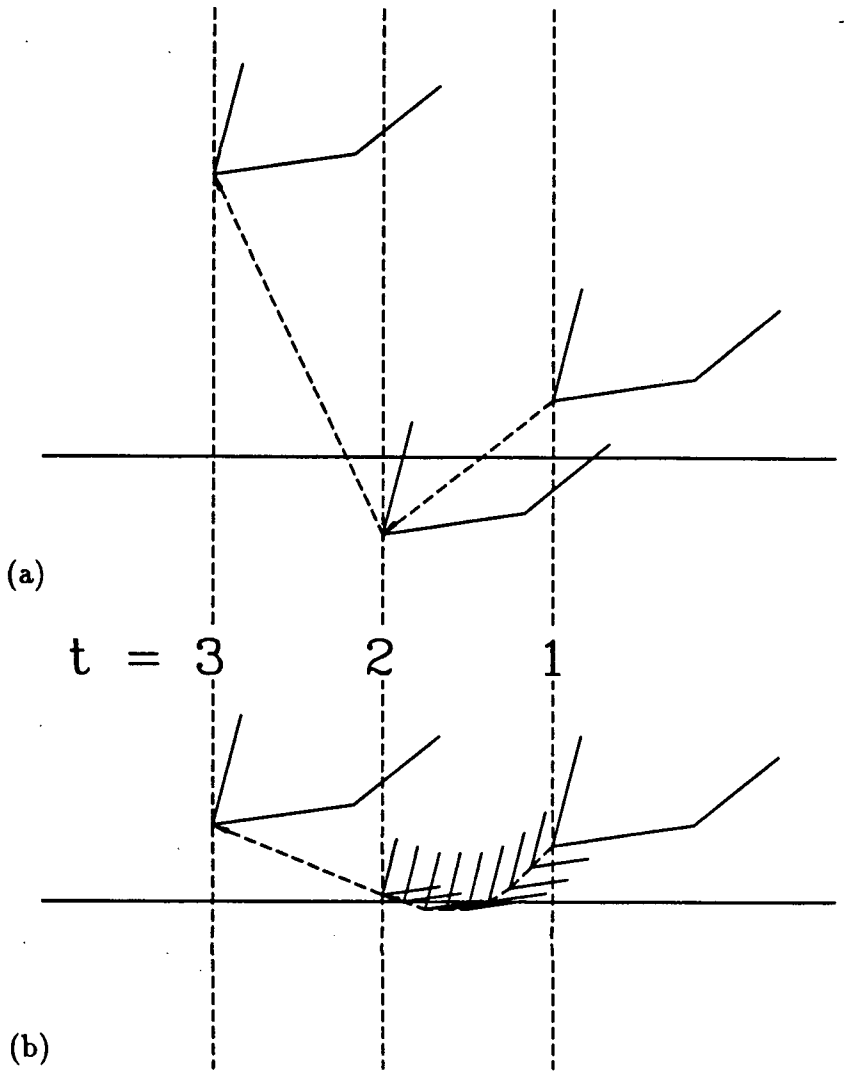


Figure 3.4: Implementation of small time steps during ploughing.

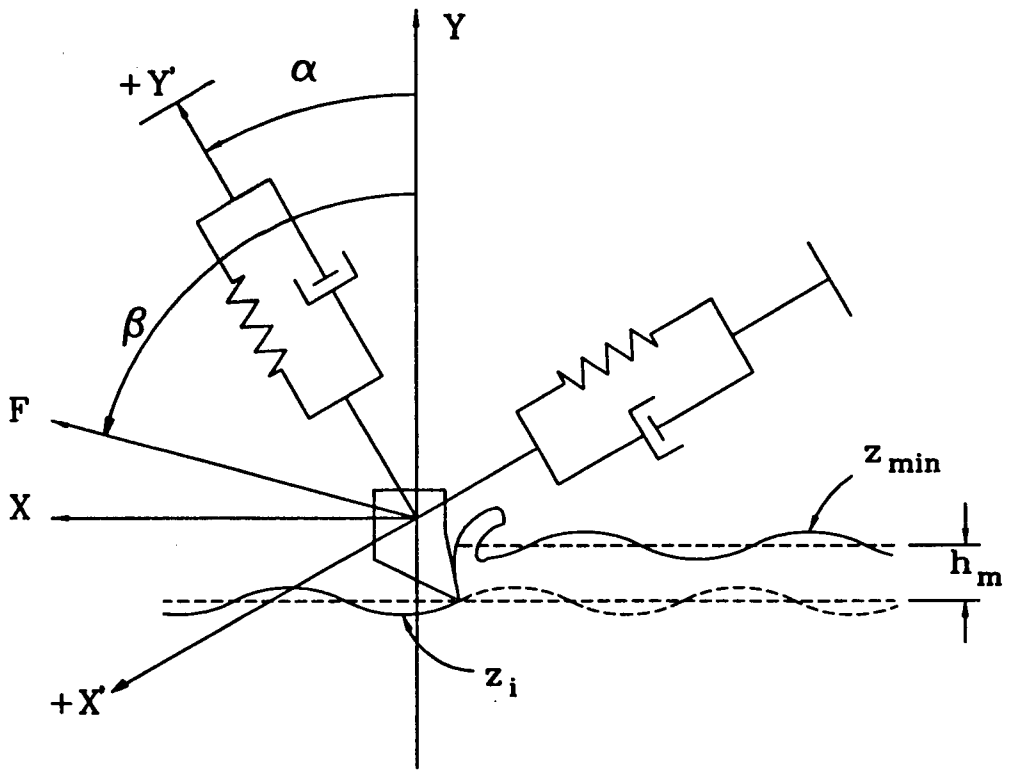


Figure 3.5: Turning simulation geometry.

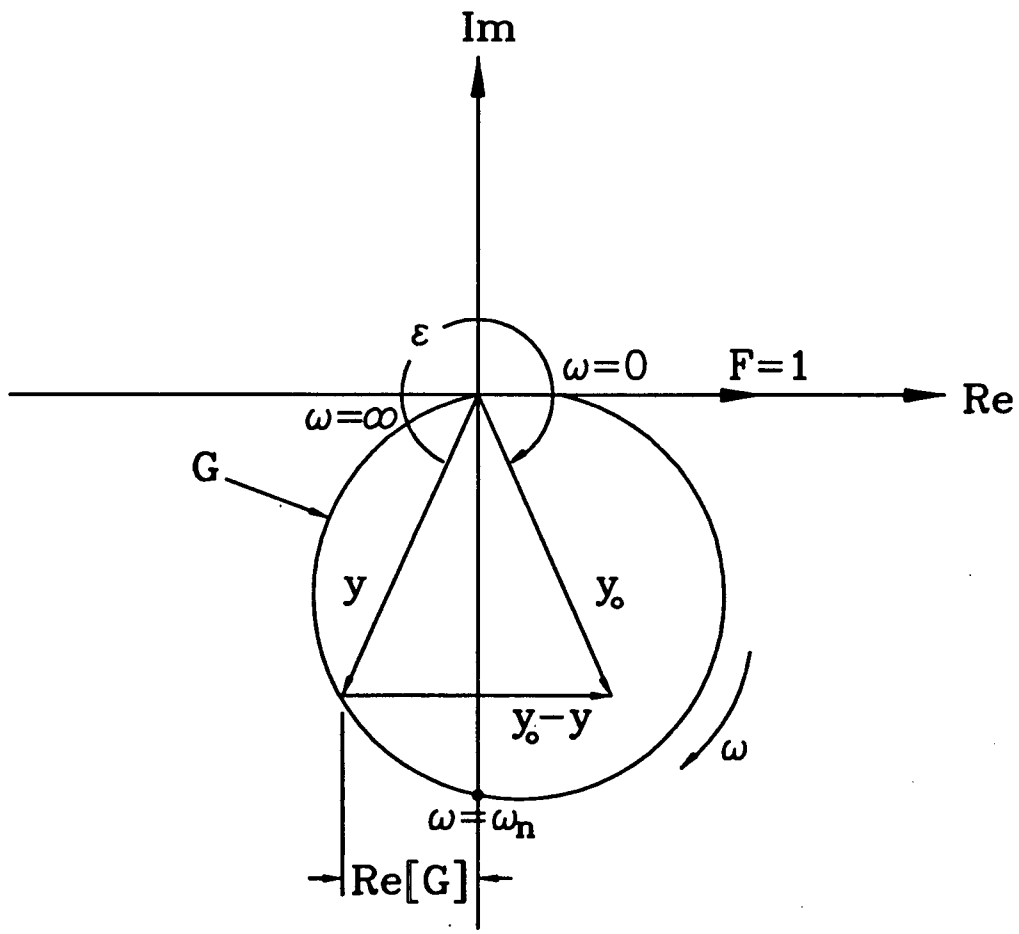


Figure 3.6: Phase plane plot at the limit of stability.

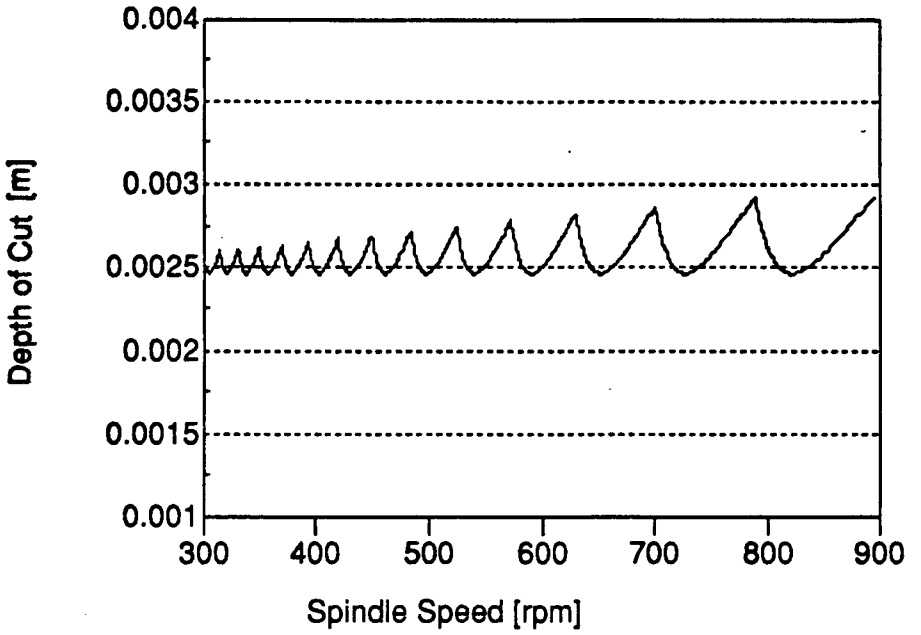


Figure 3.7: Stability lobe curve for turning simulation.

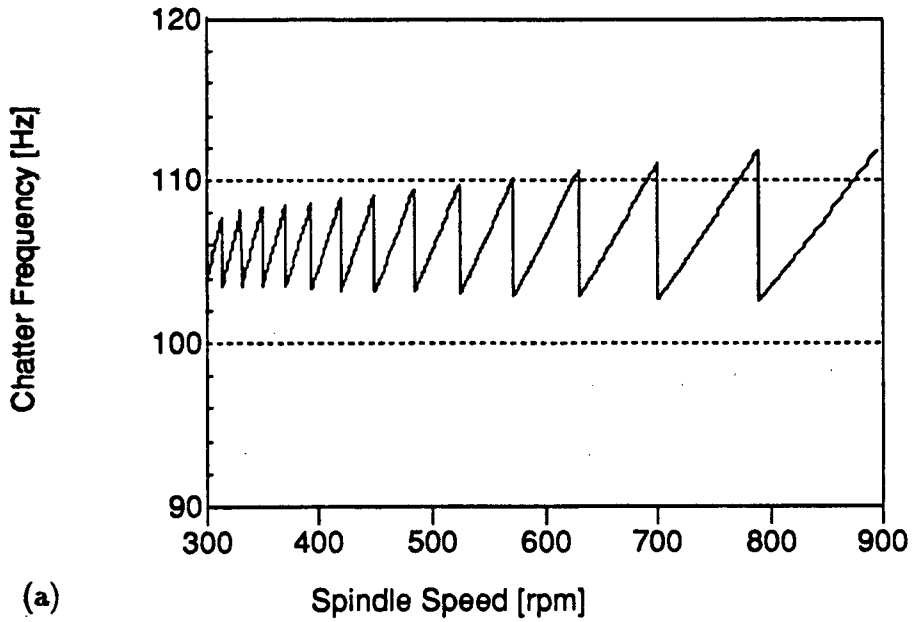
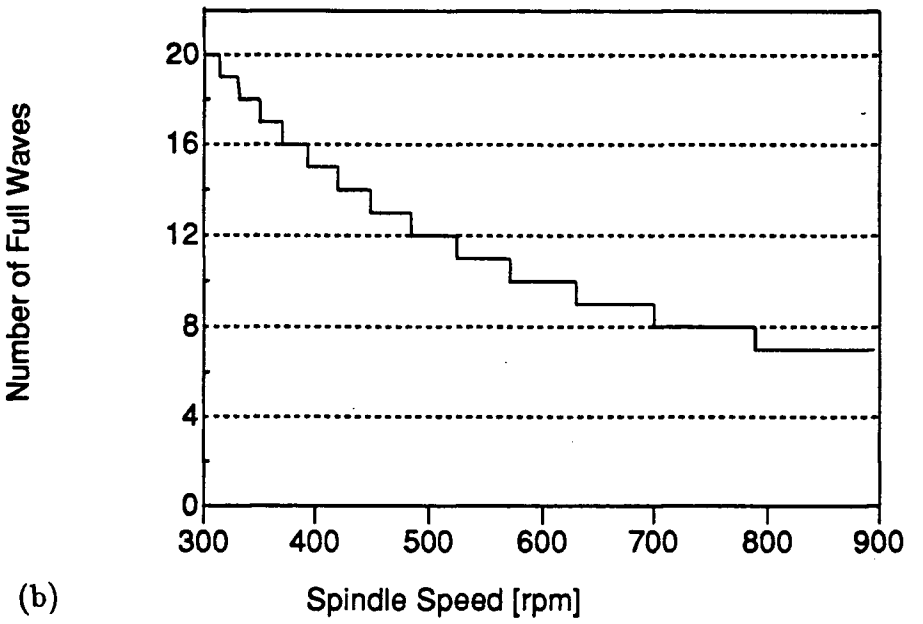
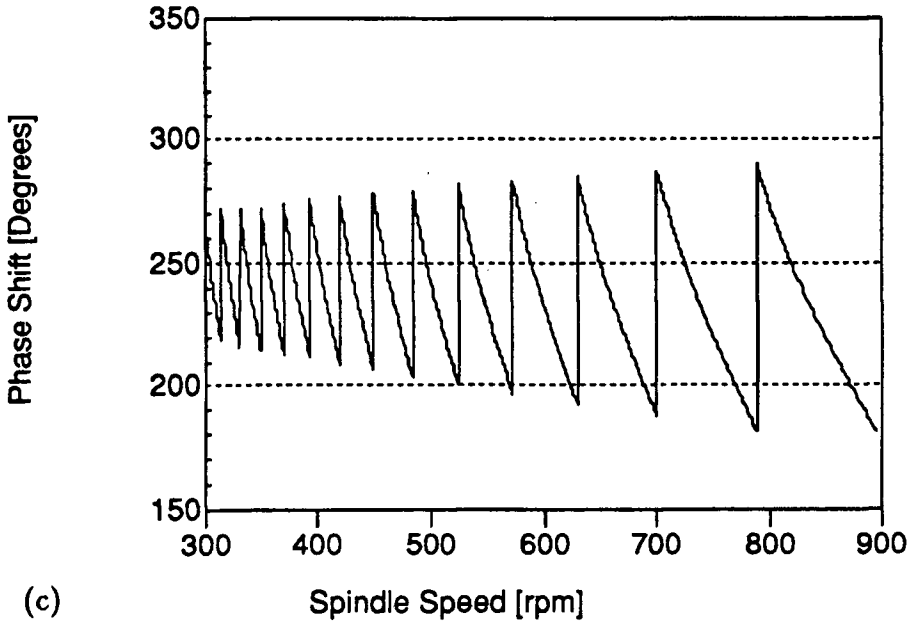


Figure 3.8: Variables associated with stability lobe curve.



(b)



(c)

Figure 3.8: Variables associated with stability lobe curve.

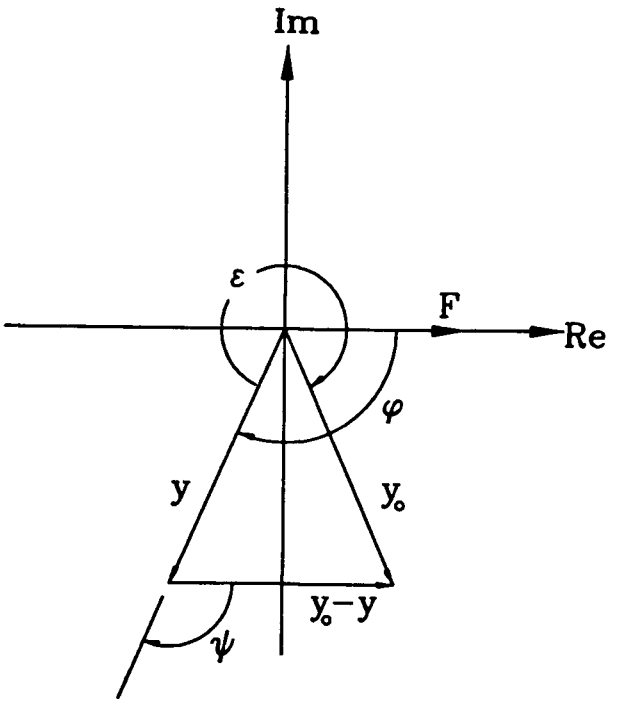


Figure 3.9: Relationship between cutting force (F), present vibration (y), and previous vibration (y_o).

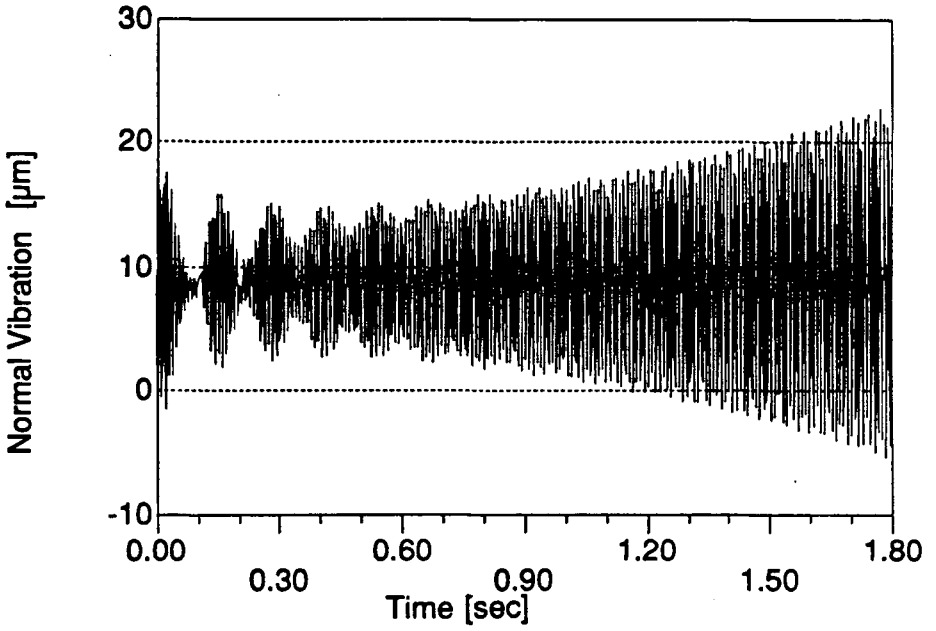
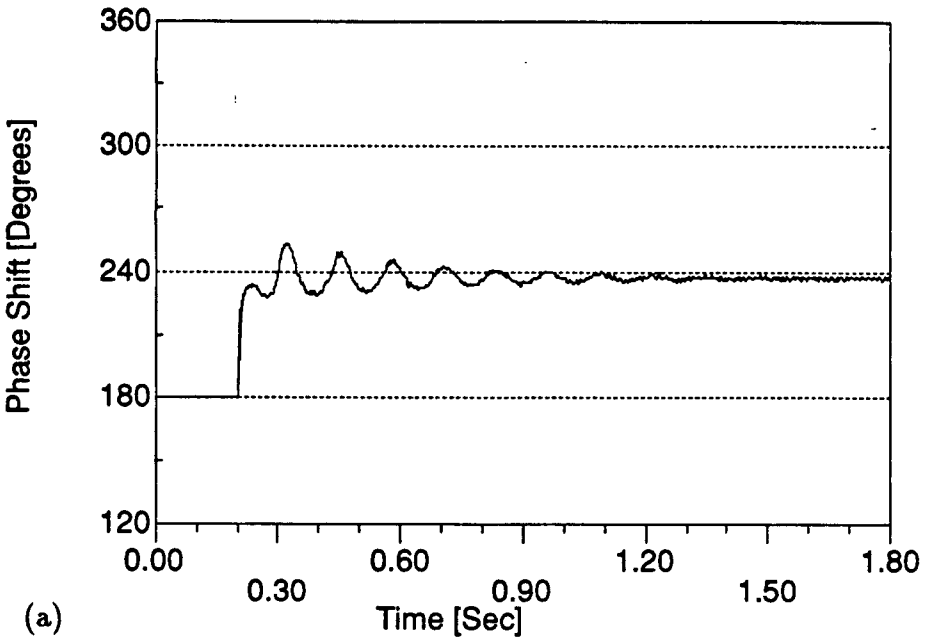


Figure 3.10: Normal vibration of fixed speed turning simulation - unstable. $n = 600$ [rpm], $a = 2.7$ [mm].



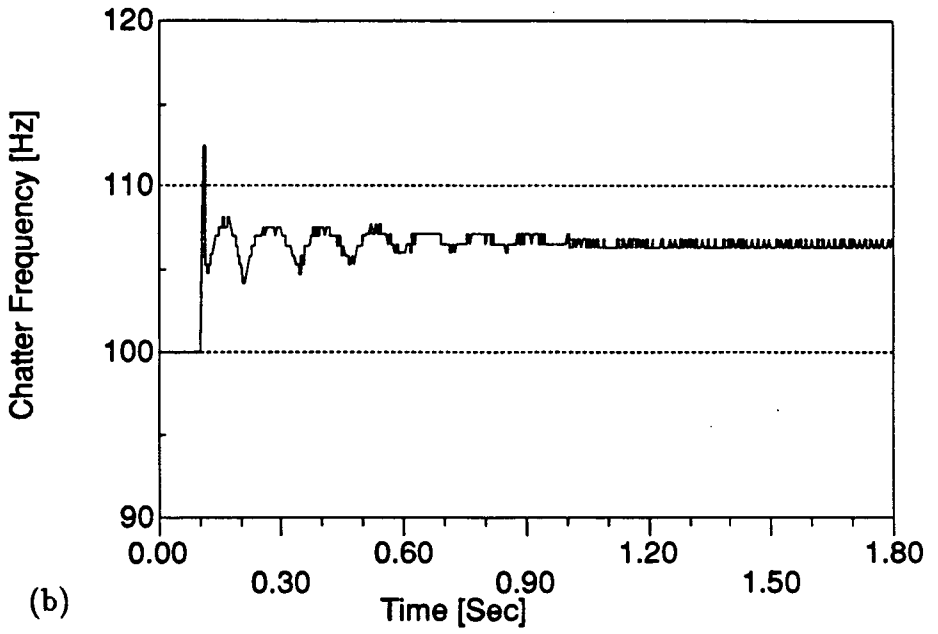


Figure 3.11: Chatter frequency and phase shift for fixed speed turning - unstable. n = 600 [rpm], a = 2.7 [mm].

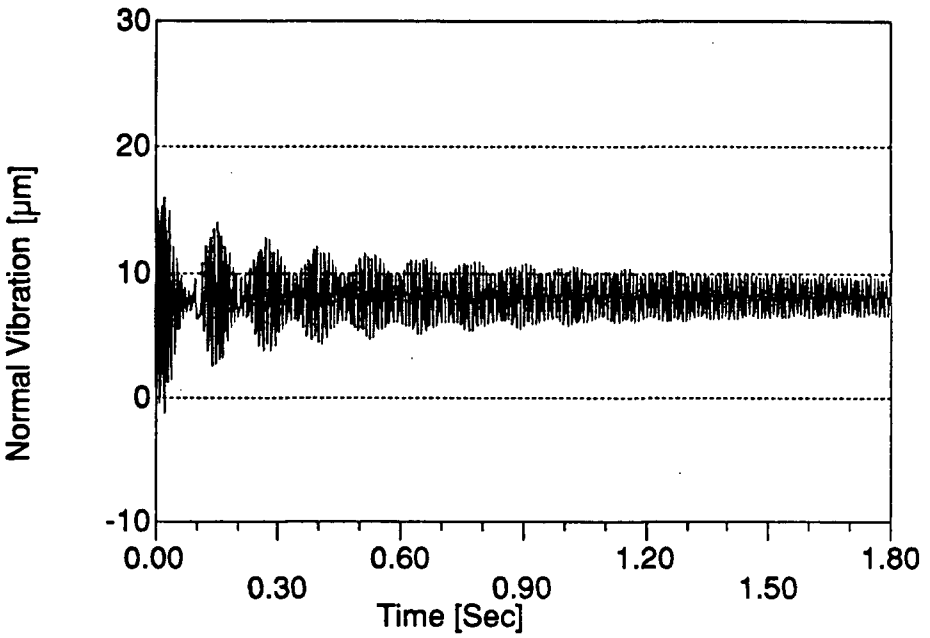


Figure 3.12: Normal vibration of fixed speed turning simulation - stable. $n = 636$ [rpm], $a = 2.5$ [mm].

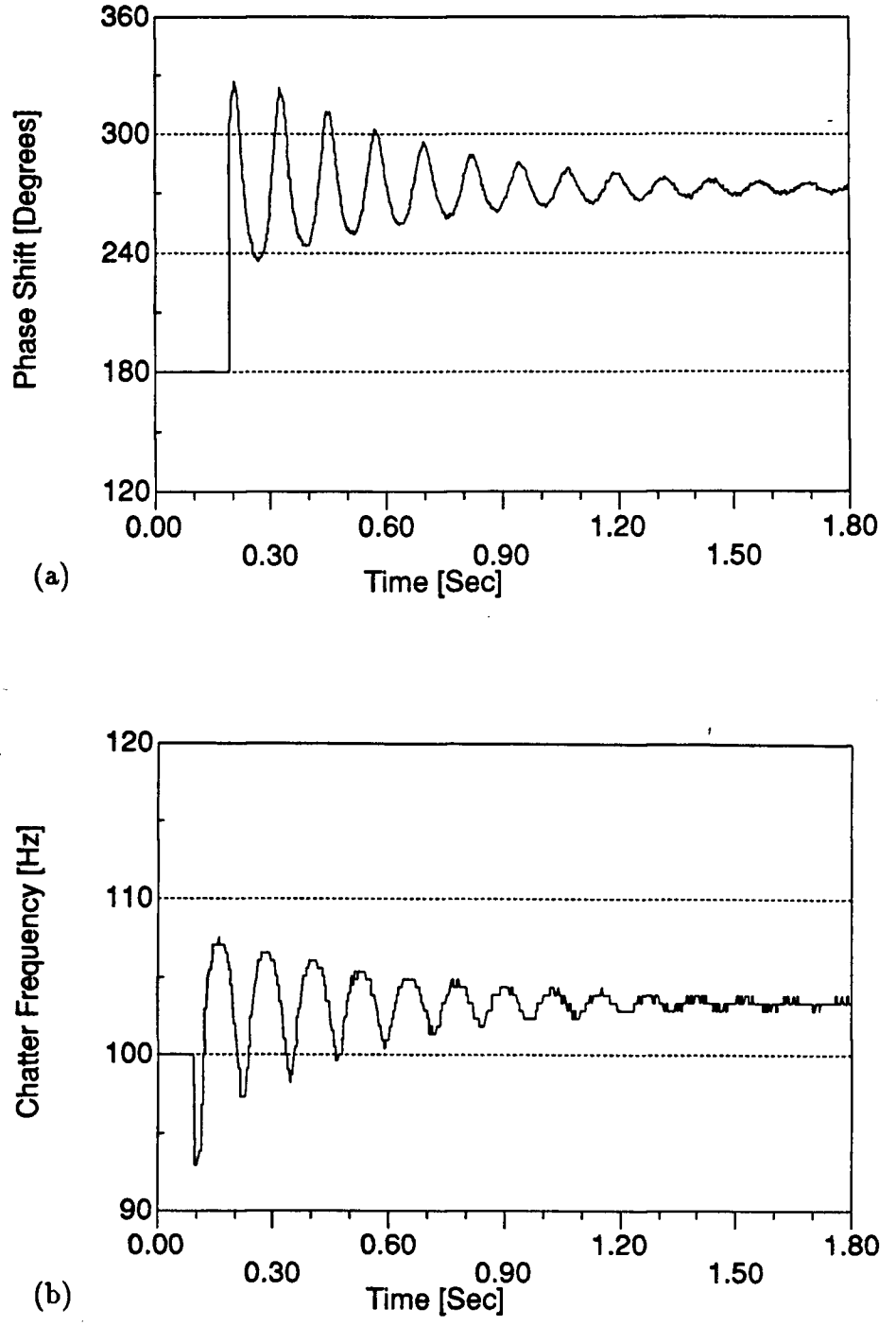


Figure 3.13: Chatter frequency and phase shift for fixed speed turning - stable. $n = 636$ [rpm], $a = 2.5$ [mm].

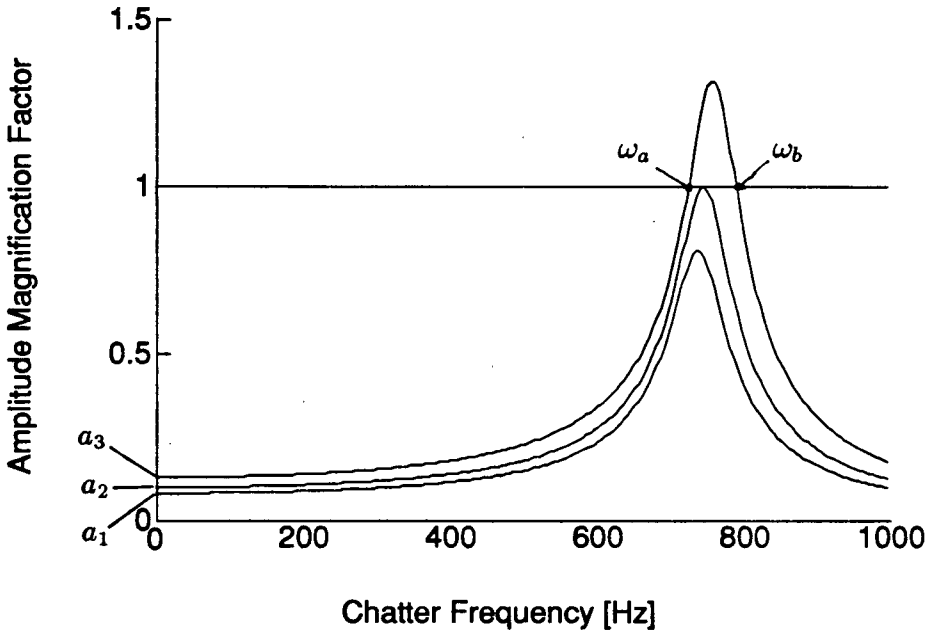


Figure 3.14: Amplitude magnification factor versus chatter frequency.

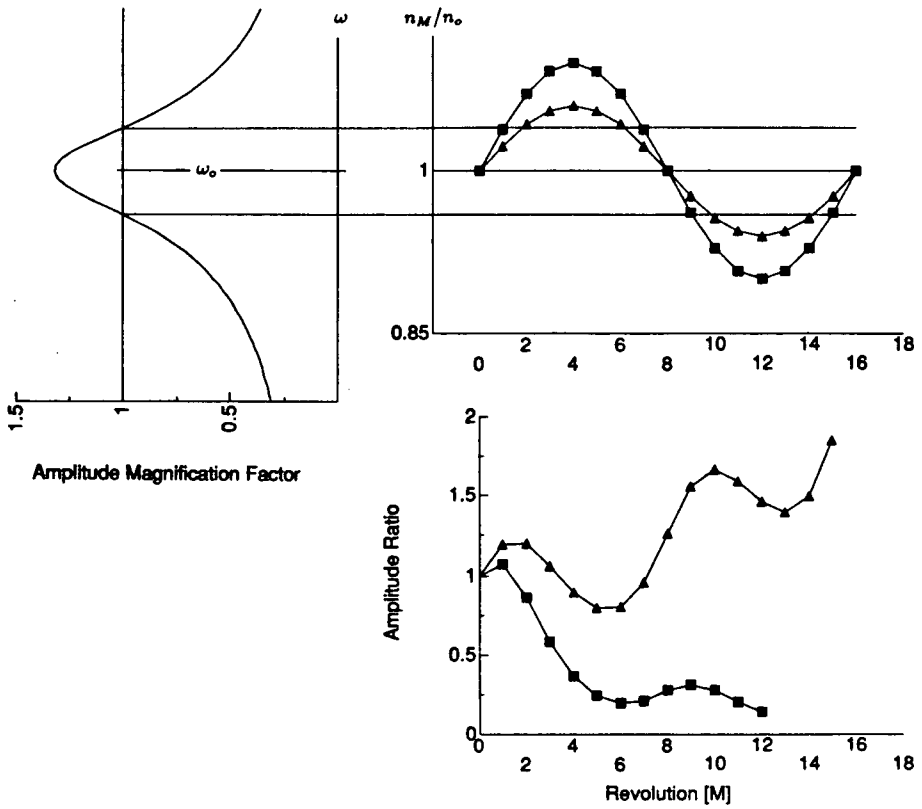


Figure 3.15: Theoretical prediction of amplitude ratio.

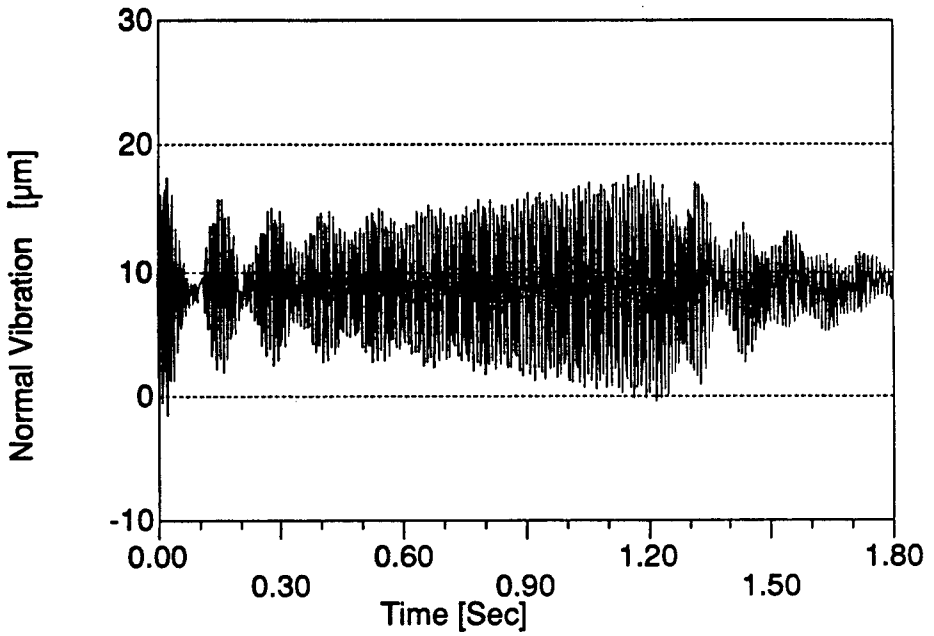
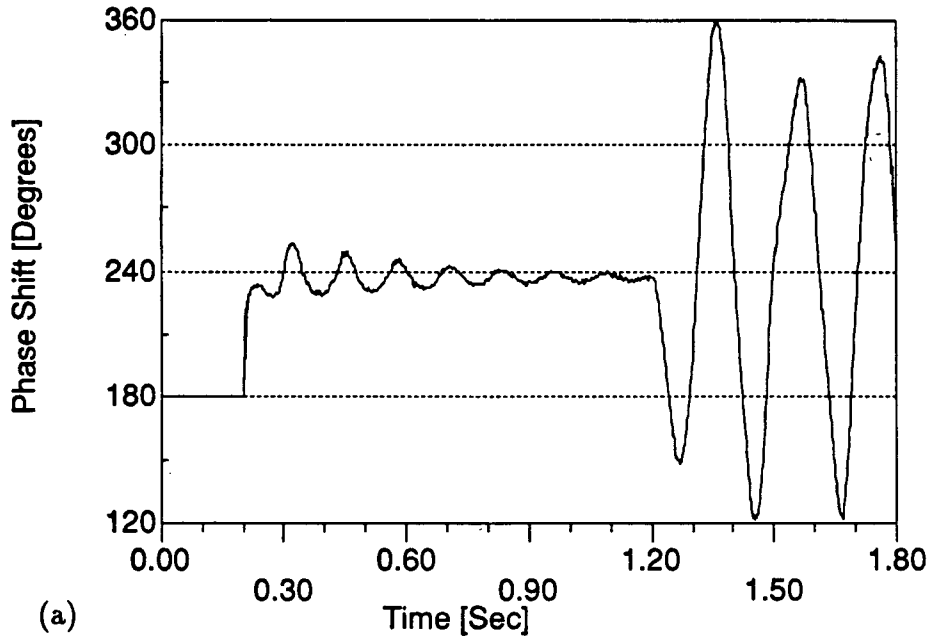


Figure 3.16: Normal vibration of variable speed turning simulation. $n = 600$ [rpm], $a = 2.7$ [mm].



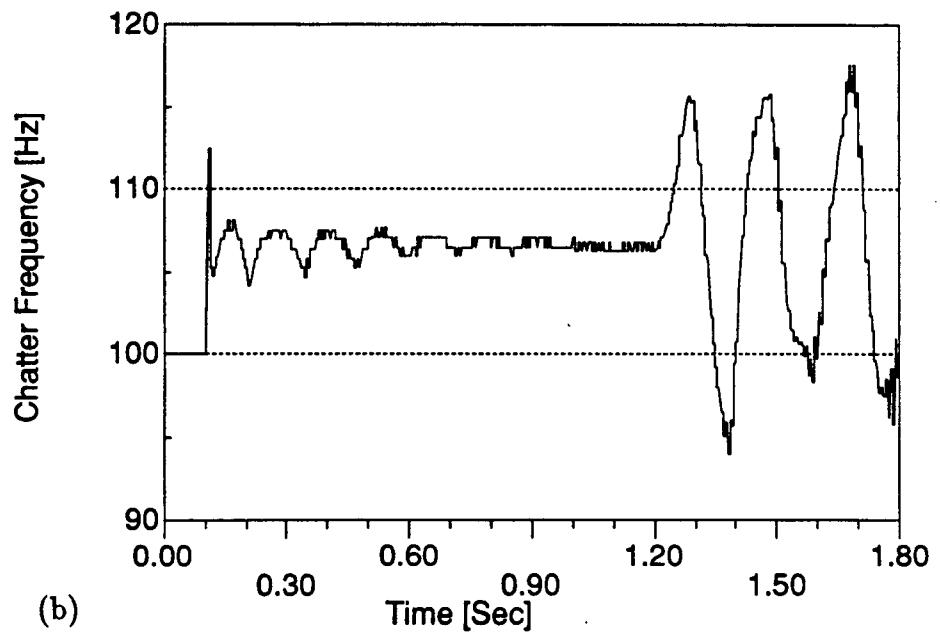


Figure 3.17: Chatter frequency and phase shift of variable speed turning simulation. n = 600 [rpm], a = 2.7 [mm].

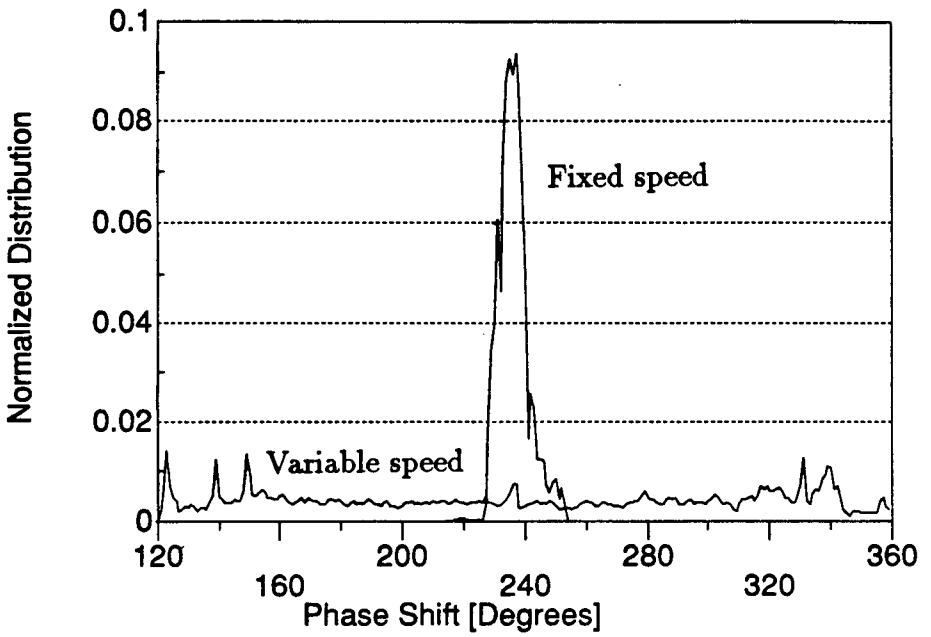


Figure 3.18: Phase shift distribution of fixed and variable speed turning.

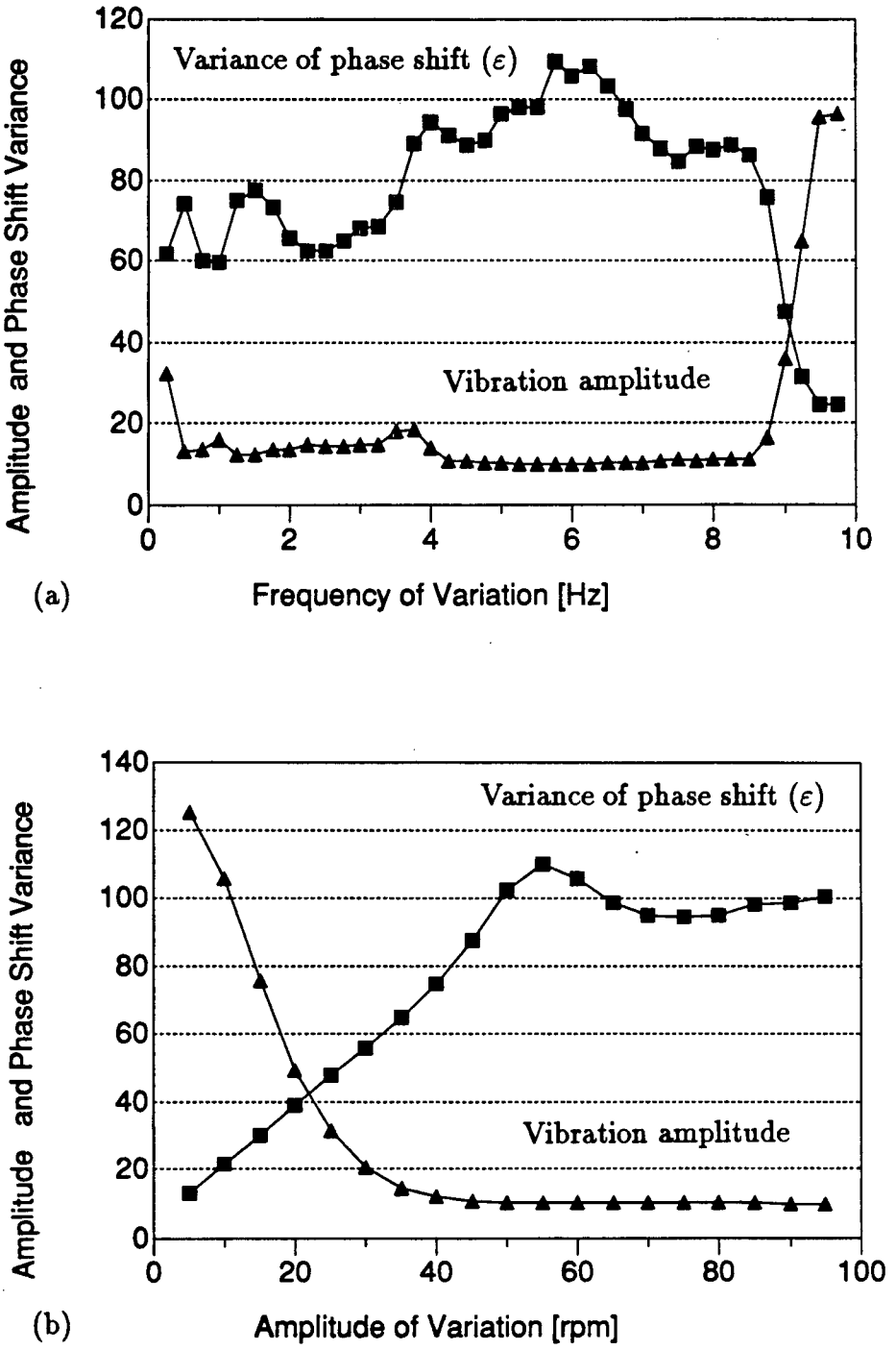
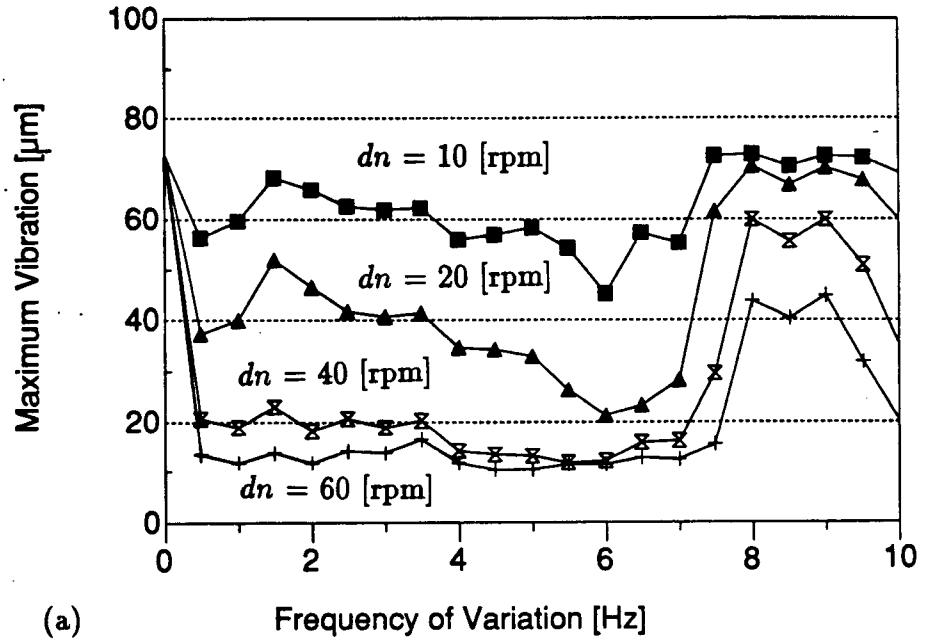


Figure 3.19: Comparison of phase shift variance with variation parameters. $n = 600$ [rpm], $a = 2.7$ [mm].



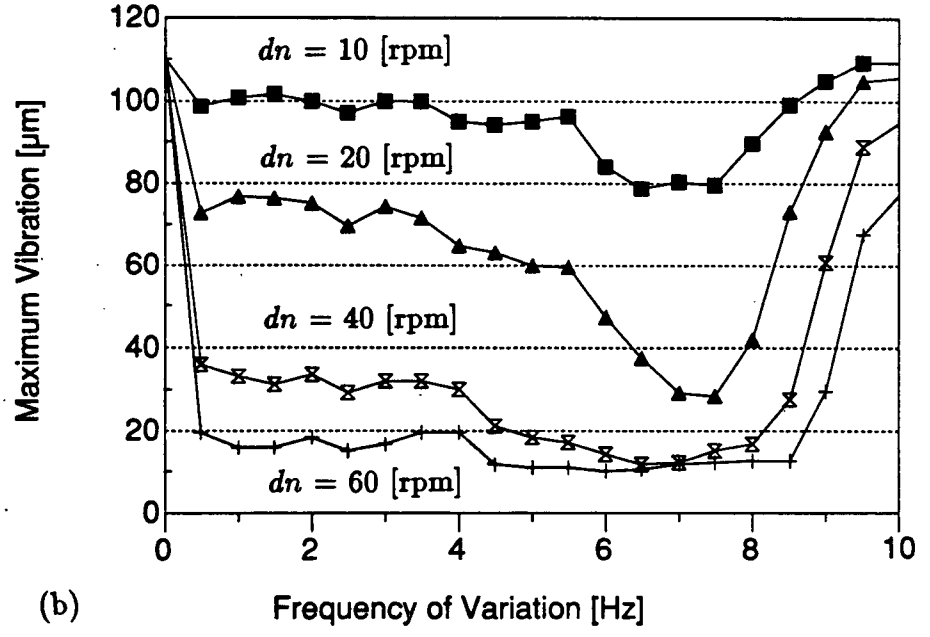


Figure 3.20: Maximum vibration with different variation frequencies and amplitudes. n = (a) 500, (b) 600 [rpm], a = 2.7 [mm].

Chapter 4

Simulation and Experimental Results of Dynamic Milling

4.1 Introduction

Results of dynamic milling tests are presented in this chapter. Simulation results from the milling algorithm introduced in the previous chapter are shown first. Fixed speed simulations illustrate the stability characteristics of dynamic milling, while variable speed simulations introduce the effects of variable speed on milling stability. The presentation of simulation results concludes with a discussion on nonuniform tooth pitch cutters. A general description of the experimental set-up is presented next. Experimental verification of fixed speed and variable speed dynamic milling stability follows.

4.2 Simulation Results for Dynamic Milling

4.2.1 Fixed Speed Milling

Having detailed the milling algorithm and the basis of variable speed cutting, the results of the time domain milling simulation are presented in this section. Various parameters which affect cutting stability are studied. All of the simulations are generated with the parameters listed in Table 4.1, unless otherwise specified. Note that all simulations are either up milling or full immersion cuts. The structural parameters represent the dynamics of a 25.4 [mm] diameter high speed steel end mill with 13.2 [cm] overhang. The specific cutting pressure, cutting force ratio, and yield strength are approximate for the workpiece material of 7075-T6 aluminum. Fixed speed simulation results are

examined first.

The first set of figures illustrates the widely known dependency of stability on the depth of cut. The plots show results for half immersion experiments at 600 [rpm]. The (X) force time responses in Figure 4.1 show the increase in instability as cutting depth is increased from 0.625 [mm] (in Figure 4.1(a)) to 0.781 [mm] (in Figure 4.1(b)) to 0.977 [mm] (in Figure 4.1(c)) . The periodic modulation of forces is a result of the static modulation of uncut chip thickness for a 90 degree immersion cut. However, the growth of high frequency force fluctuations, especially from Figure 4.1(b) to Figure 4.1(c), are a result of increased self-excited instability. The corresponding (X) deflection time histories are shown in Figures 4.2(a), 4.2(b), and 4.2(c). They show the increase in vibrations as cutting depth increases. The detrimental influence of axial depth of cut can also be seen in Figure 4.3, which plots the peak-to-peak (ptp) (X) vibration associated with increasing cutting depths for half immersion up milling.

The time domain responses in Figure 4.4 and Figure 4.5 show a similar trend of decreased stability with increased width of cut. The following figures show the (X) force responses (Figure 4.4) and (X) deflection responses (Figure 4.5) for immersion angles of 50 degrees ((a) figures), 70 degrees ((b) figures), and 90 degrees ((c) figures). The spindle speed is 600 [rpm] and the depth of cut is 1.22 [mm]. Figure 4.4(a) shows the periodic forces of the 50 degree immersion cut. The periodic absence of cutting force is characteristic of low immersion cuts. When the angle between the teeth on the cutter is larger than the immersion angle, there is no tool-workpiece contact at certain times during the tooth period. As the immersion angle increases, the amount of time that the cutter is in cut increases as well. Subsequently, there is more wave regeneration and instability increases (see Figures 4.4(b) and 4.4(c)). Figures 4.5(a), 4.5(b), and 4.5(c) show the corresponding (X) deflections for the forces shown in Figure 4.4. The (X) deflection plots also reflect the increased instability at large widths of cut (especially Figure 4.5(c)).

In addition, Figure 4.6 plots the maximum amplitude for different immersion angles with the depth of cut constant at 1.22 [mm]. The decreased stability at large depths or widths of cut is attributed to increased wave regeneration due to the large cut surface.

The previous figures were all generated with process damping with the simulation parameters listed in Table 4.1. Flank interference during dynamic cutting increases stability as described in Section 2.8. Simulation results indicative of this process are shown in Figure 4.7 and Figure 4.8. Figure 4.7 compares the maximum vibration for various cutting depths between simulations with and without ploughing forces. The immersion is 90 degree up milling and the spindle speed is 600 [rpm]. There is minimal difference between the two curves in amplitude during stable cutting. However, as wave regeneration destabilizes the system and increases vibrations in the no ploughing case, flank interference forces increase to partially counteract the instability and limit the vibration in the ploughing case. Without the inclusion of ploughing forces, deflection increases sharply with cutting depth at approximately 0.55 [mm]. The limiting depth of cut for chatter increases to 0.87 [mm] with ploughing forces. The chatter limit is usually quite clear from plots of deflection or force versus depth of cut. The deflection is primarily a result of static deflection when the depth is below the limit; thus there is a gradual increase in vibration or force amplitude. However, when the depth is greater than the chatter limit, amplitudes grow much faster with depth of cut as self-excited vibrations dominate. In Figure 4.7, the peak-to-peak (X) deflection for the no ploughing case increases gradually until 0.55 [mm], where chatter starts to develop. On the other hand, the increase in deflection of the ploughing case is gradual until a depth of 0.87 [mm].

Figure 4.8 illustrates the cutting speed dependency of process damping as reflected in vibration amplitudes. The depth of cut is 0.977 [mm] and the immersion is 90 degrees. Increased instability is apparent as spindle speed increases, as first explained by Tobias

[19]. The peak-to-peak (X) deflection continues to grow as the spindle speed increases because of decreasing flank interference.

The complication of cutter runout can also be examined. The runout used in the following simulation is 0.09 [mm], 0.13 [mm], and 0.04 [mm] long relative to the first tooth. These values were measured from the actual high speed steel end mill mounted on the research milling machine. The spindle speed is 600 [rpm] and the immersion is 90 degree up milling. The feed per tooth is 0.05 [mm]. Figure 4.9 compares the (X) force histories between the cases with and without runout. Figure 4.10 shows the (X) deflection for the two cases. In the case with runout (Figures 4.9(b) and 4.10(b)), one tooth does not cut during stable machining because of the large runout relative to the feed per tooth. Thus runout results in an uneven chip load which disturbs wave regeneration. While the maximum force in the runout case is larger (see Figure 4.9(b)), dynamic forces are significantly less than the no runout cut (see Figure 4.9(a)). With an uneven chip load, there is more time for vibrations to damp out between tooth passes, as observed in Figure 4.10(b). In addition, the phase shift between the inner and outer modulation is disrupted from pass to pass. To illustrate this disturbance from runout, the sinusoidal chip thickness variation inherent in milling is simplified to a linear variation. Thus two succeeding tooth passes with no runout may be illustrated by Figure 4.11(a). If the tooth of the second pass is short, then the process and resulting surface will be as shown in Figure 4.11(b). The second tooth does not start cutting at the beginning of the pass because of the runout. Thus the surface has a discontinuity (in the circled area) where the phase changes abruptly. The next tooth encountering this surface causes the system to start adjusting itself to the outer modulation left two tooth periods ago. However, part way through the cut, the phase of the outer modulation changes and the system must readjust itself. Runout provides a way to delay the system from establishing a fixed phase shift and thus the most unstable state. The effect of different runout magnitudes has

not been investigated, but following the same reasoning as above, maximum stabilization should be achieved when the outer modulation phase discontinuity is in the middle of the tooth pass. This way, the vibrational system has equally little time to adjust to either of the outer modulation and converge to a steady state.

The inclusion of a flank wear flat also has the expected influence on vibrations. Flank wear increases flank interference with the cut surface, bringing about more stability from increased process damping. The next two figures compare the (X) force (Figure 4.12) and the (X) deflection (Figure 4.13) for half immersion up milling cuts with and without a flank wear flat. The axial depth of cut is 0.781 [mm] and the spindle speed is 600 [rpm]. The length of the flat is 0.02 [mm] and the primary flank face length is 0.7 [mm]. Comparison of the force histories in Figure 4.12 shows the improved stability of the worn tool case (Figure 4.12(b)) over the sharp tool case (Figure 4.12(a)). The force is significantly less vibratory in the simulation with a flank wear flat. The majority of vibrations in the sharp tool case (Figure 4.13(a)) is damped out by flank interference in the worn tool case (Figure 4.13(b)). Further proof is evident in the next plot (shown in Figure 4.14), which compares the peak-to-peak (Y) deflection between sharp tool and worn tool milling at different depths of cut. Apparent from the graph, the inclusion of the flank wearland raises the chatter limit from 0.57 [mm] to 0.87 [mm]. The length of the flank wearland used here and in all other milling simulations was chosen to give a reasonable combination of stabilizing effect and simulation efficiency. Larger flats require higher digitization frequencies in order to avoid inaccuracies during initial impact of the tool flank.

4.2.2 Variable Speed Milling

The next series of figures show simulation results for variable speed milling. It is necessary to set the criteria for comparison between fixed and variable speed cutting before

discussing the results. Two parameters can be used: cutting forces and deflections. The maximum magnitudes of the force or deflection can be compared, but a better way is to compare time domain signals after a fixed time when transients of the computer simulation have diminished. In this way, the maximum magnitudes can be observed as well as the vibratory content. However, the latter method is significantly more time consuming because it does not lend itself to an objective comparison by computer. Therefore only maximum values are shown for comparison in most cases. Time domain plots are occasionally shown for illustrative purposes.

Another factor that must be mentioned in determining effectiveness of speed oscillation is the inherent modulation of the chip load from spindle speed variation. A fixed feed rate coupled with a variable spindle speed results in a changing feed per tooth which causes modulation of the cutting forces and static deflections. This is a stable modulation and should not be considered as a sign of greater instability.

An additional point that should be noted is the vast number of variables that may change the stability of the simulations, variables which are associated with actual cutting as well as ones that are only part of the computer algorithm. Due to the complexities of their influences, not all these factors could be examined with respect to the stabilizing ability of variable cutting speed.

The first set of figures contains time and frequency domain plots that illustrate the stability improvement from spindle speed oscillations. The depth of cut in these simulations is 0.977 [mm] and the immersion is 90 degree up milling. Figure 4.15 to Figure 4.18 compare the (X) force and (X) deflection and their spectrums between a fixed speed cut (the (a) figures) and a cut with the same cutting conditions but with spindle speed oscillation (the (b) figures). The nominal speed for both cases is 600 [rpm]. All variable speed cutting simulations in this chapter are produced with 50 [rpm] variation amplitude and 2.5 [Hz] variation frequency, unless otherwise specified. Figure 4.15(a) shows the (X)

force for the fixed speed cut. There is significant high frequency force variations due to chatter. The static force modulation due to chip thickness variation is still distinguishable, but the cut is definitely not stable. The same cut simulated with variable speed, however, is more stable (see Figure 4.15(b)). There is a marked decrease in peak forces, but more importantly the vibratory content is reduced.

Figure 4.16 compares the frequency spectrums of the (X) force. The fixed speed spectrum (in Figure 4.16(a)) is dominated by the tooth frequency, its harmonics, and the chatter frequency. The tooth frequency of 40.0 [Hz] has the largest magnitude since the static uncut chip thickness modulation is still very prominent. However, as the tooth frequency harmonics decrease in magnitude, they give way to the chatter frequency which peaks at approximately 900 [Hz]. A characteristic of chatter is the prominence of dynamics close to the tool natural frequency, which is 800 [Hz] for the (X') direction in this case. The (X) force spectrum for the variable speed cut, shown in Figure 4.16(b), exhibits much less high frequency dynamics. The strength of the high frequency signal between 600 [Hz] and 1000 [Hz] is reduced as a result of the increased stability. The maximum magnitude in this frequency range is only slightly larger than the second tooth harmonic. In fact, the magnitudes of the tooth frequency and its harmonics are also slightly less than the fixed speed case. The spindle speed oscillation of course results in oscillation of the tooth frequency as well.

The (X) deflection time histories in Figures 4.17(a) and 4.17(b) show the same increase in stability from fixed speed to variable speed, as with the (X) forces. Peak-to-peak deflections are reduced from 0.1 [mm] to 0.05 [mm]. The (X) deflection spectrums in Figure 4.18(a) and Figure 4.18(b) also reflect the decrease in vibrational activity in the 600 [Hz] to 1000 [Hz] frequency range.

The next task is to determine some of the factors involved in variable speed milling, i.e. when it is most or least successful. Simulations with different cutting conditions indicate

that there is one fundamental factor that influences its effectiveness. As suggested in the last chapter, the stability of the original fixed speed system is very important. Figure 4.19 shows the peak-to-peak (X) deflection for a fixed speed cut and a variable speed cut, with increasing cutting depths. The immersion for this comparison is 90 degree up milling. The fixed speed curve, similar to others seen in earlier figures, shows stable cutting up to a depth of 0.87 [mm] where vibration amplitude starts to grow rapidly. The variable speed curve exhibits a similar shape, but the limiting depth is increased to 0.97 [mm]. The fixed speed curve can be divided into three regions: Region 1) cutting depths where the system is stable; Region 2) depths where the system is in transition to instability; and Region 3) depths where the system is well into unstable cutting. Speed oscillation has essentially no effect in Region 1 since the system is inherently stable. Slightly larger vibrations actually may result in this region due to the imposed fluctuation on the chip load by variable speed. Peak forces may be higher for the variable speed cut when the reduction in high frequency vibrations is smaller than the sinusoidal modulation in static deflection brought upon by spindle speed variation. Region 2 is where variable speed cutting has the most benefits. With fixed speed, self-excitation is becoming increasingly active and sufficient energy is returned to the system to establish large vibrations, or what is generally called chatter. In this region, speed oscillation disturbs wave regeneration enough to reduce the energy that sustains chatter. However, as cutting depth increases to Region 3, the system gets increasingly unstable and variable speed can no longer disturb wave regeneration enough to reduce vibration. The decrease in vibration from speed oscillation is typically minimal in this region. Nevertheless, speed oscillation succeeds in extending the stability zone or equivalently, increasing the chatter limit.

The results from different immersion angle cuts are presented next. Figure 4.20 compares the maximum peak-to-peak vibrations for fixed and variable speed cuts for various immersion angles. Interestingly, variable speed does not noticeably increase the

chatter limit for immersion angles of 50 and 65 degrees (Figures 4.20(a) and 4.20(b)). The chatter limit for the fixed speed and variable speed cuts are the same for both, 50 degree immersion and 65 degree immersion. On the other hand, stability does improve at immersion angles of 90, 135, and 180 degrees (Figures 4.20(c), 4.20(d), and 4.20(e)). Figure 4.20(c) shows the comparison for a 90 degree immersion cut, which was shown previously in Figure 4.19. The fixed speed chatter limit of 0.87 [mm] is increased to 0.97 [mm] with variable speed. For the 135 degree immersion case shown in Figure 4.20(d), the fixed speed limit is 0.25 [mm], and wave regeneration disturbance increases it to 0.28 [mm]. A similar increase in stability is shown for the 180 degree immersion case in Figure 4.20(e). The reason for the difference in stability for different immersions may be attributed to two factors.

The first factor is the difference in phase shift (ε) between the present vibration and the modulation left by the previous tooth. A change in the immersion angle alters the length of cut since the arc length is equal to the product of the cutter radius and immersion angle. Thus the number of waves left on the surface is also different for different immersion angles. Consequently, speed oscillation may result in different and perhaps less optimal phase shift variation for low immersion angles.

The other factor is the contribution of mode coupling relative to wave regeneration at different immersions. Although theoretical derivations in Section 2.7 suggest that mode coupling induced instability is approximately half as strong as wave regeneration instability, they both typically contribute in actual chatter development. Variable speed suppresses chatter by disturbing wave regeneration, but in the process it does little to deter mode coupling. Variations imposed on the speed are very small compared to the vibration frequency of the system and thus the elliptical motion of the tool during active mode coupling is not altered significantly. Therefore, if instability is attributed mostly to mode coupling, then speed oscillation has negligible benefit. An example of the influence

of mode coupling is presented next.

When a vibratory system has two orthogonal modes that are equal, stability or time domain responses are identical for cutting in either direction (i.e. the feed direction close to either the (X') or (Y') axis). However, when the dynamics of the two modes are mismatched slightly such as with end mills, the direction of cut does change stability characteristics. As explained by Tlusty [6], the difference in stability is attributed to the amount of mode coupling during cutting in each direction. Mismatched axes with the normal direction close to the higher natural frequency cause a more active mode coupling mechanism than when the feed direction is close to the higher mode. More information can be found in [6], but an illustrative comparison is shown in Figure 4.21. The figure shows half immersion cuts for two cases: the typical case where the higher natural frequency is close to the feed (X) direction (which will be called the (X) cut), and the case where the higher natural frequency is close to the normal (Y) direction (which will be called the (Y) cut). The difference in maximum amplitudes between the (Y) direction cut and the (X) direction cut can be observed with all depths of cut. Furthermore, the chatter limit is 0.87 [mm] for the (X) cut and 0.65 [mm] for the (Y) cut because of mode coupling.

More evidence can be found in Figure 4.22. Figure 4.22 compares the peak-to-peak (X) deflection for simulations generated with and without wave regeneration for both, (X) and (Y) cuts. The simulation conditions for the curves with wave regeneration are identical to those of Figure 4.21 except for the exclusion of ploughing forces. The "no wave regeneration" curves are generated by neglecting the previous surface modulations during computation of the dynamic uncut chip thickness. Thus mode coupling is the only self-excitation mechanism in those cuts. Figure 4.22 shows that with the presence of wave regeneration, the (Y) cut is much more unstable than the (X) cut, as was the case in Figure 4.21. The figure also shows that without wave regeneration, the (Y) cut

is still more unstable than the (X) cut. The (Y) cut without wave regeneration becomes unstable from mode coupling at approximately 0.70 [mm], while the corresponding (X) cut is stable up to at least 1.0 [mm].

Due to this mechanism, when the same spindle speed variation is applied to both cases, their results are very different. In the (X) cutting case, variable speed increases the limiting depth of cut as shown previously in Figure 4.20(c). The (Y) cutting case, however, does not exhibit a similar increase; speed oscillation has minimal effect in the (Y) cutting case (see Figure 4.23). The reason is attributed to the strong contribution of mode coupling to vibrational instability in the (Y) cutting case. As mentioned above, variable cutting speed does not prevent mode coupling from making the system unstable.

The frequency (n_f) and amplitude (dn) of speed variation have an important effect on stability, as shown by the plunging results in Section 3.4.2. The amount of wave regeneration disturbance is directly related to these two parameters. It was not possible to derive any relationships between these parameters and stability; however, it became apparent that large values of either were definitely more beneficial than small values. Figure 4.24 illustrates the increase of the chatter limit for large variations in speed. The nominal speed of these half immersion up milling cuts is again 600 [rpm]. The original chatter limit of 0.87 [mm] is increased to 0.97 [mm] for a sinusoidal variation of 100 [rpm] and 2.5 [Hz], and to 1.2 [mm] for a variation of 50 [rpm] and 5.0 [Hz]. When variation is increased to 100 [rpm] and 5.0 [Hz], chatter is not reached until a depth of 1.55 [mm], almost 80% larger than the fixed speed chatter limit. The next two plots compare the maximum (X) deflection for different variation amplitudes and frequencies at two nominal speeds. Figure 4.25(a) shows the vibration for a nominal speed of 800 [rpm] and a cutting depth of 0.781 [mm]. The general trend of reduced vibration amplitudes with increased variation amplitudes and frequencies is quite apparent in this case. The same speed variations applied at 1100 [rpm] and axial depth of 0.625 [mm] are shown

in Figure 4.25(b). The speed variations have less influence at 1100 [rpm] than at 800 [rpm] since they are a smaller percentage of the nominal speed. Thus wave regeneration is disturbed less and stabilized less. For the 1100 [rpm] case, only a 5.0 [Hz] variation frequency succeeds in stabilizing the system.

4.3 Stability of Cutters with Nonuniform Tooth Pitch

The use of milling cutters with nonuniform tooth pitch to increase stability against chatter is compared with the use of variable spindle speed in this section. These cutters increase stability by disturbing the regeneration of surface waviness, like variable speed cutting. Consider three teeth which are vibrating together, and cutting the wavy surface left by the preceding teeth, as shown in Figure 4.26(b). If the pitch between the teeth are nonuniform, then the phase shift (ε) produced between the first tooth and the second tooth, between the second tooth and the third tooth, and between the third tooth (from a preceding pass) and the first tooth will all be different. Thus the most unstable phase shift at that certain speed and depth of cut (as governed by stability lobe analysis) cannot be obtained for all three pitches. As a result, stability is increased over a cutter with uniform tooth pitch. The frequency domain analysis of cutters with nonuniform tooth pitch is similar to that of cutters with uniform tooth pitch. It is summarized here from Tlusty [6].

To appreciate the stability increase from cutters with nonuniform tooth pitch, it is desirable to first analyze cutters with uniform tooth pitch. Consider the geometry of a cutter with three teeth cutting simultaneously, as shown in Figure 4.26(a). The teeth vibrations (y_1, y_2, and y_3) are all equal and in phase,

$$y = y_1 = y_2 = y_3. \tag{4.73}$$

The modulation produced by the preceding teeth are (y_{o1}, y_{o2}, and y_{o3}). Since the teeth

have uniform pitch between them, the phase shifts (ε) between the present vibrations and previous vibrations are the same for all teeth. At the limit of stability, the amplitudes of the surface waviness are equal in magnitude and in phase. In addition, the amplitude of the tool vibration is also equal to the amplitude of the surface waviness since vibration neither increases nor decreases,

$$y_o = y_{o1} = y_{o2} = y_{o3} \tag{4.74}$$

$$\text{and } |y| = |y_o|. \tag{4.75}$$

The resulting phase plane plot is shown in Figure 4.27. Following the same procedures used in Section 3.4.1 to derive the stability limit for a single tooth in cut, a similar limiting depth of cut can be found. The limiting depth of cut for a cutter with uniform tooth pitch and (z_t) teeth in cut is,

$$a_{lim} = -\frac{1}{K_s(\sum y_{oi} - \sum y_i)}.$$

Since the vibrations are equal in phase and amplitude (Equation 4.73), and the surface waves are also equal in phase and amplitude (Equation 4.74), the limit can be rewritten as,

$$a_{lim} = -\frac{1}{z_t K_s(y_o - y)},$$

or more simply as,

$$a_{lim} = -\frac{1}{2z_t K_s Re[G(\omega)]}. \tag{4.76}$$

The limiting depth of cut for a cutter with nonuniform pitch is derived in a similar way. A cutter where the tooth pitch varies linearly is used here for simplicity. The geometry of a cutter with linearly increasing pitch and three teeth in cut is shown in Figure 4.26(b). Unlike the cutter with uniform pitch, the phase shifts for the three teeth are not equal. While the present vibrations (y_1, y_2, and y_3) are equal and in phase, the

teeth encounters the previous modulations (y_{o1}, y_{o2}, and y_{o3}) at different phase shifts (ε_1, ε_2, and ε_3) due to the nonuniform pitch. At the limit of stability, the magnitudes of the present and previous vibrations are still constant and equal,

$$|y| = |y_o|.$$

However, the vectors (y_{oi}) are no longer in phase with each other; the vector (y_{o2}) is inclined by a phase shift difference of ($\Delta\varepsilon$) with respect to (y_{o1}), while the vector (y_{o3}) is inclined by a phase shift difference of ($2\Delta\varepsilon$) with respect to (y_{o1}). The corresponding phase plane plot is shown in Figure 4.28. The sum of the vectors (y_{o1}, y_{o2}, and y_{o3}) is labelled (y_e). Due to the changes in phase shift, (y_e) is shorter here than in the uniform tooth pitch case (see Figure 4.27),

$$|y_e| = |y_{o1} + y_{o2} + y_{o3}| < 3|y_o|.$$

Consequently, the vector ($\sum y_{oi} - \sum y_i$) is shorter as well. Since the limiting depth of cut is inversely proportional to ($\sum y_{oi} - \sum y_i$), cutters with nonuniform tooth pitch always increase stability. The limiting depth is given by,

$$a_{lim} = -\frac{1}{K_s(\sum y_{oi} - \sum y_i)},$$

or subsequently,

$$a_{lim} = -\frac{1}{qz_tK_sRe[G(\omega)]}. \qquad (4.77)$$

where $q < 2$.

Analyses using this simplified approach by Tlusty [6] and Vanherck [20] yield large increases in stability. Increase in the chatter limit over a wide continuous range of spindle speed can be as large as 600%, and increases in a narrow spindle speed range can be upwards of 3000%. A typical stability lobe curve for a cutter with linearly varying

tooth pitch is shown in Figure 4.29. A curve for a similar cutter with uniform tooth pitch is shown for comparison in the same diagram. The curves are generated using the orthogonal modes listed in Table 4.1. The uniform tooth pitch stability curve assumes there are four teeth in cut simultaneously. The nonuniform tooth pitch curve also assumes there are four teeth in cut, but with the linearly increasing tooth pitch variation of 1.0 degree which results in tooth pitches of 88.5 degree, 89.5 degree, 90.5 degree, and 91.5 degree. The uniform tooth pitch curve is similar to stability lobe curves shown in earlier figures. The critical limit of stability is at 0.137 [mm] with stable peaks of up to 0.28 [mm] (at approximately 2500 [rpm]). The nonuniform pitch cutter is always more stable than the uniform pitch cutter. The small tooth pitch variation provides sufficient phase shift variation between teeth for shorter waves at low spindle speeds. The chatter free cutting depth below 1000 [rpm] for the nonuniform tooth pitch cutter is at least 100% larger than the uniform tooth pitch critical limit of stability. At even lower speeds, the increase in stable cutting depth is in excess of 250%. However, the small pitch variation does not disturb wave regeneration sufficiently for longer waves at high spindle speeds. Above 1500 [rpm], the increase in the limiting depth of cut from tooth pitch variation is limited to only about 15%. As pointed out by Tlusty and Vanherck, large pitch variations are most efficient at high speeds while small pitch variations are most efficient at low speeds.

Anyhow, the results of this simplified analysis strongly over estimates the increase of chatter free depth of cut. Analyses using time domain simulations by Tlusty [6] and Fu [2] show more modest increases in stability. A brief example of stability increase from cutters with linearly varying tooth pitch using time domain simulations is given below.

The dynamic system used in the following simulations is the same one used for variable speed simulations, and is listed in Table 4.1. To decrease the demand on computing time, however, the simulations in this section are all generated without process damping. The results are of half immersion up milling at 600 [rpm]. A number of linear pitch variations

were simulated in an attempt to find an optimal variation. Figure 4.30 plots the peak-to-peak (X) deflection versus depth of cut for different linear tooth pitch variations. The 0.25 degree tooth pitch variation results in teeth that are 89.625 degree, 89.875 degree, 90.125 degree, and 90.375 degree apart. The 0.5 degree tooth pitch variation yields pitch angles of 89.25 degree, 89.75 degree, 90.25 degree, and 90.75 degree. The 1.0 degree tooth pitch variation results in pitches of 88.5 degree, 89.5 degree, 90.25 degree and 91.5 degree ,and the 2.5 degree tooth pitch variation gives pitch angles of 86.25 degree, 88.75 degree, 91.25 degree and 93.75 degree. Figure 4.30 shows that the limiting depth of cut is largest for a linearly varying pitch of 0.5 degree. Sharp increases in peak-to-peak deflection with cutting depth does not occur until 0.625 [mm] for the 0.5 degree tooth pitch variation. On the other hand, instability sets in at 0.5 [mm] and 0.56 [mm] for the other three tooth pitch variations.

Figure 4.31 compares simulation results of fixed speed cuts with uniform and nonuniform tooth pitch cutters, and variable speed cuts with uniform tooth pitch cutters. The nominal speed is again 600 [rpm] for these half immersion up milling simulations. The fixed speed nonuniform pitch curve has a linear variation of 0.5 degree, which was shown earlier in Figure 4.30. Two curves from variable speed simulations for a cutter with uniform tooth pitch are also shown. One curve is with variation amplitude of 50 [rpm] and frequency of 2.5 [Hz], while the other is for variation amplitude of 100 [rpm] and frequency of 5.0 [Hz]. Figure 4.31 shows a chatter limit of 0.5 [mm] for the fixed speed uniform tooth pitch curve. Stability against chatter is increased about 25% by the cutter with nonuniform tooth pitch. The limiting depth of cut for that curve is at 0.625 [mm]. The stability of the uniform pitch cutter with variation amplitude of 50 [rpm] and frequency of 2.5 [Hz] compares favorably with the stability of the nonuniform tooth pitch cutter. The transition to instability is not as sharp in this case, but the chatter limit appears to be at approximately 0.625 [mm] as well. The greatest stability is a result of

the uniform tooth pitch cutter with speed variation of 100 [rpm] and 5.0 [Hz]. The gentle rise in deflection magnitude with cutting depth indicates that stability is present up to at least 0.98 [mm]. Thus this particular case shows that greater stability can be obtained by using variable spindle speed than by cutters with nonuniform tooth pitch.

This section contained a brief discussion of nonuniform pitch cutters, and their improvement of cutting stability. Although frequency domain analysis predict large improvements in stability, Tlusty and Fu have shown that increases in the chatter limit are much smaller. Comparison of fixed speed nonuniform tooth pitch stability with variable speed stability shows that variable speed cutting is at least as efficient as nonuniform tooth pitch cutting.

4.4 Experimental Set-Up

The experiments presented in this chapter and the following chapter were all performed with the same basic set of equipment. The experimental set-up is shown in Figure 4.32.

All cutting tests were performed on a vertical milling machine which was retrofitted with Pulse-Width-Modulated permanent magnet DC motors for each of the three axes and a complementary three axis servo motor controller. Motion control of the axes can be accomplished from the PC through a PC bus to Multibus adapter, but for the following experiments the servo motor controller interfaces with a VT-100 terminal via a RS-232 serial link. Thus the PC is left only to manipulate the spindle speed, details of which will be given later.

Dynamic milling signals can be obtained from three sources: a table mounted three axis piezoelectric dynamometer, a standard audio microphone, and a noncontact proximity sensor. The workpieces are fixed to the Kistler Model 9257a dynamometer; thus

cutting forces in the three perpendicular directions can be measured. The natural frequency of the unit is claimed to be upwards of 4000 [Hz]. However, with the added workpiece and clamping mass used in the experiments, the bandwidth of the dynamometer reduces to less than 1000 [Hz]. A practical way to monitor dynamic cutting is to measure the sound pressure or noise level. A remotely positioned microphone measures the sound pressure during cutting from an unobstructed location. Another signal which is indicative of vibrational stability is the deflection of the workpiece or end mill. In fact, Rahman [12] performed a series of tests to determine the most suitable variable for chatter prediction in turning, and he concluded that the relative deflection between the workpiece and tool was the only variable (amongst deflections, forces, and noise level) that seemed "to predict rather than to indicate the onset of chatter". Since the milling table traverses as opposed to the spindle assembly, a floor mounted proximity sensor is capable of in-process measurement of end mill deflection if necessary. A steel sleeve fitted just above the cutting region on the cutter assists in the proper operation of the proximity sensor.

The spindle, once a gear selected fixed speed drive, is fitted with a variable speed controller. Once the appropriate code is downloaded to the controller's processor, speed control is made from the PC through the serial port. Due to the high inertia of the existing spindle, speed variation is limited to between 10% and 20% of the nominal cutting frequency in variation amplitude and frequency. The maximum frequency variation for a 10% variation amplitude, and maximum amplitude for a 20% variation frequency are given in Figures 4.33(a) and 4.33(b), respectively.

The other main constituent of the system is an IBM-compatible PC. It is responsible for data acquisition of the force, sound, or deflection signal, their subsequent signal processing, and finally for sending the control signal to the variable speed drive. An 8088 central processing unit based computer alone is incapable of performing all these tasks,

nor can it perform them at nearly the required speed. The computer is fitted with an 8087 math coprocessor, a Dalanco Spry TI-TMS320C25 based digital signal processing board, as well as a Data Translation general purpose DT-2801-A data acquisition board. In addition to performing the dedicated signal processing, the signal processing board also includes an 153 Khz analog to digital convertor and digital to analog convertor. Therefore, both data acquisition and signal processing can be accomplished by the single board resident in the PC bus. However, in the event that more data acquisition channels are required, the independent data acquisition board can be put to use.

4.5 Experimental Results for Dynamic Milling

Using the general experimental set-up described in Section 4.4, a number of experiments were performed to verify the theoretical analysis of variable speed milling. Numerous cuts were necessary to establish some conclusions. The relevant results are presented in this and the following sections. Cutting tests can be separated into two categories,

- fixed speed cutting tests
- variable speed cutting tests.

4.5.1 Fixed Speed Milling Tests

The first cutting tests were straight forward cuts to determine the nature of the frequency spectrums at various stages of cutting stability. Two different end mills were used in this series of test, and their structural parameters as approximated by two orthogonal modes are given in Table 4.2. The natural frequencies are approximately 676 [Hz] and 588 [Hz] for Tool 1 and 420 [Hz] and 410 [Hz] for Tool 2, with a preload of 20 [lbs] applied to the end mill to minimize nonlinearities of the bearings. The next several sets of figures show

the frequency spectrum of the (X) force and (X) deflection for Tool 1. All experimental results in this chapter are from up milling or full immersion cutting tests.

Figure 4.34 shows air cutting data at 600 [rpm]. The deflection (Figure 4.34(b)) exhibits strong spindle period modulation due to runout of the end mill sleeve, and that results in a very large low frequency content. Since the force shown is actually the force on the workpiece rather than on the cutter, air cutting has no effect on the signal (Figure 4.34(a)). The content of the frequency spectrum is a consequence of electrical noise and offsets inherent in the dynamometer set-up. As a result of the poor deflection signal, an analog highpass filter was added to the experimental set-up. All subsequent deflection signals are filtered by a 200 [Hz] 12 [db/octave] highpass filter to minimize the low frequency signal. Air cutting signal of the filtered deflection is shown in Figure 4.35. Notice the drop in magnitude in the low frequency range, and also the prominence of the vibration signal near the end mill natural frequencies (in the 500 [Hz] to 700 [Hz] range).

Figure 4.36 shows the results of a half immersion up milling cut at 600 [rpm]. The cutting depth is 1.27 [mm] and the feed per tooth is 0.05 [mm]. The force spectrum in Figure 4.36(a) shows practically no evidence of vibratory cutting. On the other hand, the (X) deflection spectrum in Figure 4.36(b) clearly shows a peak between 500 [Hz] and 700 [Hz]. Visual inspection of the cut surface revealed no noticeable chatter marks, and this fact was further reinforced by the lack of any loud pitch noise during cutting.

Figure 4.37 shows the spectrums when the depth is increased to 2.54 [mm] and all other parameters are kept the same. There was some evidence of chatter during cutting from minor vibration marks on the finished surface. The force signal shows signs of stronger high frequency vibration than the previous cut, although it is at nearly 1100 [Hz] (see Figure 4.37(a)), a point that will be taken up later. The deflection spectrum has the same form as the previous cut except the high frequency signal around 550 [Hz] has a larger amplitude (see Figure 4.37(b)). The presence of tooth frequency harmonics

is also increasing, although the low frequency harmonics are attenuated by the highpass filter.

The spectrums in Figure 4.38 are for a cut with the depth increased to 5.08 [mm]. Chatter was apparent during cutting from both, surface finish and cutting noise. The (X) deflection spectrum has very strong tooth frequency and chatter frequency components (see Figure 4.38(b)). The (X) force has a similar appearance as the last cut, but with further amplification at approximately 1100 [Hz] (see Figure 4.38(a)). The high frequency signal from the dynamometer makes little sense in comparison with the high frequency deflection signal, which has peaks close to the end mill frequencies and is as expected. More cutting tests, not shown in the thesis, suggest that the cutting force signal is not always indicative of the actual cutting force. The force spectrums for certain cutter-workpiece combinations are instead indicative of the dynamics of the dynamometer. A measurement of the dynamometer transfer function with all the mounting hardware and workpiece confirms this fact (see Figure 4.39). With a total experimental set-up mass of 3.5 [Kg], the first natural frequency of the dynamometer in the (X) and (Y) directions are 948 [Hz] and 996 [Hz], respectively. Therefore, the cutting forces are not reliably measured by the dynamometer when the frequency is above the dynamometer's bandwidth (i.e. approximately 600 [Hz]).

Having established that the dynamometer does not give accurate measurements of the cutting force in this case, deflection measurements are used and recorded for all subsequent cuts. The force signals, however, still serve as a good tool for the comparison of relative stability between cuts. In addition, the reliability of the force signal can usually be determined from comparison with the deflection or sound pressure signal. The nature of cutting stability as observed from the experiments is examined next. Various stages of cutting stability will be examined through time history data.

First, the time history of the (X) force and filtered (X) deflection for a stable full

immersion cut using Tool 2 is shown in Figure 4.40. The spindle speed is 600 [rpm], the axial depth is 1.27 [mm], and the feed per tooth is 0.1 [mm]. With a four tooth cutter in full immersion, the cutting forces are supposed to be constant. The modulation of forces shown in Figure 4.40(a) are attributed to runout of the end mill. The minimal dynamics of this cut can be observed in the (X) force. All dynamics damp out before the next tooth engages in cut, where the entry and exit impacts from cutter runout result in some transient force. The content of the deflection signal in Figure 4.40(b) is primarily noise as opposed to vibrations.

The next plots in Figure 4.41 are from a cut with the depth increased to 2.54 [mm]. Not much has changed, except the magnitudes of the force signals. The increased chip load increases the cutting force for all teeth (see Figure 4.41(a)) and marginally increases vibration (see Figure 4.41(b)).

Figure 4.42 shows the time histories when the depth is increased to 3.81 [mm]. The corresponding forces (Figure 4.42(a)) and deflections (Figure 4.42(b)) increased greatly from the previous cut shown in Figure 4.41. The peak-to-peak forces increased from 2 to 3 times. The vibration also increased by a similar amount over the previous cut. In addition, the noise present during cutting was very loud and disturbing. This cut, shown in Figure 4.42, can be considered unstable, but there are a couple of peculiarities associated with this cut which are absent from theoretical basis. Figure 4.42(a) shows that although the forces are very large in amplitude, they seem to be primarily a consequence of cutter teeth impact rather than wave regeneration. Away from the impact transients, the amplitudes are not nearly as large and there is actually minimal dynamic forces. The two plots in Figure 4.42 show unstable cutting forces and deflections, but not of a form which is totally consistent with unstable cutting predicted by classical theory and time simulations.

The problem is partially related to the large influence of runout. The transient spikes

during entry and exit are much larger than expected. This is due to the large runout on the cutter which was measured to be of the same order as the feed per tooth. The measured runout was 0.07 [mm], 0.04 [mm], and 0.10 [mm] with respect to the first tooth. The large amplitude vibrations result in strong flank interference. The consequent damping forces due to flank interference is considered and accounted for in the simulation analysis. The extent of interference between the tooth and the cut surface, though, was not anticipated. When rubbing does not exceed beyond the primary flank face, ploughing forces are approximated in the simulation. However, in light of the large transients in the force and deflection histories, it seemed likely that there was actually more ploughing than expected. Inspection of the end mill and finished surface after cutting verified these observations. It was obvious that rubbing extended beyond the primary flank face and well into the secondary flank face. The extent of interference not only induces large ploughing force magnitude, it also changes the resultant force direction. Ploughing forces from the secondary flank face are at a different direction than the cutting forces and primary flank face ploughing forces. The relative magnitude of these forces determine the final orientation of the resultant force direction.

The most influential aspect of these large ploughing forces, however, is the alteration of the freshly cut surface by the tooth geometry. If interference is within the primary flank face, changes in the cut surface should be negligible, since the primary flank face length for the cutter is approximately 0.7 [mm]. The simulation assumed that the surface is formed by the shearing process and that ploughing only contributes damping forces. However, the amount of interference visible on the tool and on the finished surface indicates that the profile of the surface is instead almost entirely formed by interference between the tool flank and workpiece. The surface waviness necessary for wave regeneration chatter is largely absent. Thus in addition to generating damping during a tooth pass, rubbing also alters the modulation of the cut surface which disturbs wave regeneration. The results

shown in Figure 4.42 for the cutting depth of 3.81 [mm] would have been much more unstable if the amount of process damping was decreased.

In an attempt to decrease the amount of flank interference, a similar cut was performed but with the spindle speed doubled; increased speed decreases the amount of flank interference. The immersion and feed per tooth are the same, but the speed is now 1200 [rpm]. In this case, an axial depth of only 2.54 [mm] precipitated instability. A comparison between the (X) force histories for the 600 [rpm] cut in Figure 4.42(a) and the 1200 [rpm] cut in Figure 4.43(a) reveals a significant difference. Despite the fact that maximum peak-to-peak forces are about the same magnitude for both cases, the dynamic forces between transient spikes are much more prominent in the 1200 [rpm] case. Force fluctuations (between transient spikes) are approximately 400 [N] in Figure 4.43(a), while there are practically no dynamic force fluctuations between spikes in Figure 4.42(a). The filtered (X) deflections show the same difference. For the 600 [rpm] cut (Figure 4.42(b)), deflections away from the spikes are minimal. The deflection signal fluctuates mainly between –0.03 [mm] and +0.02 [mm], and is primarily a result of noise. In the 1200 [rpm] case (Figure 4.43(b)), vibrations away from spikes fluctuate between –0.05 [mm] and 0.05 [mm], and are a result of wave regeneration. It is apparent that at this speed, there is less process damping and self-excited chatter is present.

Additional tests reveal many attributes of dynamic milling found from simulation results in Section 4.2.1. The dependency of stability on the axial depth and width of cut is illustrated in Figure 4.44. Figure 4.44 shows the filtered peak-to-peak (X) deflection for 60 degree and 90 degree immersion up milling cuts made with Tool 1. The spindle speed is 600 [rpm] and the feed per tooth is 0.10 [mm]. With the half immersion cut (b = 12.7 [mm]), dynamic deflection rises sharply with axial depth at 3.8 [mm]. The 60 degree immersion cut (b = 6.35 [mm]) taken with the same cutting conditions is also shown in Figure 4.44. The limiting depth rises to approximately 7.6 [mm] for this cut since the

width of cut is reduced by 50%. Note that milling tests were usually not carried out for axial depths too much larger than the chatter limit to avoid cutter damage. Thus Figure 4.44 and subsequent plots rarely show more than two data points in the unstable region.

The next three plots show the influence of process damping in a number of ways. Figure 4.45 and Figure 4.46 illustrate the cutting speed dependency of flank interference. Figure 4.45 compares the peak-to-peak (X) deflection of similar cuts at two spindle speeds, 600 [rpm] and 1200 [rpm]. Tool 1 is used for these half immersion up milling cuts. In the 600 [rpm] case, the chatter limit seems to be at about 3.8 [mm]. Deflection amplitudes are almost constant from 1.3 [mm] to 3.8 [mm]. It is only at 5.1 [mm] that the deflection almost doubles in magnitude. The number of experimental data points is scarce for the 1200 [rpm] case. Nevertheless, it seems probable that the chatter limit is at 3.8 [mm]. Even though both cuts seem to be unstable at 5.1 [mm], the difference in magnitude at that depth is more than three times, from 0.35 [mm] to 1.2 [mm], because there is less process damping at 1200 [rpm].

Figure 4.46 shows the peak-to-peak (X) force at various spindle speeds for up milling cuts with Tool 1. The depth of cut is 3.81 [mm] and the immersion angle is 112 degrees. The general trend of decreasing stability with increasing speed is evident here as it was with simulations. The force more than doubles as the spindle speed changes from 600 [rpm] to 1200 [rpm] (with the feed per tooth kept constant).

Tool wear, as pointed out earlier, is an important factor in cutting stability. Figure 4.47 shows the difference in peak-to-peak (X) deflection for full immersion cuts at 1200 [rpm] with the same tool but with different wear. The "new tool" was used for about a minute before the shown cuts were made, while the "worn tool" was through approximately ten minutes of fairly vibratory cutting before the shown cuts were made. Figure 4.47 clearly shows the limiting depth of cut for the worn tool case at 1.3 [mm]. Although it is not possible to determine the chatter limit for the new tool case, it is most likely

less than 1.0 [mm]. The sharp increase in dynamic deflections from depths of 1.0 [mm] to 1.5 [mm] imply vibrational instability. In addition, the vibration at 1.0 [mm] for the new tool case differs significantly from the stable worn tool case, which suggests a difference in stability. Thus tool wear in this case increased the chatter limit by at least 30%, from 1.0 [mm] to 1.3 [mm]. In addition, the peak-to-peak deflections for the new tool are also from 60% to 230% larger than the worn tool case.

In summary, the fixed speed cutting tests exhibited all the trends that were expected and predicted from simulations. The importance of depth of cut, width of cut, flank interference, and tool wear were all apparent from the test results. In addition, the problems of flank interference and cutter runout as they relate to correlation with theoretical results were also emphasized. Despite all these similar trends, the comparison of chatter limits, force, and deflection values between simulation and experiment is still not good. The main problem in matching simulation and experimental results lies in process damping. Flank interference has a stronger stabilizing influence than the simulation can model. Visual inspection of the cutter indicates that interference with the newly cut surface can extend beyond the primary flank face. As mentioned earlier in the section, the extent of this interference changes the shape of the cut surface, something that the simulation program is not capable of duplicating. The cut surface carved by the flank faces must be accounted for in order to achieve better prediction of chatter limits, forces, and deflections.

4.5.2 Variable Speed Milling Test

The next phase of experimental testing is controlled variable speed milling. These results determine if trends observed in simulation results are evident in actual cutting. Before presenting the relevant results in graphical form, it is worthwhile to elaborate on the cutting results in a general manner. While successful data were obtained, they were not

obtained without problems. A few difficulties had to be overcome before satisfactory results were achieved with regularity. The first difficulty was the extreme influence of flank interference on stability and chatter. Process damping is well known as a major factor in cutting dynamics, but its importance was further amplified here due to the limitation of the available experimental set-up.

After increasing the spindle speed to decrease process damping, it was typically necessary to determine the depth of cut at which the process became unstable. Variable speed stability is most successful near the stability limit. Problems that surfaced at this stage were tool wear and the torque limit of the variable speed spindle. Tool wear easily increased the limiting depth of cut by 30% to 50%, which made identification of the limit difficult. The torque limit of the spindle often prevented chatter from developing because of the corresponding limit on axial depth of cut. The chatter limit could be decreased by increasing the speed, but at the same the time torque limit decreased. Therefore, it became a tedious task to find the chatter limit for a given immersion angle.

Once the fixed speed limiting depth was determined, identical cuts with speed oscillations were performed in order to observe any differences in stability. Success of speed oscillation, as discussed before, depends on various parameters. Without isolation of some of these parameters, reasonable conclusions must be drawn from results that include these influences. Flank interference, stability of the original process, mode coupling, and variation amplitude and frequency all play important roles in variable cutting speed effectiveness. The potential of speed oscillations can be properly evaluated only if the contribution of these influences are minimized.

The first set of results are from milling tests using Tool 2 (as specified in Table 4.2). Figures 4.48 to 4.50 show the time histories of the filtered (X) deflection, (X) force, and (Y) force for two cuts, one with fixed speed and the other with variable speed. The cuts are 104 degree immersion cuts taken at a nominal speed of 1200 [rpm] and a cutting

depth of 1.27 [mm]. The speed variation imposed for all experimental results presented in this chapter is 50 [rpm] amplitude and 2.5 [Hz] frequency, unless otherwise specified. Figure 4.48 compares the highpass filtered (X) cutter deflection for the two cuts. Figure 4.48(a) shows that the fixed speed cut is very vibratory, with peak-to-peak deflections of approximately 0.30 [mm]. The variable speed cut produced the deflection time history shown in Figure 4.48(b). Although there are still transient spikes, their amplitudes are reduced to about 0.15 [mm] peak-to-peak. Comparison of the (X) force in Figure 4.49 gives a better indication of variable speed stabilization. The peak-to-peak amplitudes for the variable speed case in Figure 4.49(b) are about 1000 [N] as opposed to approximately 2000 [N] for the fixed speed cut shown in Figure 4.49(a). Figure 4.50(a) and Figure 4.50(b) show the (Y) force for the fixed speed and variable speed cut, respectively. A reduction in peak-to-peak amplitudes and dynamic forces can be seen as with the (X) force comparison.

Additional experiments show other similarities to simulation results. Figure 4.51 shows Tool 2 milling tests at 1200 [rpm]. Full immersion cuts with fixed and variable speed show variable cutting speed increasing the chatter limit, as described in Section 4.2.2. The fixed speed curve in Figure 4.51 shows a gradual increase of peak-to-peak force with cutting depth from 0.64 [mm] to 1.9 [mm], where forces change from 400 [N] to 1800 [N]. However, the force increases 1400 [N] from 1.9 [mm] to 2.2 [mm], indicating cutting instability. The variable speed curve exhibits the same shape, but the chatter limit is increased to 2.2 [mm] from 1.9 [mm] for the fixed speed cut. A similar comparison from simulation was shown in Figure 4.19. Here again, the largest improvement in stability occurs at the axial depths when the fixed speed cut is close to critical stability (i.e. Region 2 in Figure 4.19). In Figure 4.51, the variable speed peak-to-peak force is almost 50% less than the fixed speed force in this region.

Although not enough data are available to generate reasonable plots, experimental

results do indicate that variable speed stabilized cutting of various immersion angles. The distinction in stabilization between low and high immersion cuts found from simulation is not evident here. The degree of mode coupling activity is very sensitive to the structural dynamics of the system. Thus discrepancies in the natural frequencies and modal direction between simulation and experimental conditions may result in different mode coupling characteristics. Table 4.3 contains (X) force, (Y) force, and (X) deflection comparisons of fixed and variable speed tests showing improvements in about 70% of the cuts.

Due to the limitation of the variable spindle drive, not enough data were collected to analyze different speed variations, but the next figure indicates the importance of large variations. Figure 4.52 shows the (X) deflections from Tool 1 cuts for fixed and variable cutting speed. The axial depth of cut is 3.8 [mm] and the immersion angle is 112 degrees. The frequency and amplitude of variation remained constant at 50 [rpm] and 2.5 [Hz] in all cuts; thus the percentage variation relative to the nominal speed is decreasing as the speed is increasing. The plot establishes a general trend of diminishing stabilization as the relative variation decreases, especially at a nominal speed of 1200 [rpm] when the variable speed deflection is actually slightly larger than the fixed speed deflection. The reduction in peak-to-peak deflection calculated from Figure 4.52 is approximately 48% at 600 [rpm], 46% at 900 [rpm], 32% at 1000 [rpm], 26% at 1100 [rpm], and –6% at 1200 [rpm]. Thus the stabilization brought upon by larger variations is reflected in vibration reduction.

After the experience gained from the first series of test, a second series was performed in an attempt to judge variable speed milling under more controlled conditions. A flexible workpiece and rigid cutter was used this time instead of a rigid workpiece and flexible cutter. A workpiece of 7075-T6 aluminum was machined so that it essentially represented a single degree of freedom system. The workpiece, shown in Figure 4.53, prohibited mode

coupling from developing. A four tooth face mill with a diameter of 50.0 [mm] served as the rigid cutter. The large diameter of the face mill compared with the previous 25.4 [mm] end mill doubled the cutting speed, thus reducing process damping. In addition, it was possible to partially minimize the runout by adjusting the inserts. The experimental parameters for the following tests are listed under Tool 3 in Table 4.2.

This experimental set-up eliminated variables external to the wave regeneration process so that the full potential of variable speed chatter suppression could be assessed. Mode coupling should have been nonexistent since it requires at least two degrees of freedom; flank interference was decreased by using a large diameter cutter; and runout was minimized as much as possible. The results achieved were, in fact, much more favorable to variable cutting speed. Chatter suppression was more consistent and with greater effect.

The workpiece was mounted on the table dynamometer for force measurements as well as being monitored by a proximity sensor. The large deflection of the workpiece during chatter often displaced the sensor from its proper position, leaving only reliable force signals. Fortunately, the dynamic cutting force signals had frequencies within the bandwidth of the dynamometer.

The next series of time history plots of the feeding (X) force are examples of effective chatter suppression for these tests. They were all obtained with the aforementioned set-up and parameters listed under Tool 3 in Table 4.2. Figure 4.54 compares (X) force histories between fixed and variable speed when the immersion angle is 60 degrees and the depth of cut is 6.35 [mm]. The nominal spindle speed is 900 [rpm] and the feed per tooth is 0.068 [mm]. The feed per tooth is reduced from 0.10 [mm] in order to reduce the cutting torque on the spindle. The speed variation for the variable speed case (Figure 4.54(b)) is 50 [rpm] and 2.5 [Hz]. Figure 4.54(a) shows the (X) force for the fixed speed case. Chatter is apparent from the large peak-to-peak forces, which are at least 700 [N]

for three of the four teeth. Actually, the forces are clipped due to improper setting of the data acquisition gain, but not much information is lost. The variable speed cut in Figure 4.54(b) shows considerably more stability. The disturbance of wave regeneration largely eliminates the dynamic forces. Maximum peak-to-peak force is no more than 300 [N]. The short tooth does not cut at all in this case because vibrations are not large enough to cause contact, unlike in the fixed speed cut in Figure 4.54(a).

The next two plots show cuts for half immersion up milling. The spindle speed is 900 [rpm] and the cutting depth is 3.18 [mm]. The fixed speed response in Figure 4.55(a) is very dynamic, and characteristic of chatter. Dynamic force fluctuations dominate, and it is no longer possible to distinguish the static force modulation due to the uncut chip thickness. Peak-to-peak forces are about 500 [N]. Figure 4.55(b) shows the same cut but with speed oscillations. Stability is once again present with peak-to-peak forces of no more than 150 [N]. Dynamic forces are greatly reduced and the static force modulation can be identified.

The last case is for 105 degree immersion at a nominal speed of 600 [rpm]. The axial depth of cut is 2.54 [mm]. Figure 4.56(a) shows the fixed speed (X) force history. Only two spindle revolutions are shown because of the low spindle speed. The figure shows large force fluctuations for the three teeth that do cut. Figure 4.56(b) shows the variable cutting speed (X) force history. Speed oscillation, once again, improves stability as peak-to-peak forces drop to 100 [N] from 300 [N] in the fixed speed case (in Figure 4.56(a)).

4.6 Conclusions

This chapter presented milling results from digital simulation as well as from cutting tests. Simulation results introduced the importance of flank interference, tool wear, mode

coupling, and variable speed to stability. Speed oscillation is most successful when there are large variation amplitudes and frequencies and minimal mode coupling. Experimental results showed the trends predicted from simulations. In addition, they emphasized the problems that deterred variable speed cutting from succeeding, such as runout, flank interference, and limitations of variation amplitude and frequency.

Milling Simulation Parameters					
Specific Cutting Pressure - 700 [MPa]					
Cutting Force Ratio - 0.44					
Nominal Feed per Tooth - 0.05 [mm]					
Cutter Diameter - 25.4 [mm]					
Number of Teeth - 4					
Helix Angle - 0^0					
Flank Face Length - 0.7 [mm]					
Clearance Angle - 10^0					
Flank Wearland - 0.02 [mm]					
Material Yield Strength - 500 [MPa]					
Modal Direction - -15^0					
X' - Direction			Y' - Direction		
$f_{x'}$	$K_{x'}$	$\zeta_{x'}$	$f_{y'}$	$K_{y'}$	$\zeta_{y'}$
[Hz]	[KN/m]		[Hz]	[KN/m]	
800.0	2200.0	0.04	760.0	2100.0	0.04

Table 4.1: Milling simulation parameters.

Milling Experiment Parameters				
Parameter	Cutter-Workpiece			
	Tool 1	Tool 2	Tool 3	Tool 4
Cutter Type	End Mill	End Mill	Face Mill	End Mill
Cutter Diameter [mm]	25.4	25.4	50.0	25.4
Cutter Overhang [cm]	13.2	16.5	–	6.70
Number of Teeth	4	4	4	4
Feed per Tooth [mm]	0.1	0.1	0.075	0.1
Helix Angle [^{0}]	30	30	0	30
Clearance Angle [^{0}]	10	10	11	10
Modal Direction [^{0}]	-15	-15	0	0
X' Frequency [Hz]	676	420	461	280
Y' Frequency [Hz]	588	410	–	–
X' Stiffness [KN/m]	1650	783	2274	1100
Y' Stiffness [KN/m]	1570	665	–	–
X' Damping Factor	0.021	0.017	0.003	0.003
Y' Damping Factor	0.014	0.009	–	–

Table 4.2: Milling experiment parameters.

				Fixed Speed			Variable Speed		
speed [rpm]	b [mm]	immersion [deg]	a [mm]	Fx ptp [N]	Fy ptp [N]	dx ptp [mm]	Fx ptp [N]	Fy ptp [N]	dx ptp [mm]
Tool 1:									
600	6.35	60	7.62			0.19			0.135
600	6.35	60	11.43			0.288			0.164
600	12.7	90	1.905	510		0.176	500		0.144
600	12.7	90	2.54	510		0.147	670		0.164
600	12.7	90	3.81			0.159			0.176
600	12.7	90	5.08	2450		0.326	2300		0.3
600	17.48	112	3.81	1600		0.274	1350		0.144
600	17.48	112	5.08	1760		0.141	1500		0.15
900	17.48	112	3.175	700	900	0.199	750	900	0.205
900	17.48	112	3.81	2650	1200	0.715	2700	1850	0.386
1000	17.48	112	3.81	2200	1550	0.55	1000	1020	0.375
1000	17.48	112	4.445	2475	1600	0.611	1850	2200	0.398
1000	25.4	180	3.175	1950	1100	0.692	1850	800	0.3
1100	17.48	112	3.81	2800	2250	0.478	950	850	0.352
1200	12.7	90	3.81	1700	1250	0.271	1550	1650	0.294
1200	12.7	90	4.445	3400	1250	0.444	2400	1800	0.447
1200	12.7	90	5.08	4500	2250	1.21	3800	1700	
1200	12.7	90	5.715	6000	2850	1.124	1220	1200	0.392
1200	17.48	112	2.54	1000	880	0.291	1040	760	0.32
1200	17.48	112	3.175	3100	2350	0.55	2100	2150	0.501
1200	17.48	112	3.81	3750	2200	0.816	3850	3030	0.867
1200	25.4	180	3.175	2150	1630		2250	1120	
Tool 2 :									
600	12.7	90	2.54	2200	950	0.589	1450	900	0.712
1100	19.05	120	1.27	2000	1200	0.671	1800	1000	0.622
1100	19.05	120	1.905	3700	2150	1.883	4300	2100	1.474
1200	6.35	60	4.445	1850	1450	1.924	1900	1100	1.842
1200	6.35	60	5.08	3200	1500	1.989	2850	1250	1.662
1200	12.7	90	1.905	3300	2000	1.433	3200	2000	
1200	12.7	90	2.54	4900	2400	1.637	4700	2700	0.901
1200	15.88	105	1.905	4100	2500	2.292	4300	1800	1.965
1200	15.88	105	1.27	2600	1250	1.924	650	700	0.639
1200	19.05	120	1.27	3200	1450	1.719	1200	550	0.614
1200	19.05	120	1.27	600		0.471	585		0.565
1200	19.05	120	1.905	1240		0.532	1055		0.471
1200	19.05	120	2.54	4150		0.79	765		0.487
1200	25.4	180	0.635	400		0.254	350		0.274
1200	25.4	180	1.016	770		0.319	500		0.225
1200	25.4	180	1.27	3000	1550	0.802	3300	2000	1.146
1200	25.4	180	1.27	930		0.299	750		0.34
1200	25.4	180	1.524	2700	2000	0.716	2700	1500	1.064
1200	25.4	180	1.524	1415		0.52	1150		0.491
1200	25.4	180	1.905	1775		0.622	1495		0.479
1200	25.4	180	2.159	3200			1780		
1200	25.4	180	2.54	3600		0.643	3200		0.503
1200	25.4	180	3.175	3700			3500		

Table 4.3: Variable speed milling results.

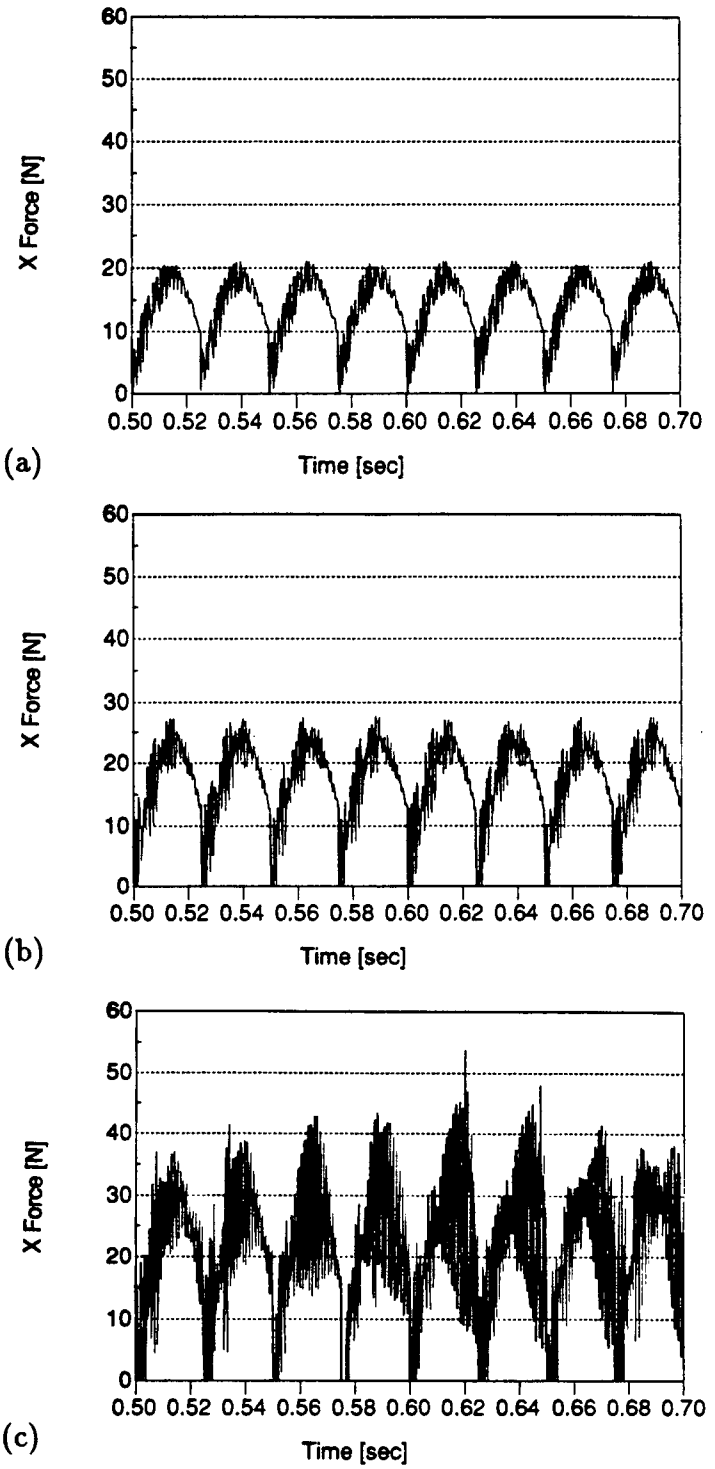


Figure 4.1: Simulated (X) cutting force for various depths of cut. n = 600 [rpm], ϕ = 90^0, a = (a) 0.625, (b) 0.781, (c) 0.977 [mm].

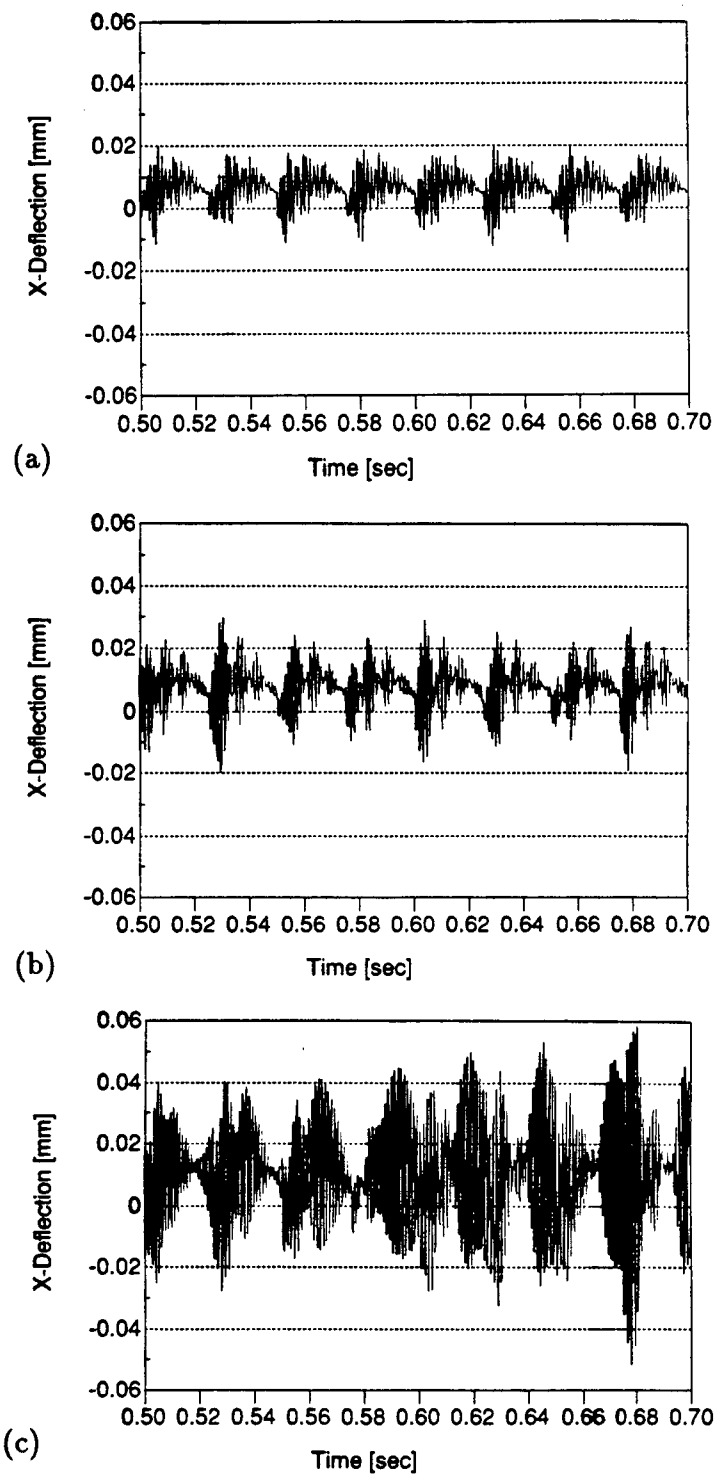


Figure 4.2: Simulated (X) cutter deflection for various depths of cut. n = 600 [rpm], ϕ = 90^0, a = (a) 0.625, (b) 0.781, (c) 0.977 [mm].

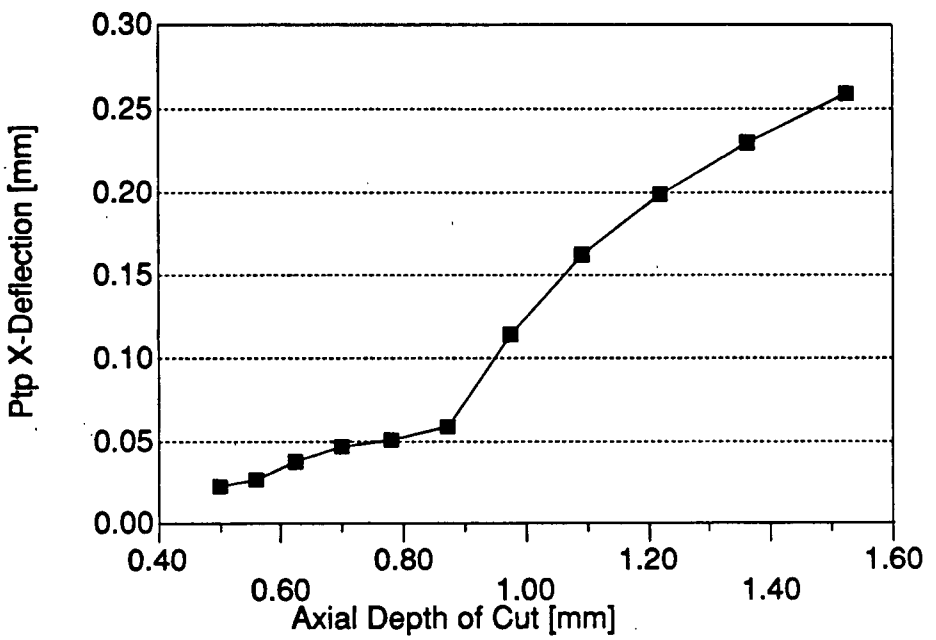


Figure 4.3: Simulated (X) cutter deflection versus axial depth of cut. n = 600 [rpm], ϕ = 90^0.

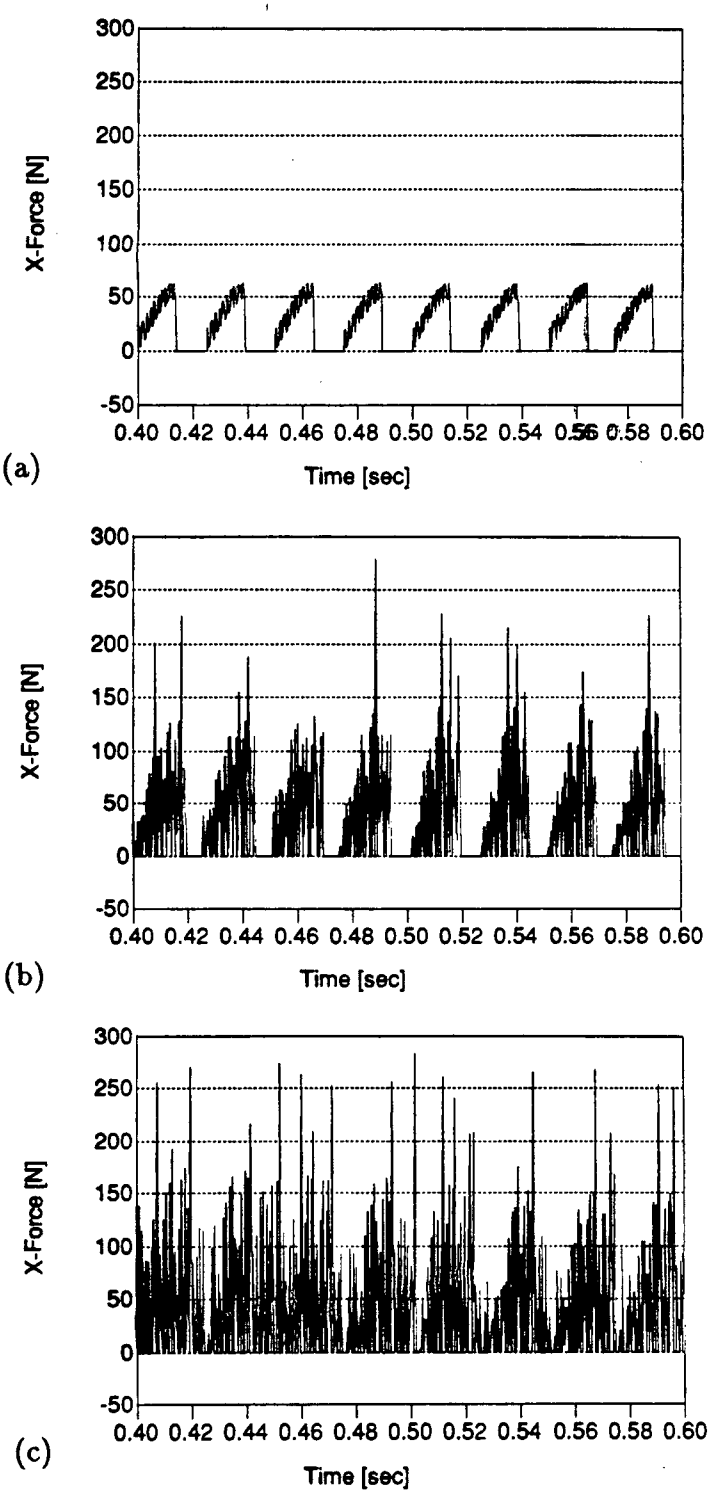


Figure 4.4: Simulated (X) cutting force for various widths of cut. $n = 600$ [rpm], $\phi =$ (a) 50, (b) 70, (c) 90^0, $a = 1.22$ [mm].

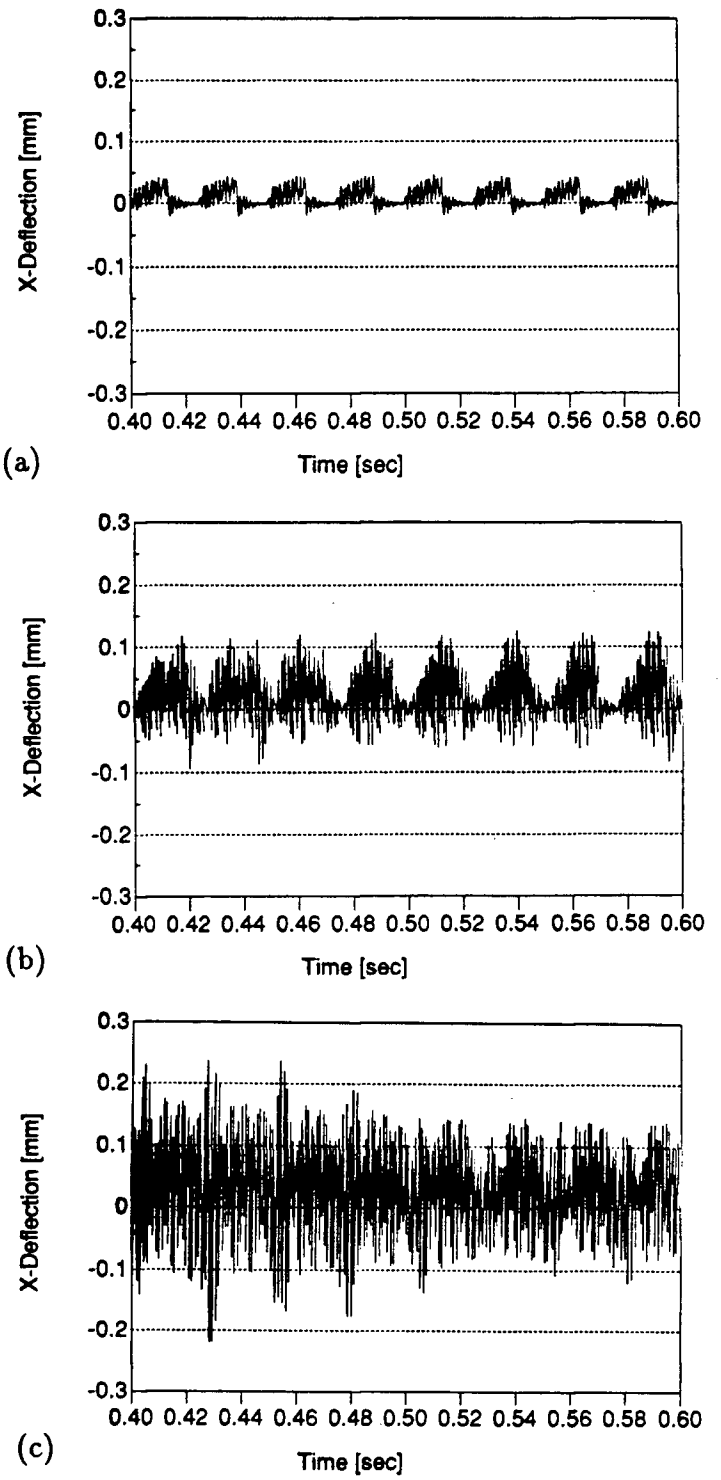


Figure 4.5: Simulated (X) cutter deflection for various widths of cut. n = 600 [rpm], ϕ = (a) 50, (b) 70, (c) 90^0, a = 1.22 [mm].

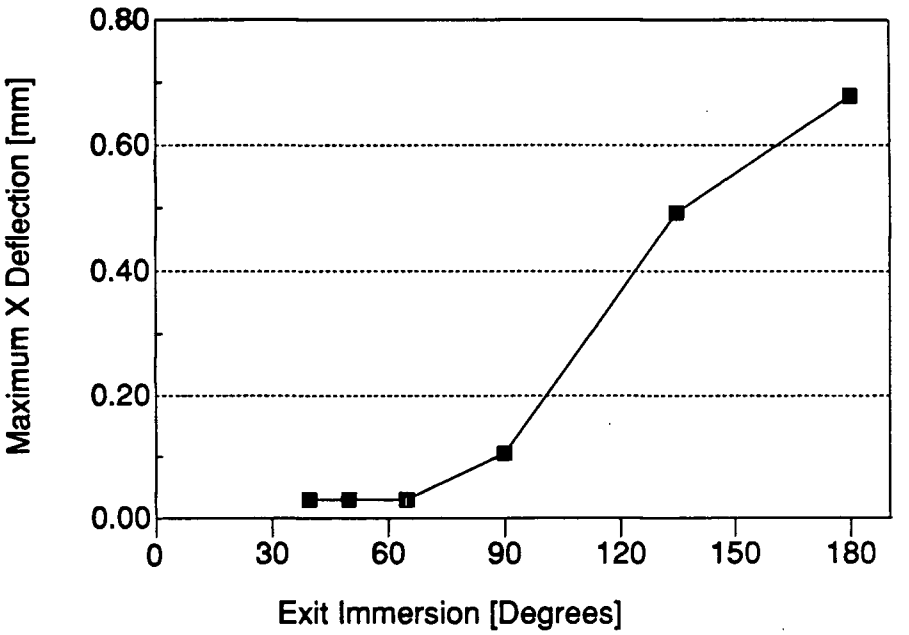


Figure 4.6: Simulated (X) cutter deflection versus exit immersion angle. $n = 600$ [rpm], $a = 1.22$ [mm].

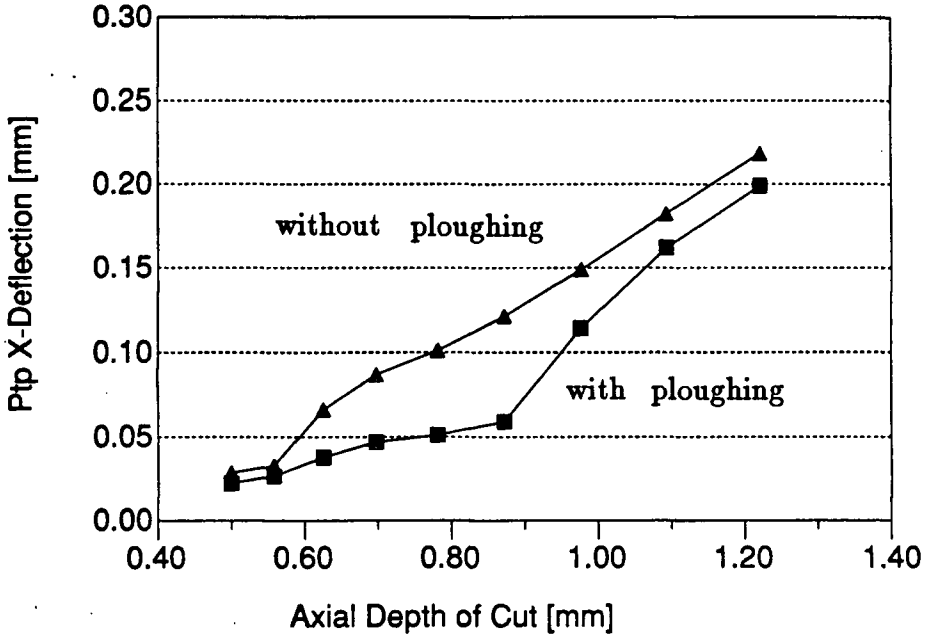


Figure 4.7: Comparison of (X) cutter deflection with and without ploughing. $n = 600$ [rpm], $\phi = 90^0$.

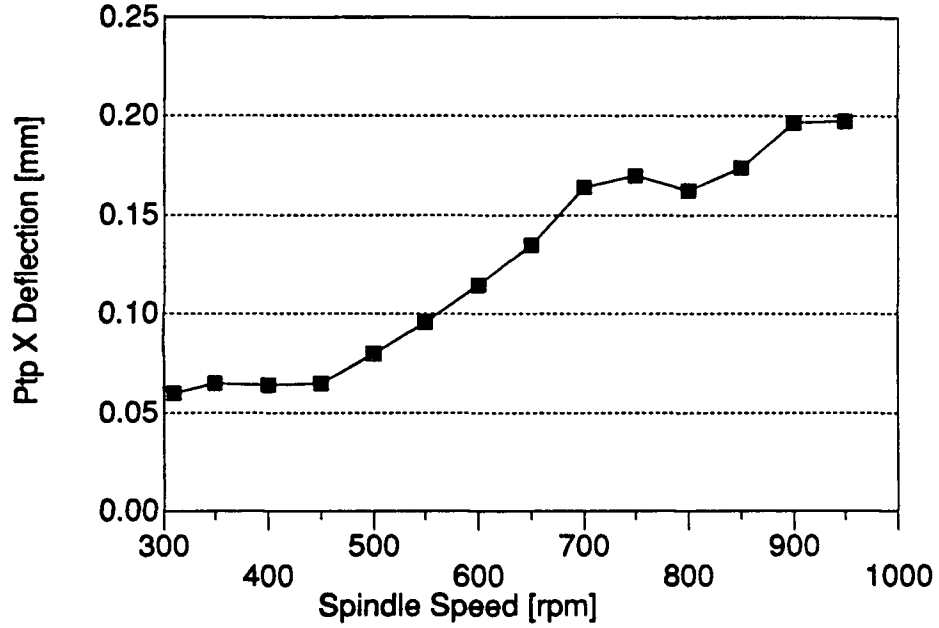
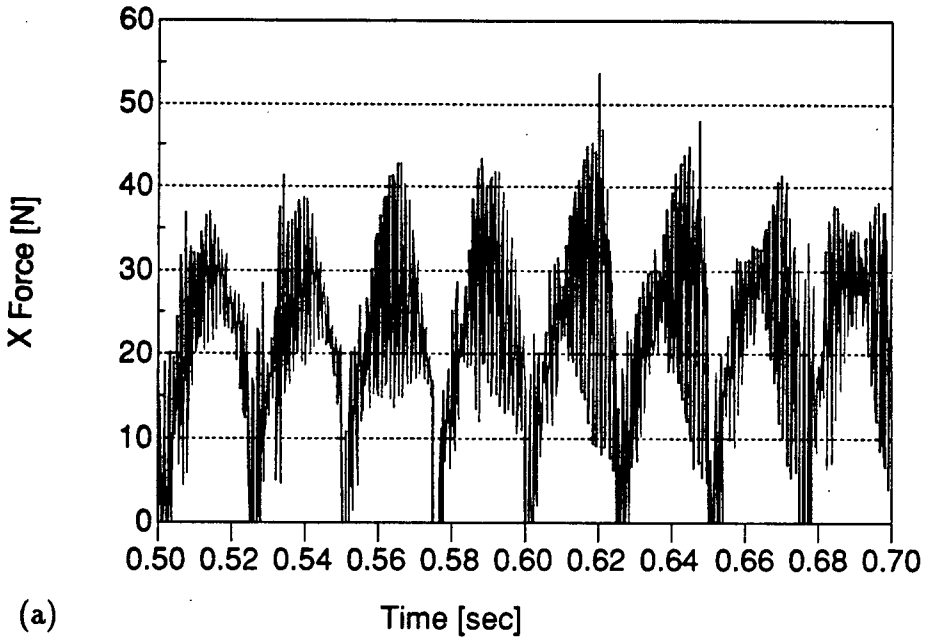
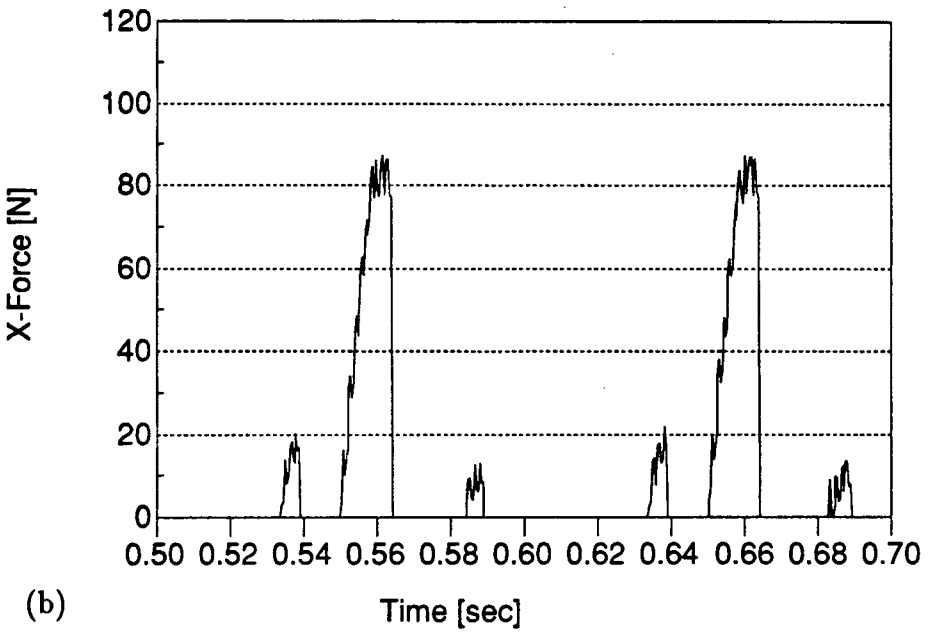


Figure 4.8: Simulated (X) cutter deflection versus spindle speed. $\phi = 90^0$, $a = 0.977$ [mm].



(a)



(b)

Figure 4.9: Comparison of simulated (X) cutting force with and without runout. n = 600 [rpm], $\phi = 90^0$, $a = 0.977$ [mm], (a) no runout (b) runout.

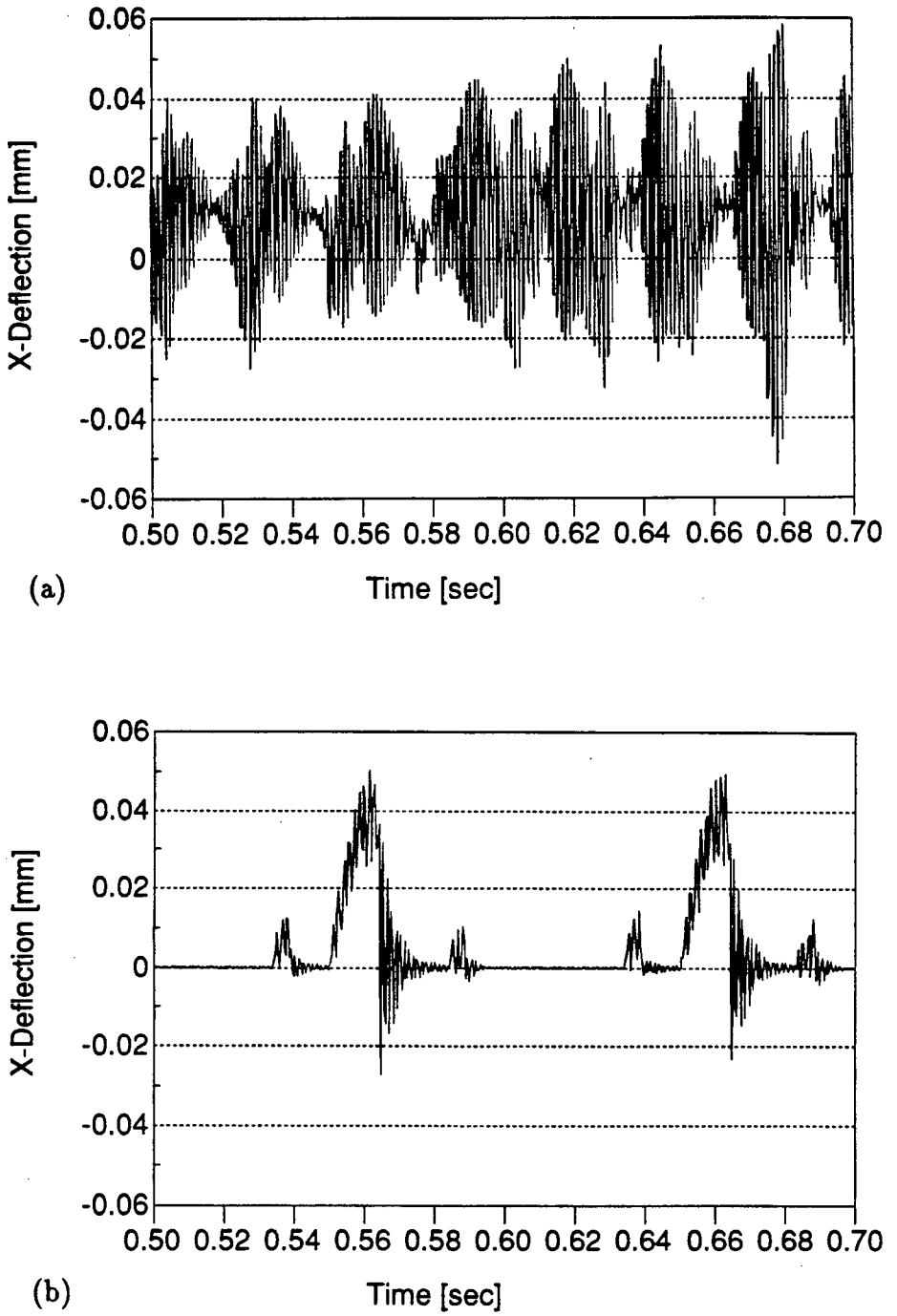
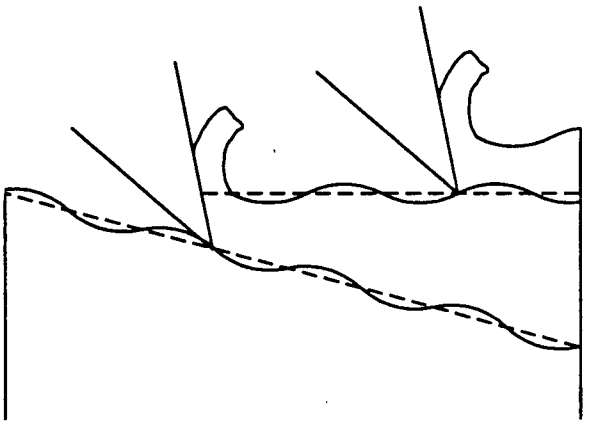
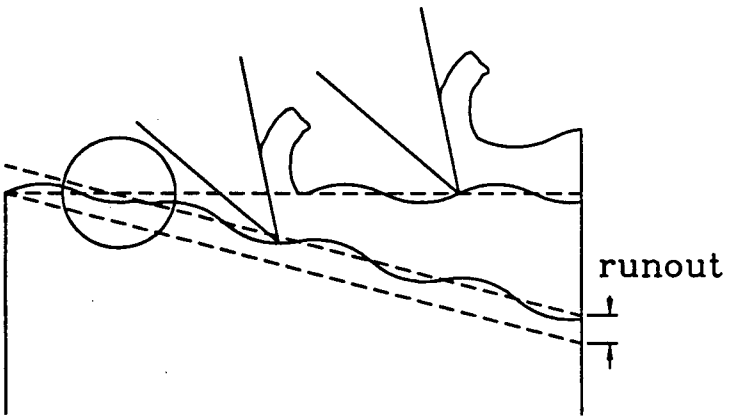


Figure 4.10: Comparison of simulated (X) cutter deflection with and without runout. n = 600 [rpm], $\phi = 90^0$, a = 0.977 [mm], (a) no runout, (b) runout.

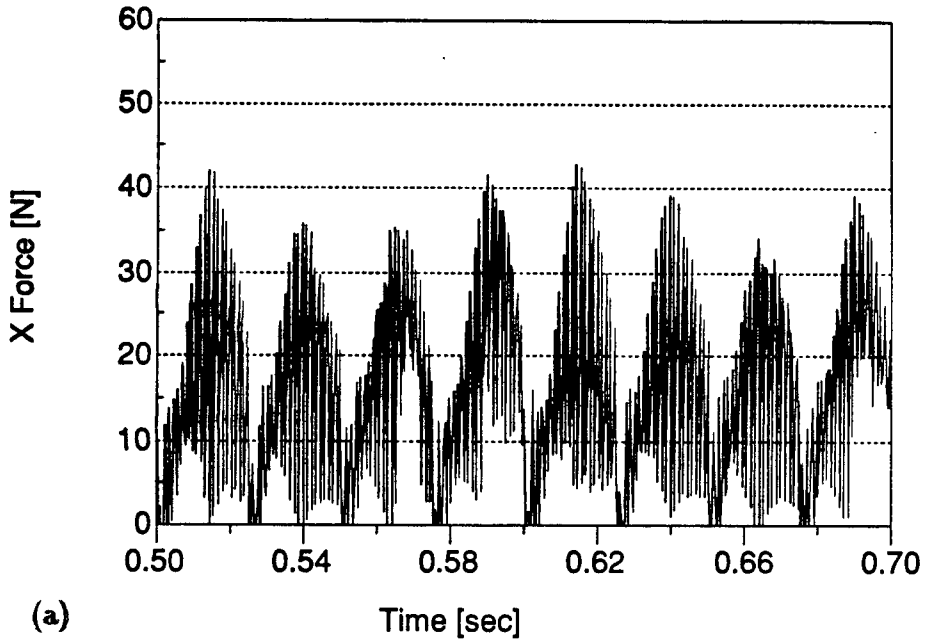


(a)



(b)

Figure 4.11: Influence of runout in milling. (a) no runout, (b) runout.



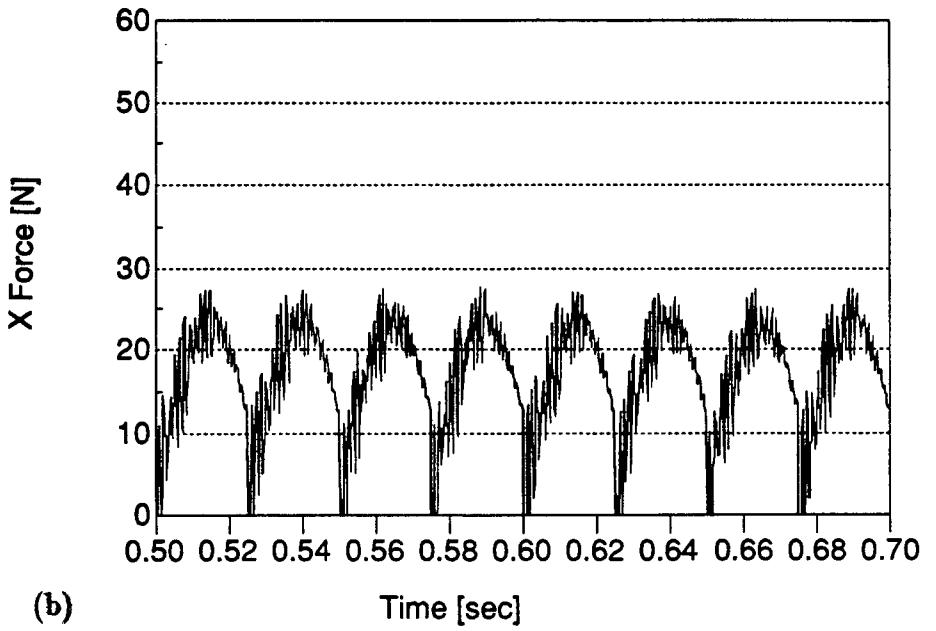
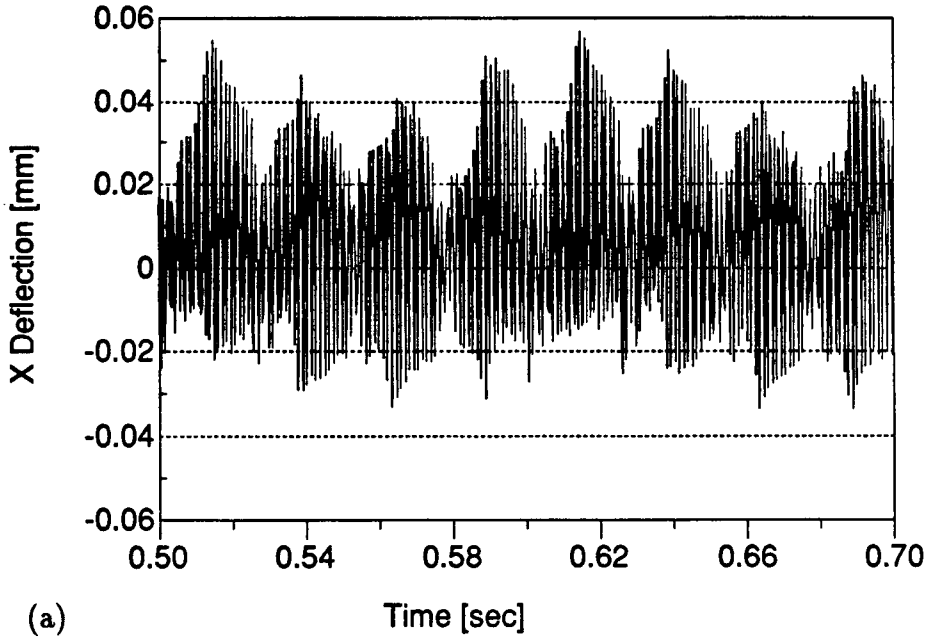


Figure 4.12: Comparison of simulated (X) cutting force with and without flank wear. n = 600 [rpm], $\phi = 90^0$, a = 0.781 [mm], (a) no wearland, (b) wearland.



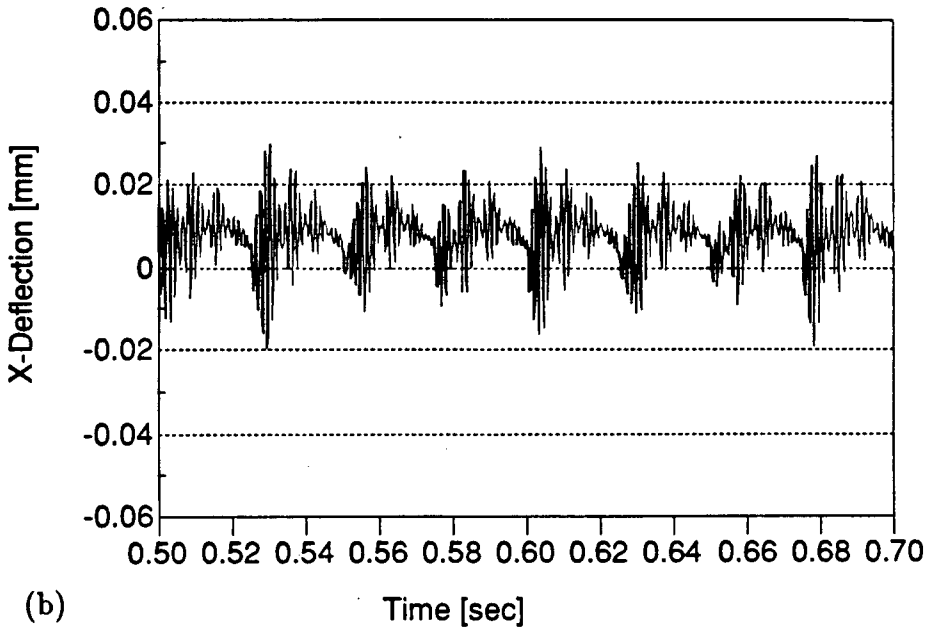


Figure 4.13: Comparison of simulated (X) cutter deflection with and without flank wear. $n = 600$ [rpm], $\phi = 90^0$, $a = 0.781$ [mm], (a) no wearland, (b) wearland.

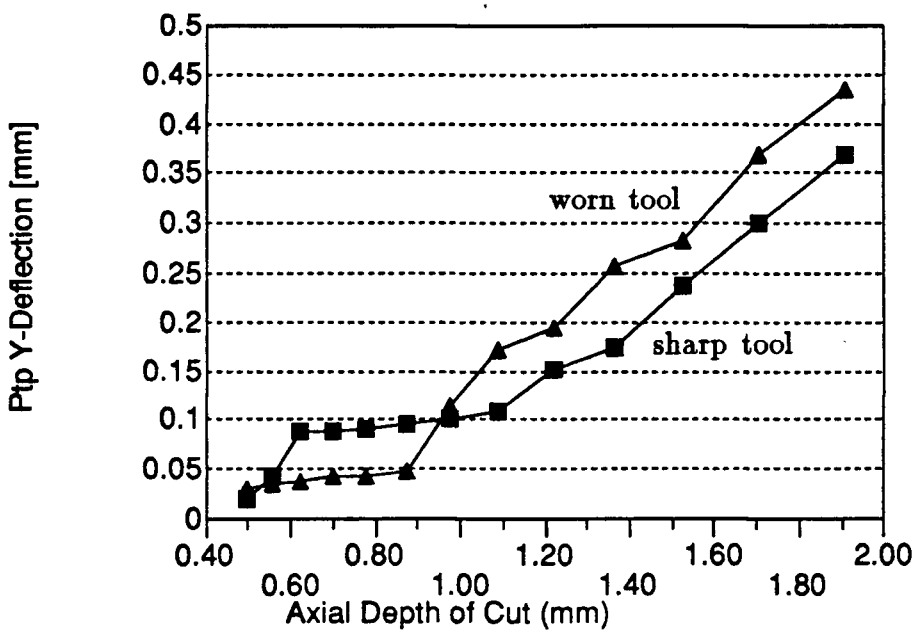
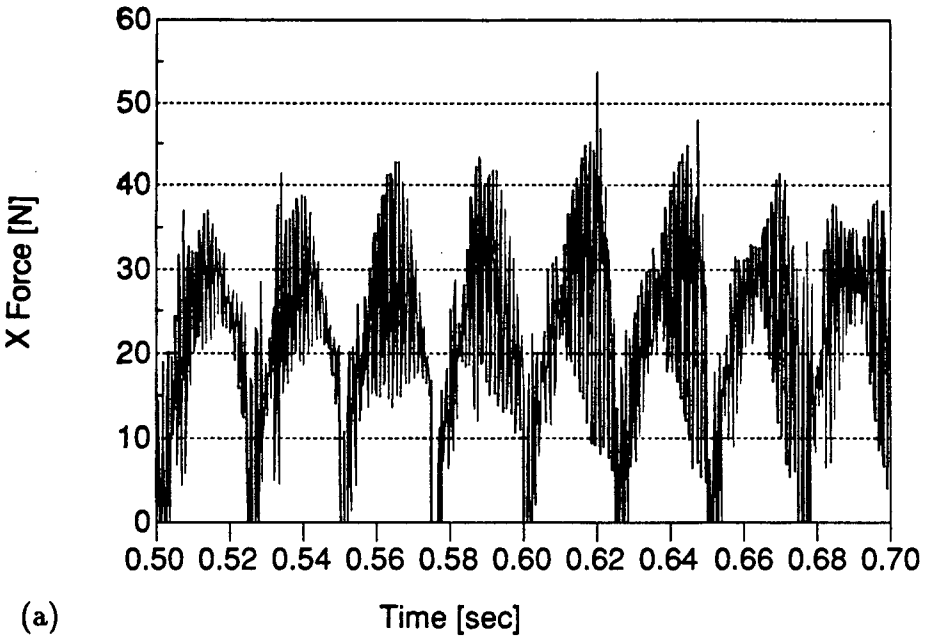
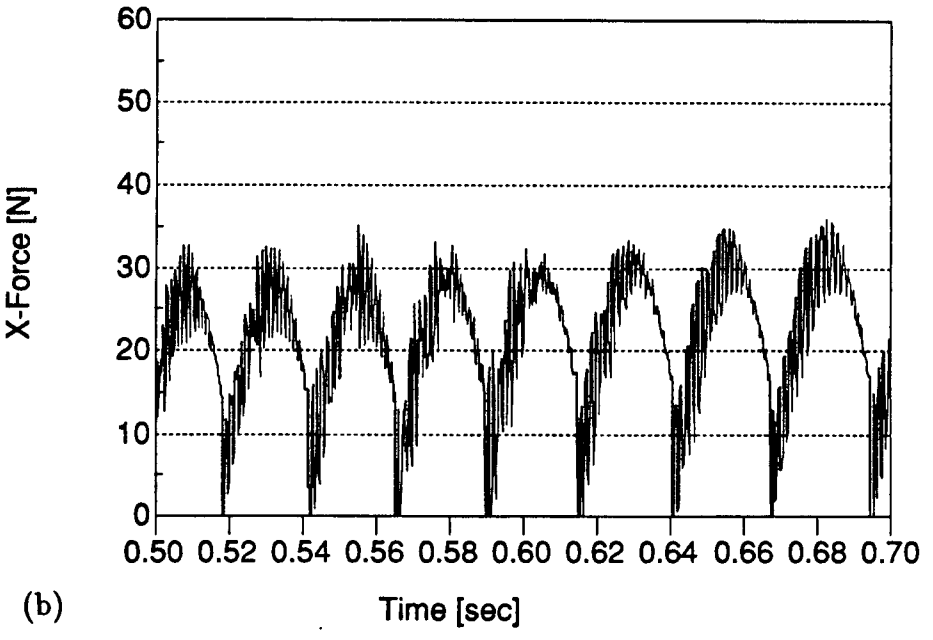


Figure 4.14: Simulated (X) cutter deflection versus depth of cut with and without flank wear. $n = 600$ [rpm], $\phi = 90^0$.



(a)



(b)

Figure 4.15: Comparison of simulated (X) cutting force with fixed and variable speed. $n = 600$ [rpm], $\phi = 90^0$, $a = 0.977$ [mm], (a) fixed, (b) variable.

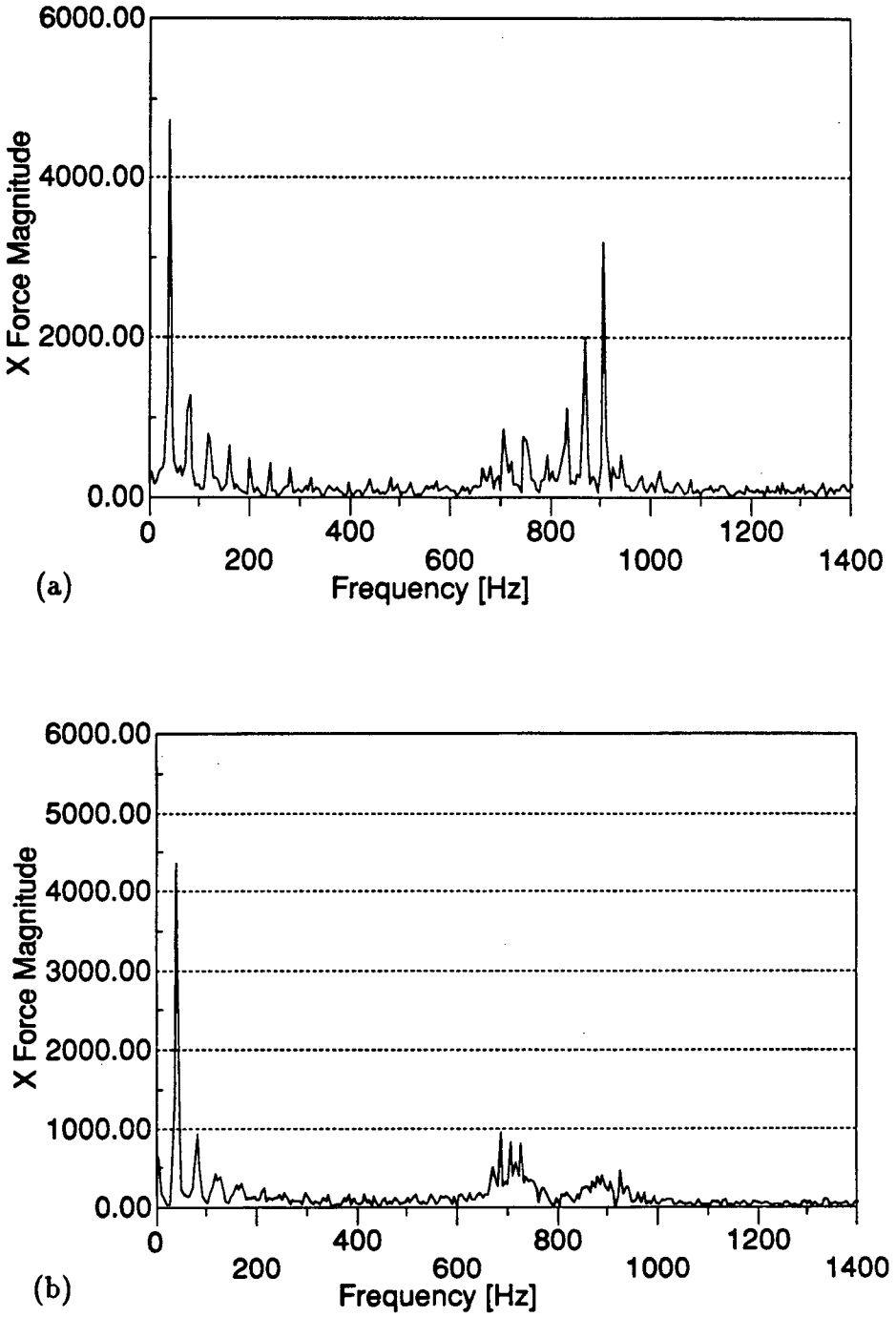
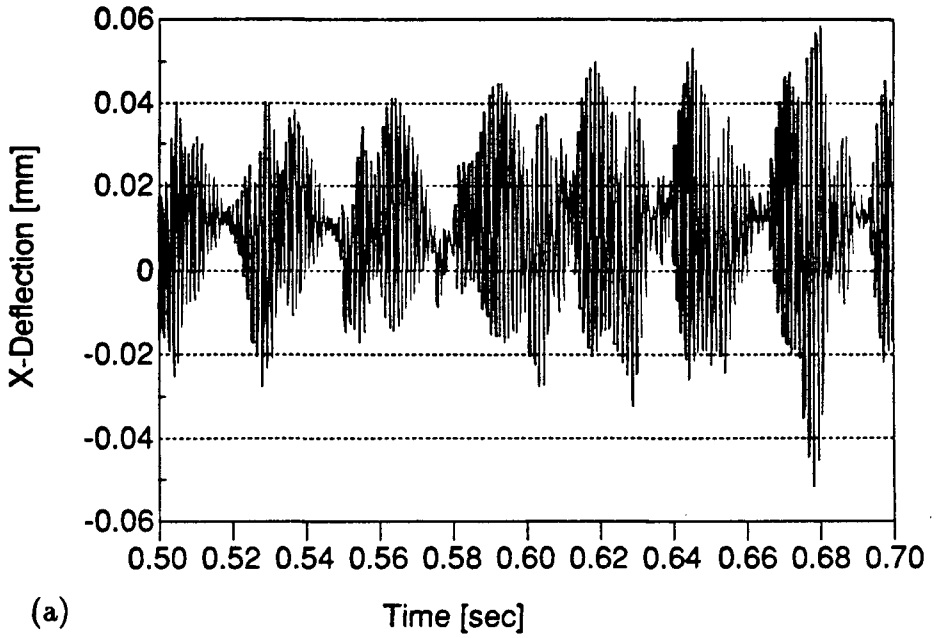


Figure 4.16: Comparison of simulated (X) cutting force spectrum with fixed and variable speed. $n = 600$ [rpm], $\phi = 90^0$, $a = 0.977$ [mm], (a) fixed, (b) variable.



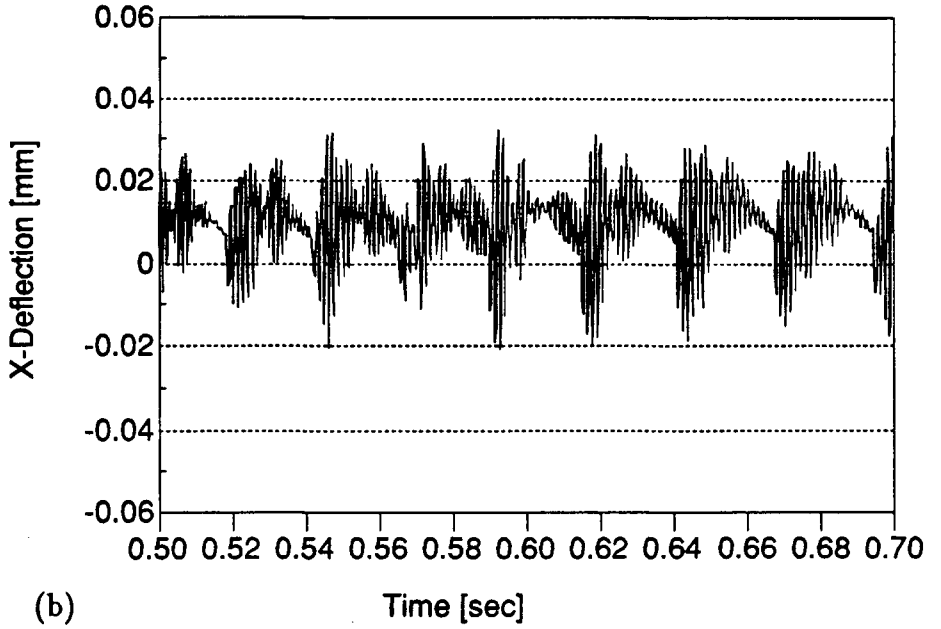


Figure 4.17: Comparison of simulated (X) cutter deflection with fixed and variable speed. $n = 600$ [rpm], $\phi = 90^0$, $a = 0.977$ [mm], (a) fixed, (b) variable.

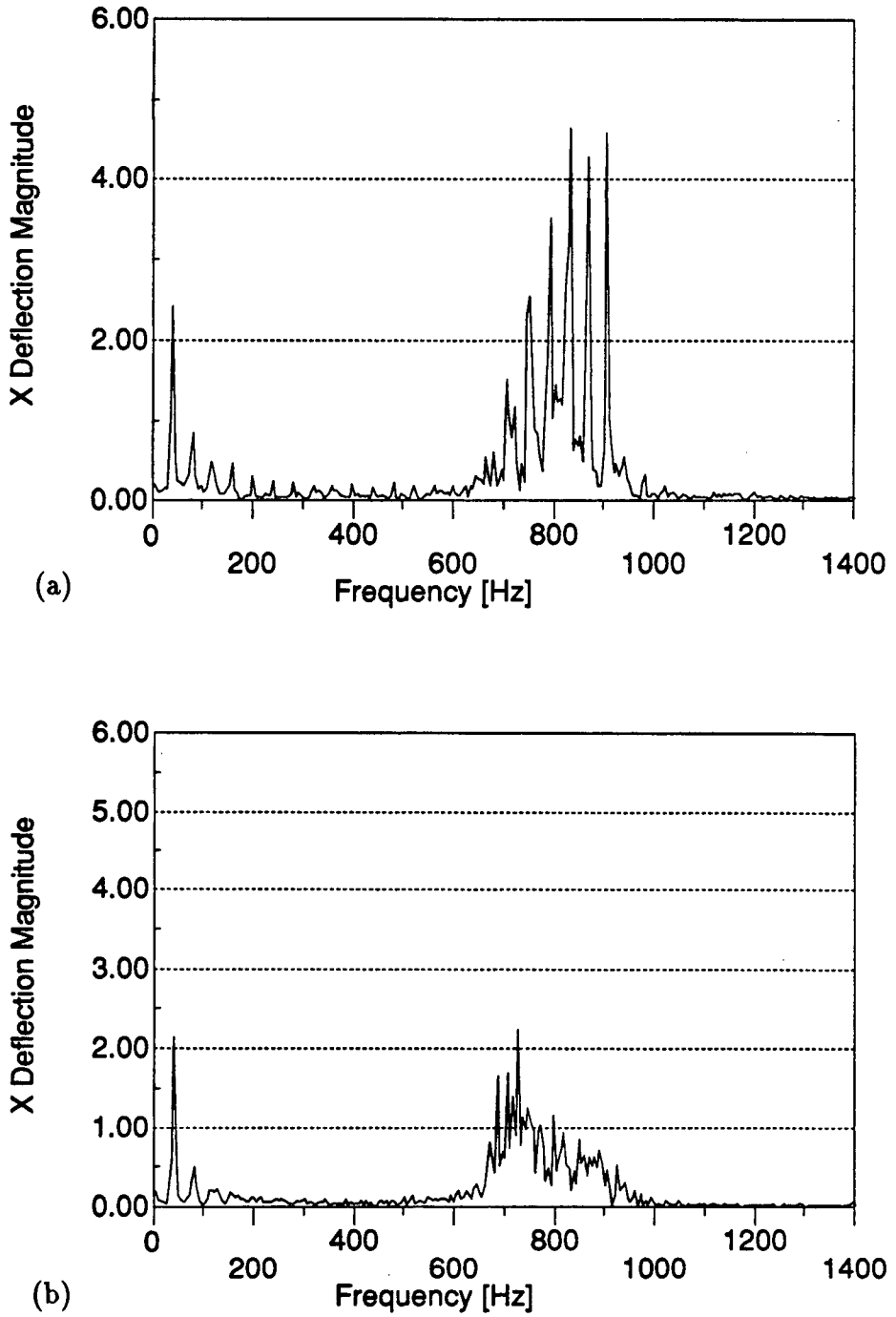


Figure 4.18: Comparison of simulated (X) cutter deflection spectrum with fixed and variable speed. $n = 600$ [rpm], $\phi = 90^0$, $a = 0.977$ [mm], (a) fixed, (b) variable.

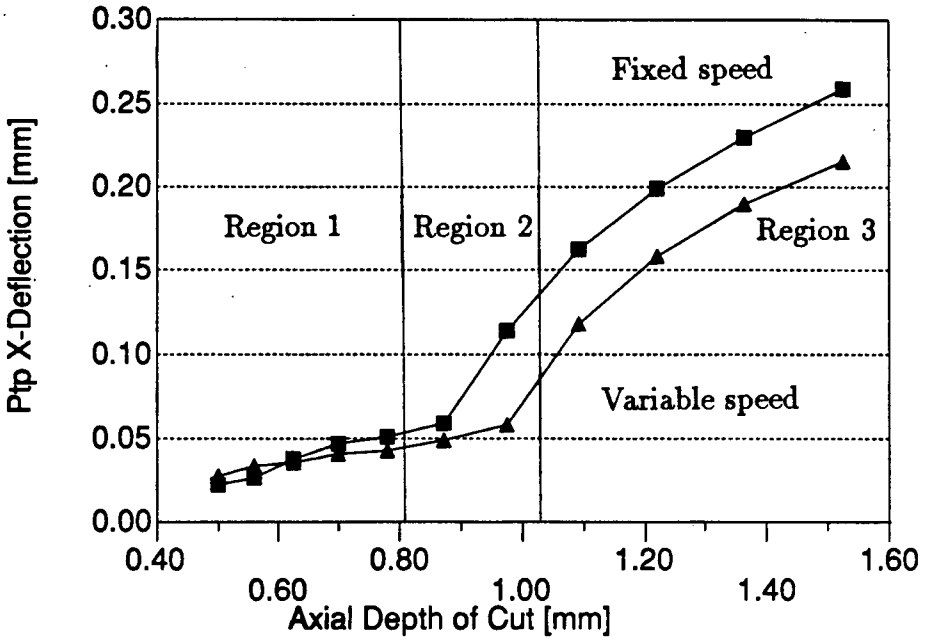


Figure 4.19: Simulated (X) cutter deflection versus depth of cut for fixed and variable speed. $n = 600$ [rpm], $\phi = 90^0$.

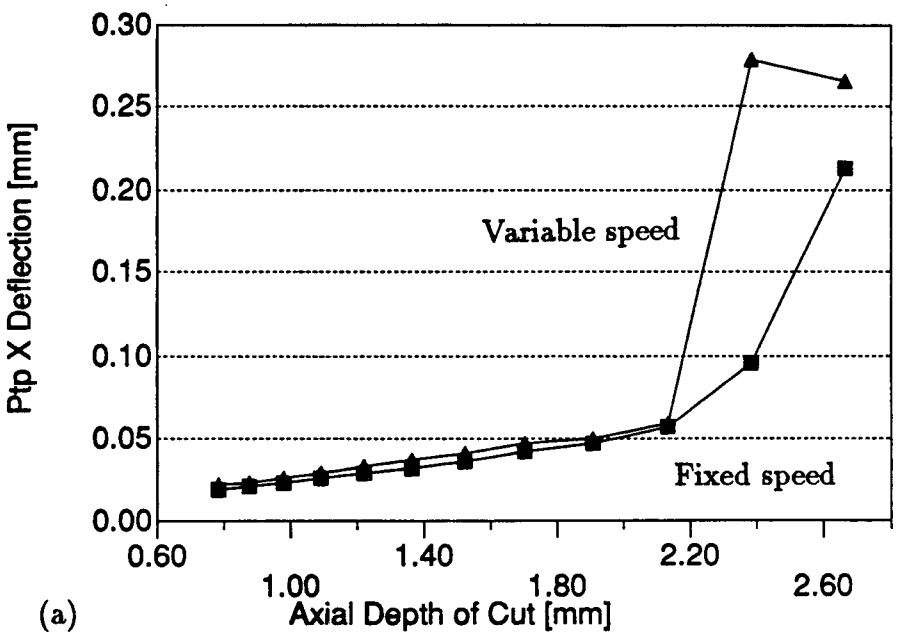
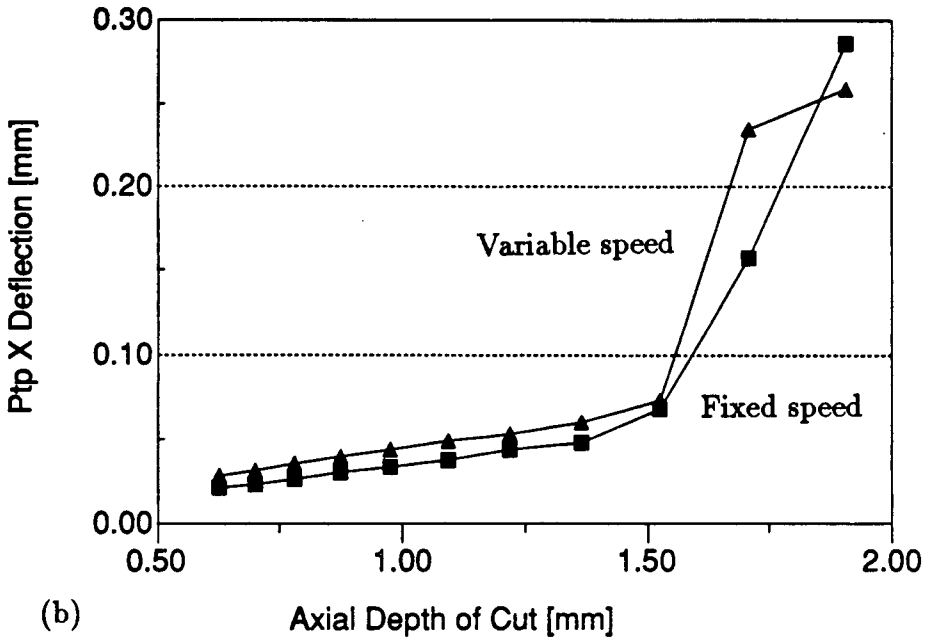


Figure 4.20: Simulated (X) cutter deflection versus depth of cut for fixed and variable speed. $n = 600$ [rpm], $\phi = 50^0$.



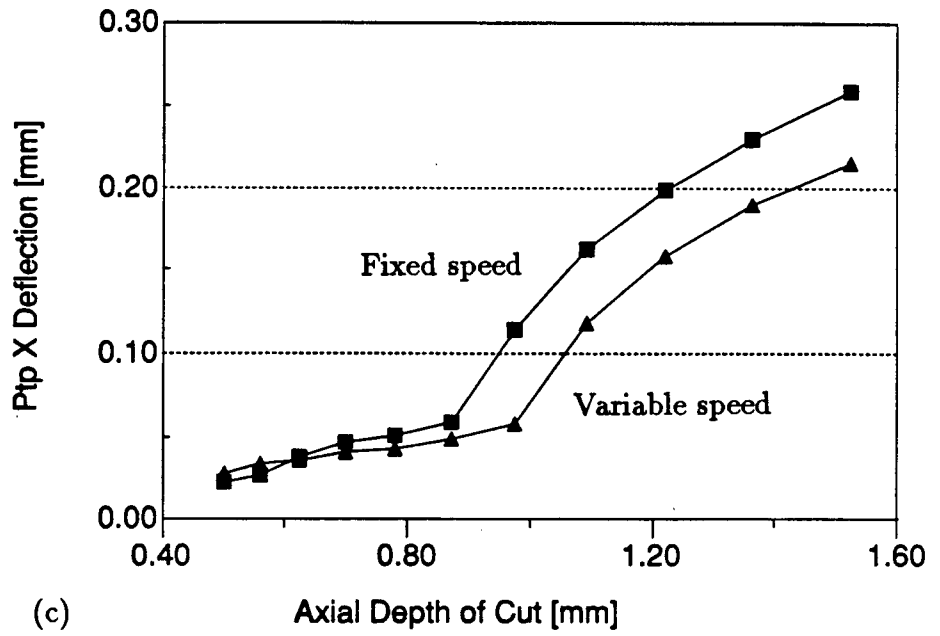
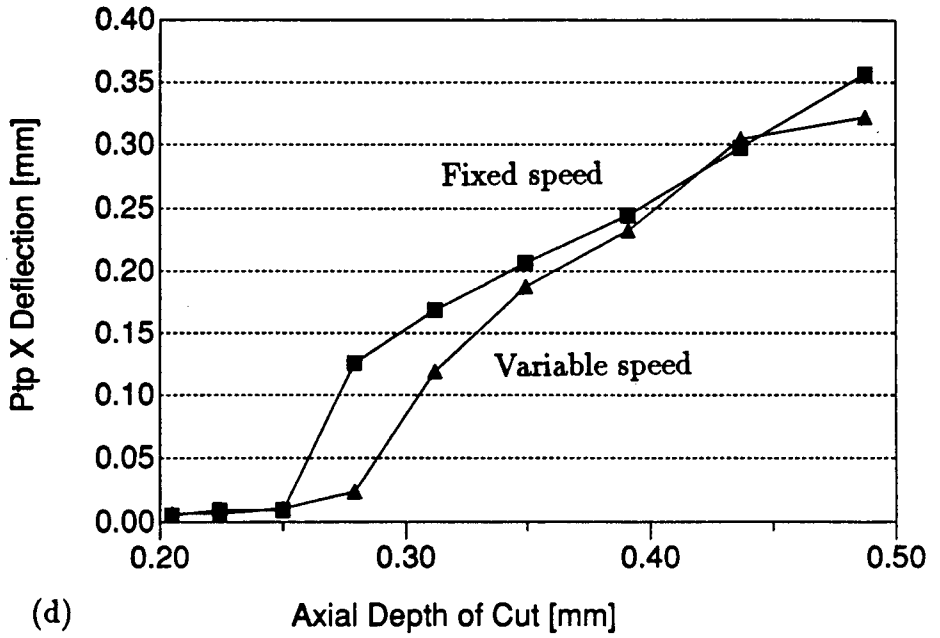
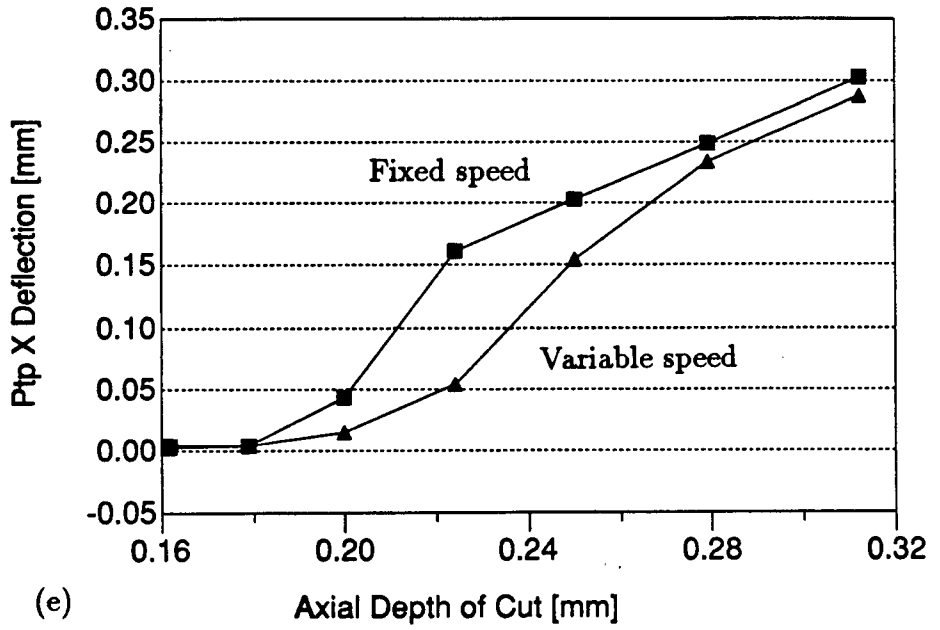


Figure 4.20: Simulated (X) cutter deflection versus depth of cut for fixed and variable speed. n = 600 [rpm], ϕ = (b) 65^0, (c) 90^0.



(d)



(e)

Figure 4.20: Simulated (X) cutter deflection versus depth of cut for fixed and variable speed. $n = 600$ [rpm], $\phi =$ (d) 135^0, (e) 180^0.

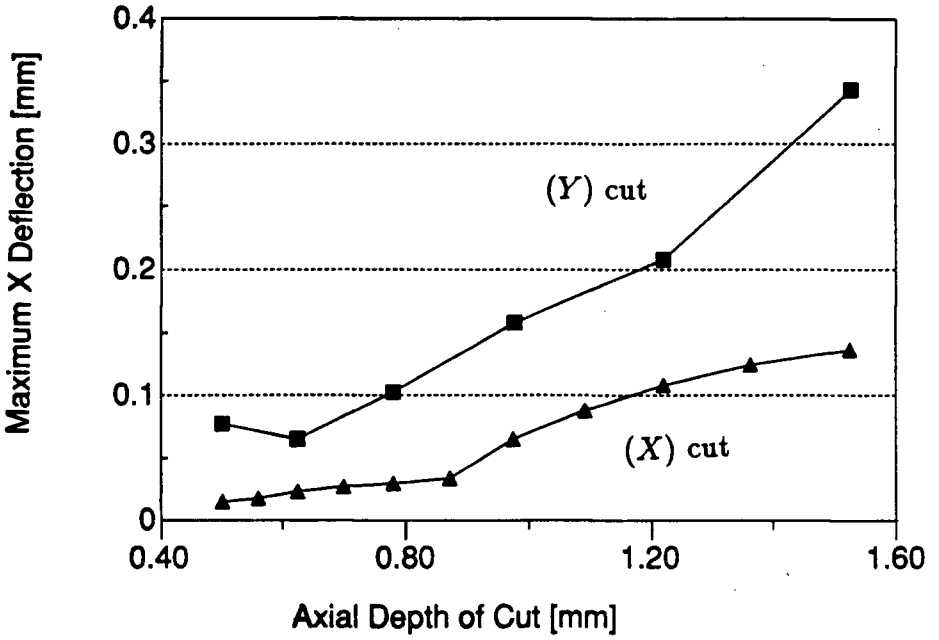


Figure 4.21: Comparison of simulated (X) cutter deflection between (X) and (Y) direction cuts. $n = 600$ [rpm], $\phi = 90^0$.

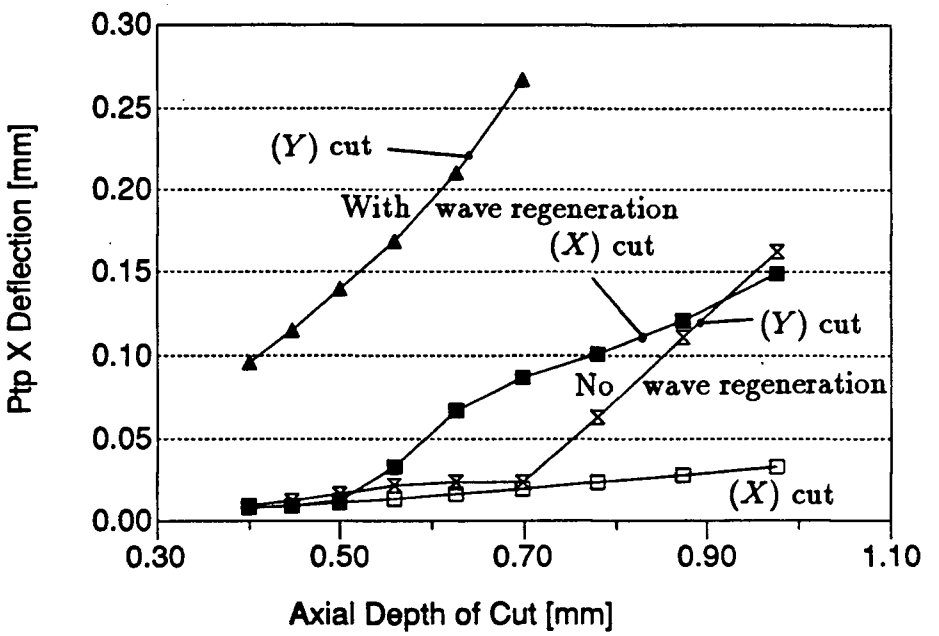


Figure 4.22: Comparison of simulated (X) cutter deflection between (X) and (Y) direction cuts with and without wave regeneration. $n = 600$ [rpm], $\phi = 90^0$.

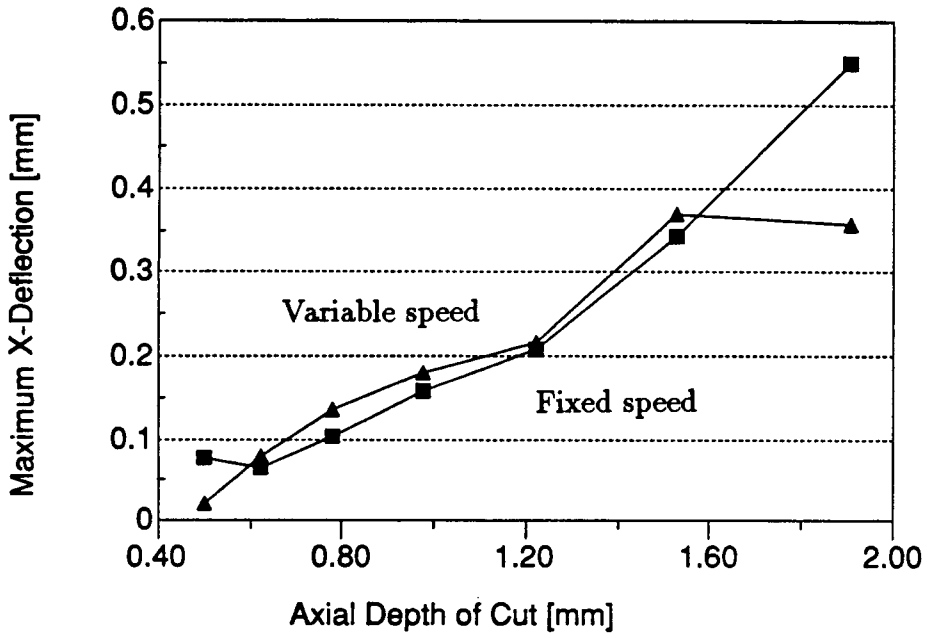


Figure 4.23: Comparison of simulated (X) cutter deflection between fixed and variable speed for (Y) direction cut. $n = 600$ [rpm], $\phi = 90^0$.

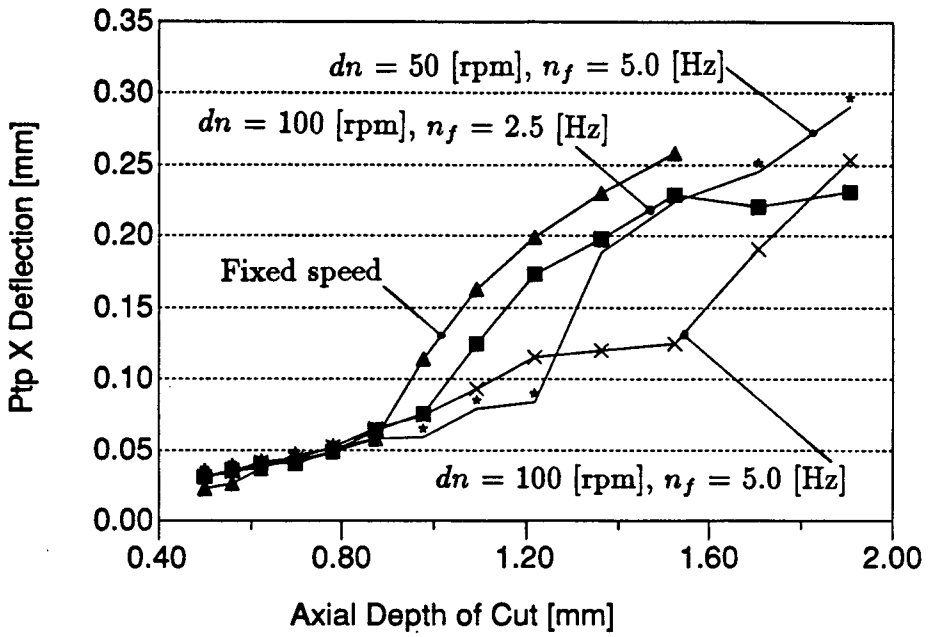


Figure 4.24: Simulated (X) cutter deflection versus different speed variations. $n_o = 600$ [rpm], $\phi = 90^0$.

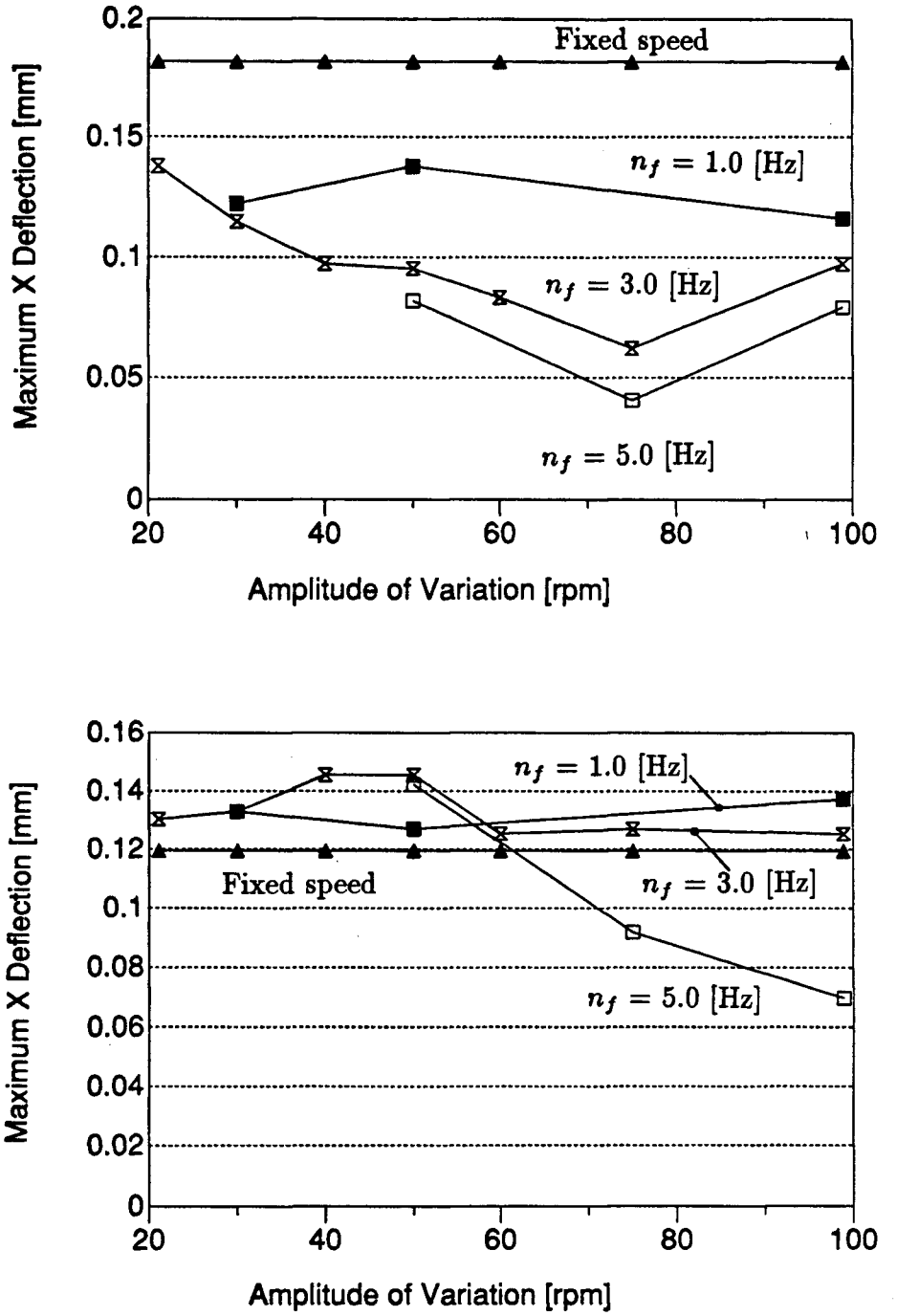
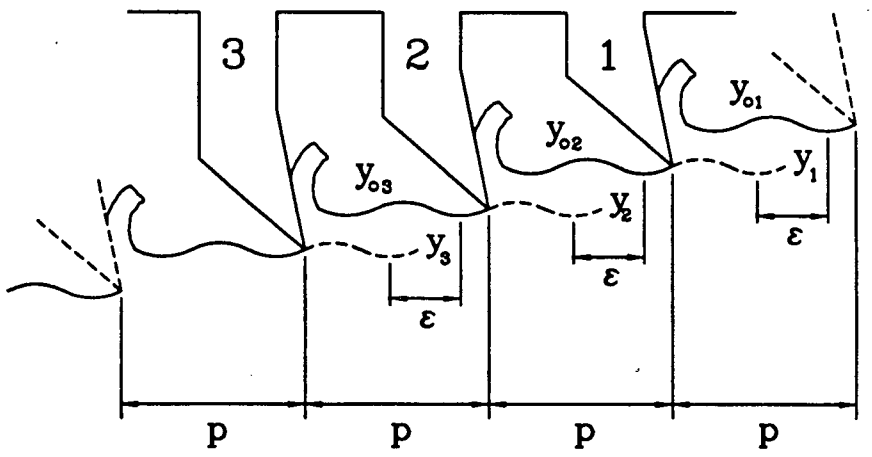
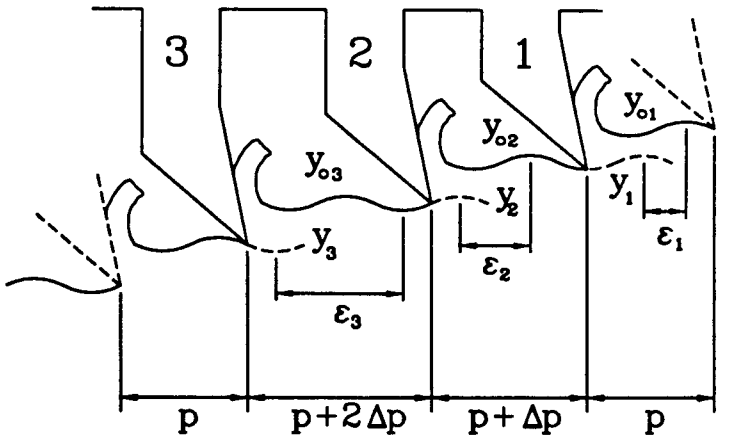


Figure 4.25: Simulated (X) cutter deflection for different speed variations at two speeds. n_o = (a) 800, (b) 1100 [rpm], $\phi = 90^0$, a = (a) 0.781, (b) 0.625 [mm].



(a)



(b)

Figure 4.26: Cutting geometry for cutter with uniform and nonuniform tooth pitch.

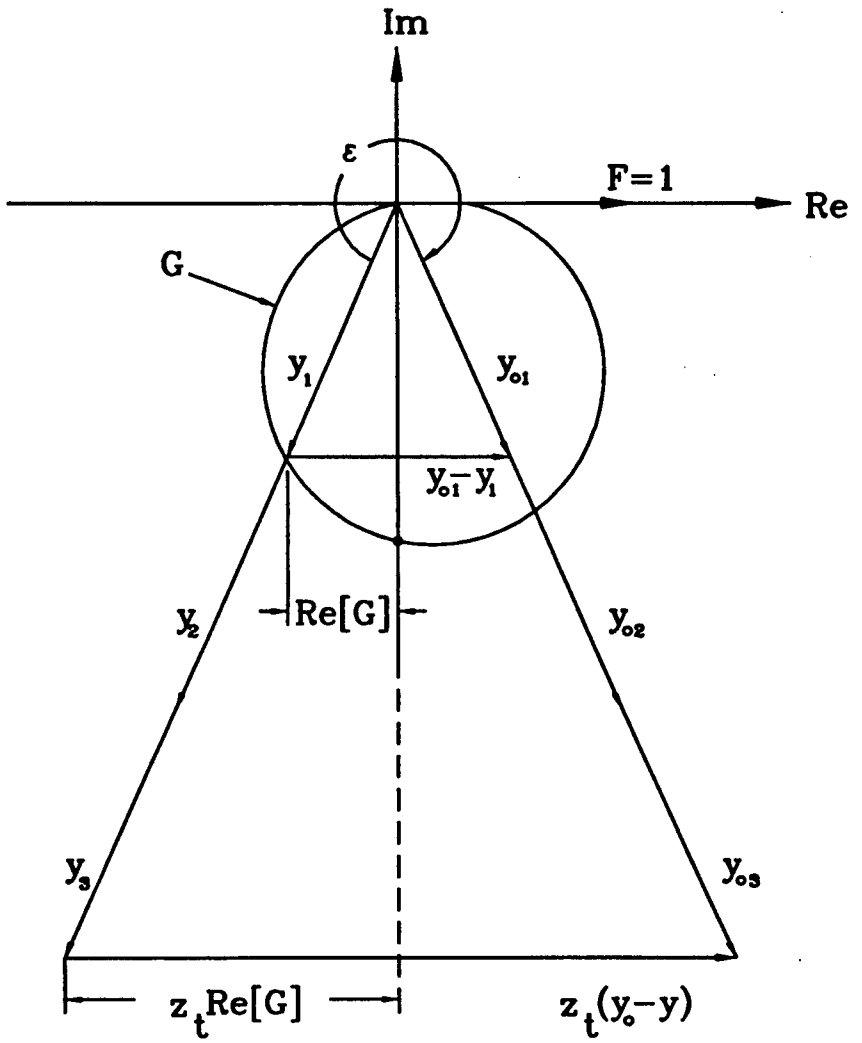


Figure 4.27: Phase plane plot at the limit of stability for a cutter with uniform tooth pitch.

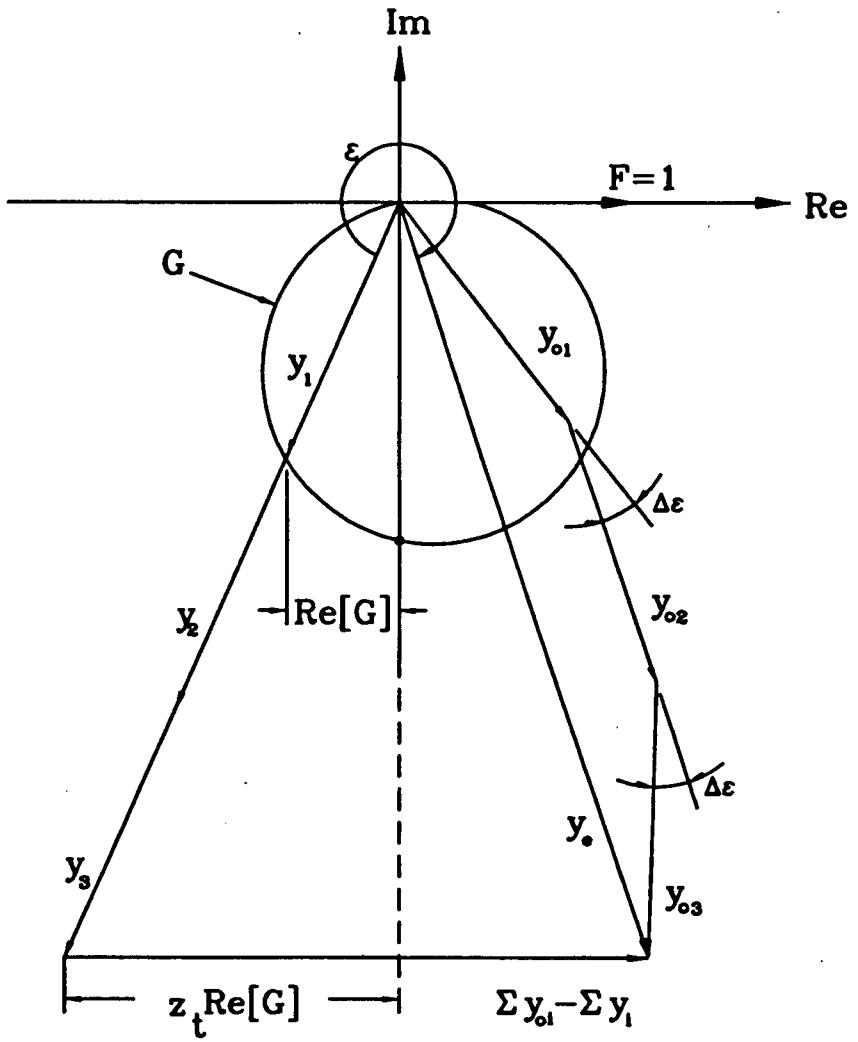


Figure 4.28: Phase plane plot at the limit of stability for a cutter with nonuniform tooth pitch.

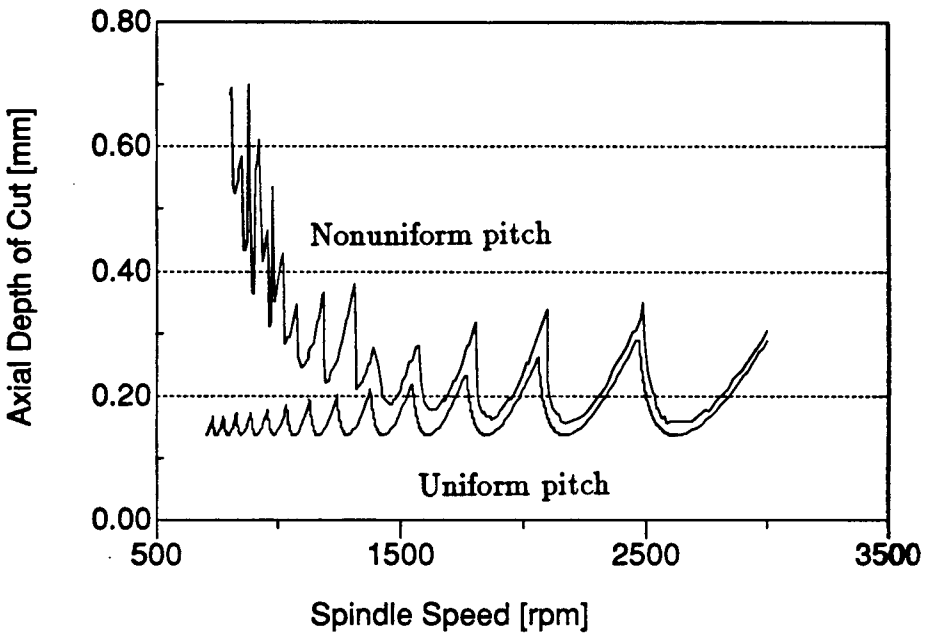


Figure 4.29: Stability lobe curve for cutter with nonuniform tooth pitch.

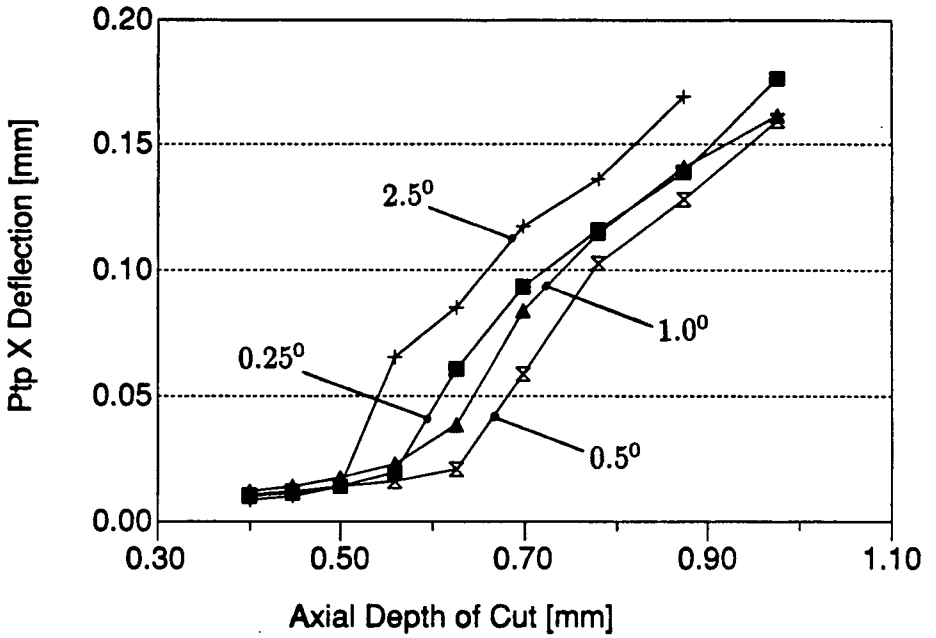


Figure 4.30: Simulated (X) deflection for cutters with linear tooth pitch variations. n = 600 [rpm], $\phi = 90^0$.

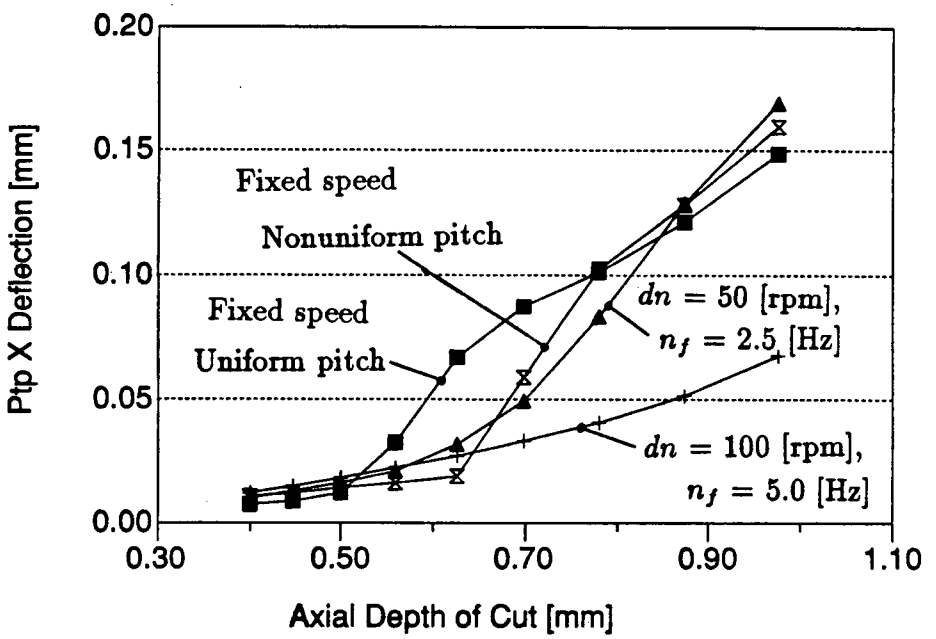


Figure 4.31: Comparison of simulated (X) deflection for nonuniform tooth pitch cutters with variable speed cutting. n_o = 600 [rpm], $\phi = 90^0$.

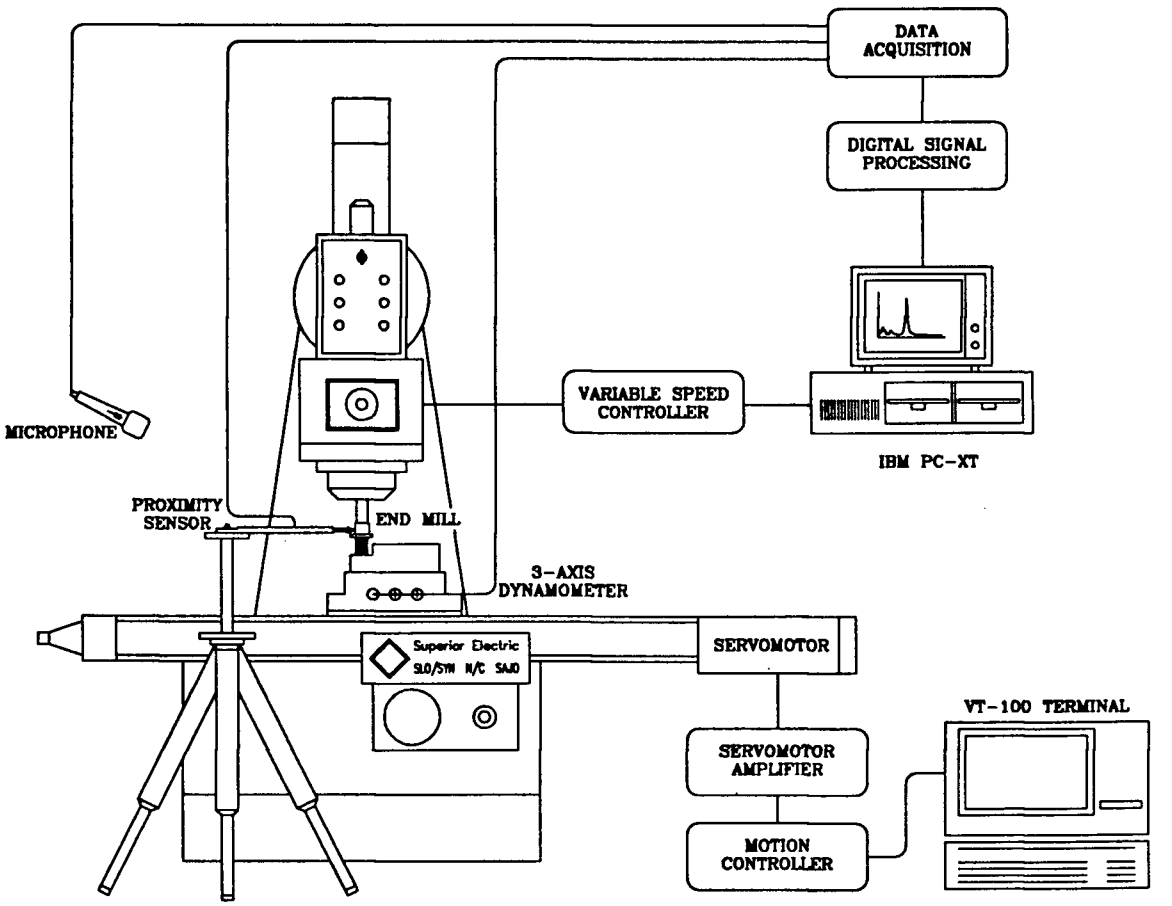
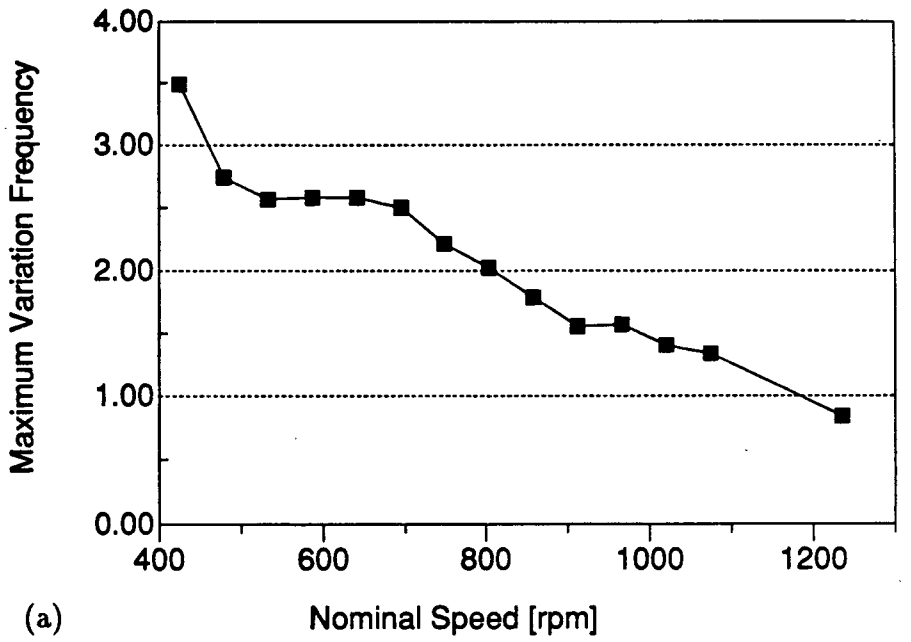


Figure 4.32: Experimental set-up.



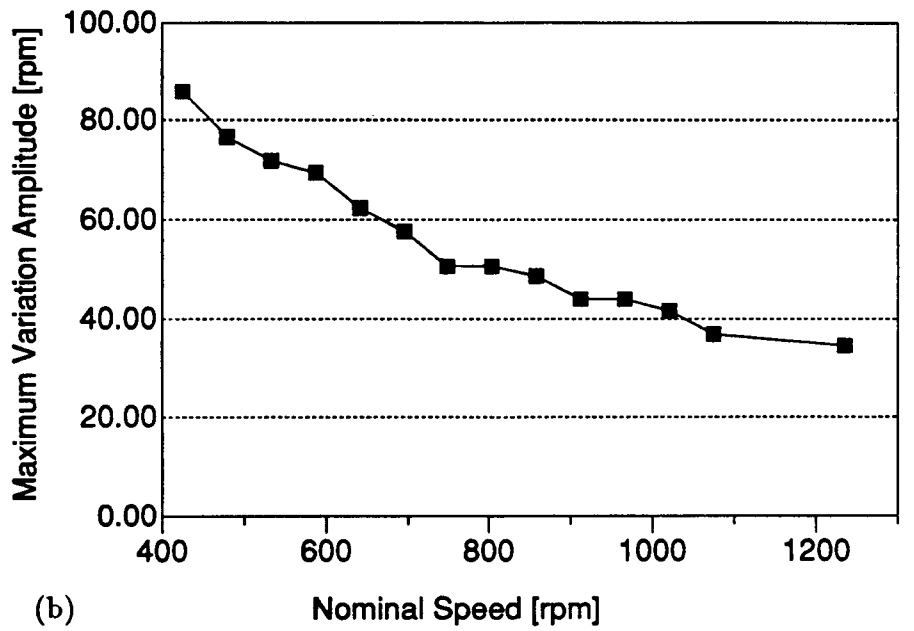
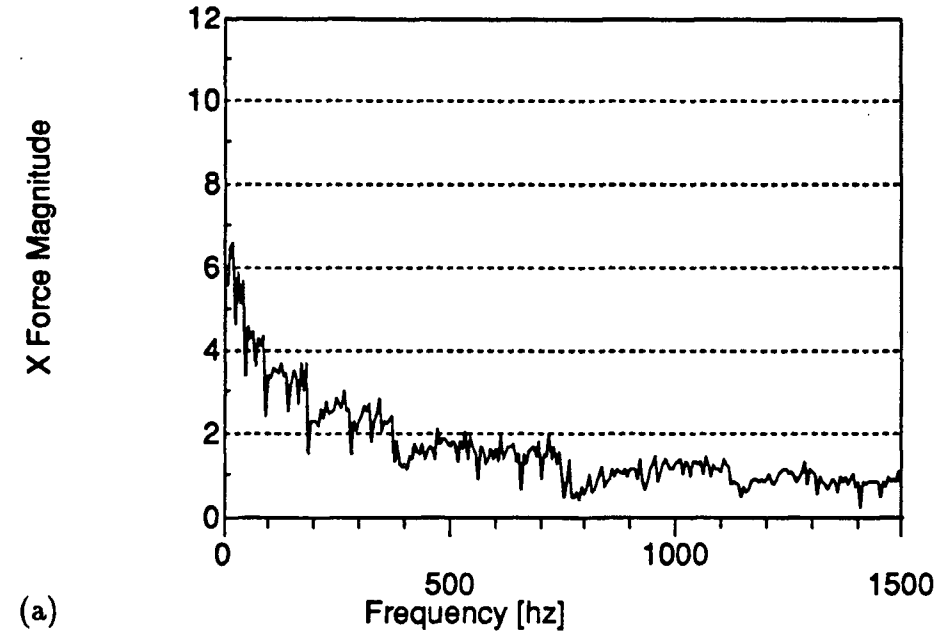


Figure 4.33: Variable speed spindle drive variation limits. (a) $dn = 0.1n_o$, (b) $n_f = 0.2n_o$.



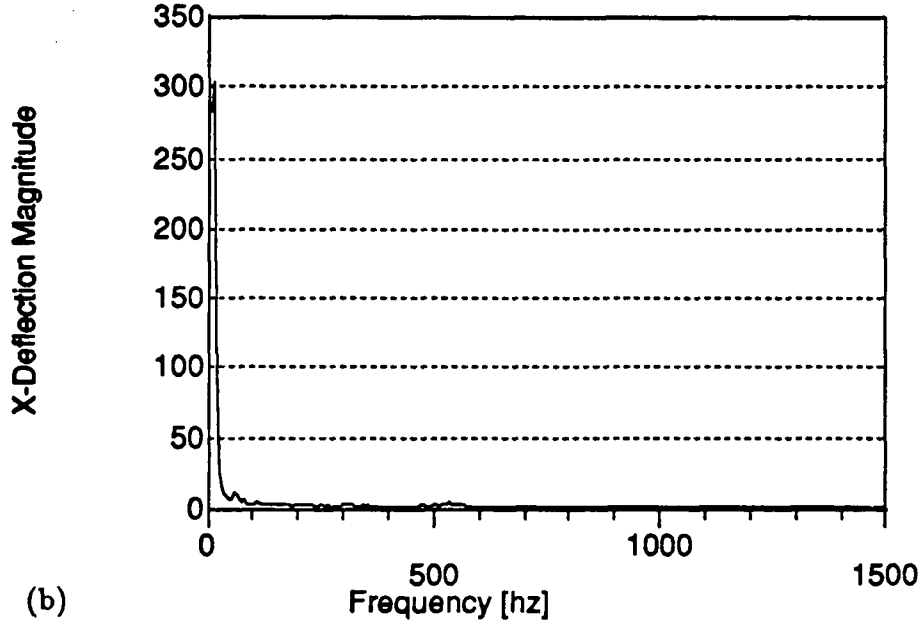


Figure 4.34: Experimental (X) cutting force and (X) cutter deflection spectrums – air cutting. $n = 600$ [rpm].

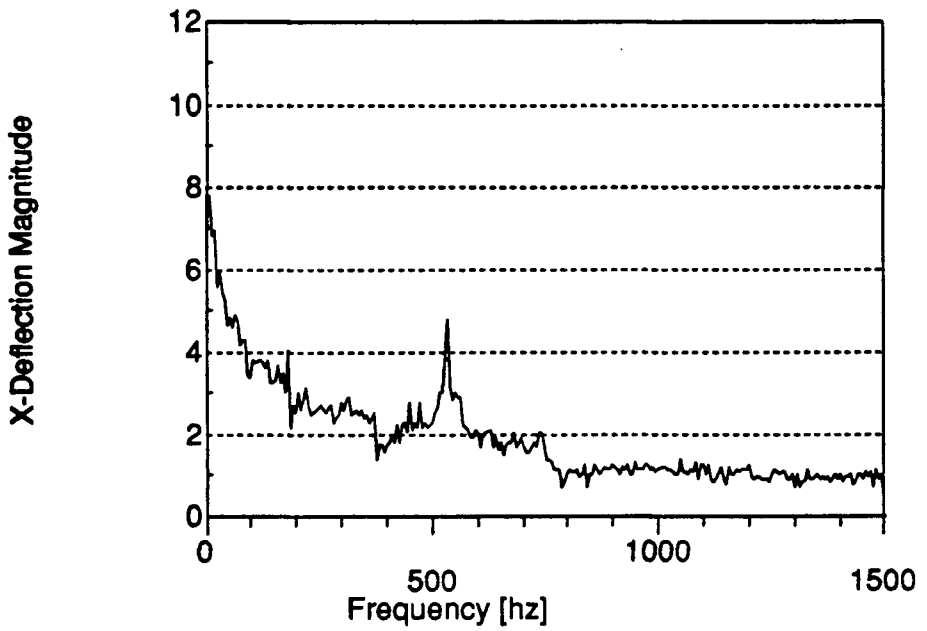


Figure 4.35: Experimental filtered (X) cutter deflection spectrum – air cutting. $n = 600$ [rpm].

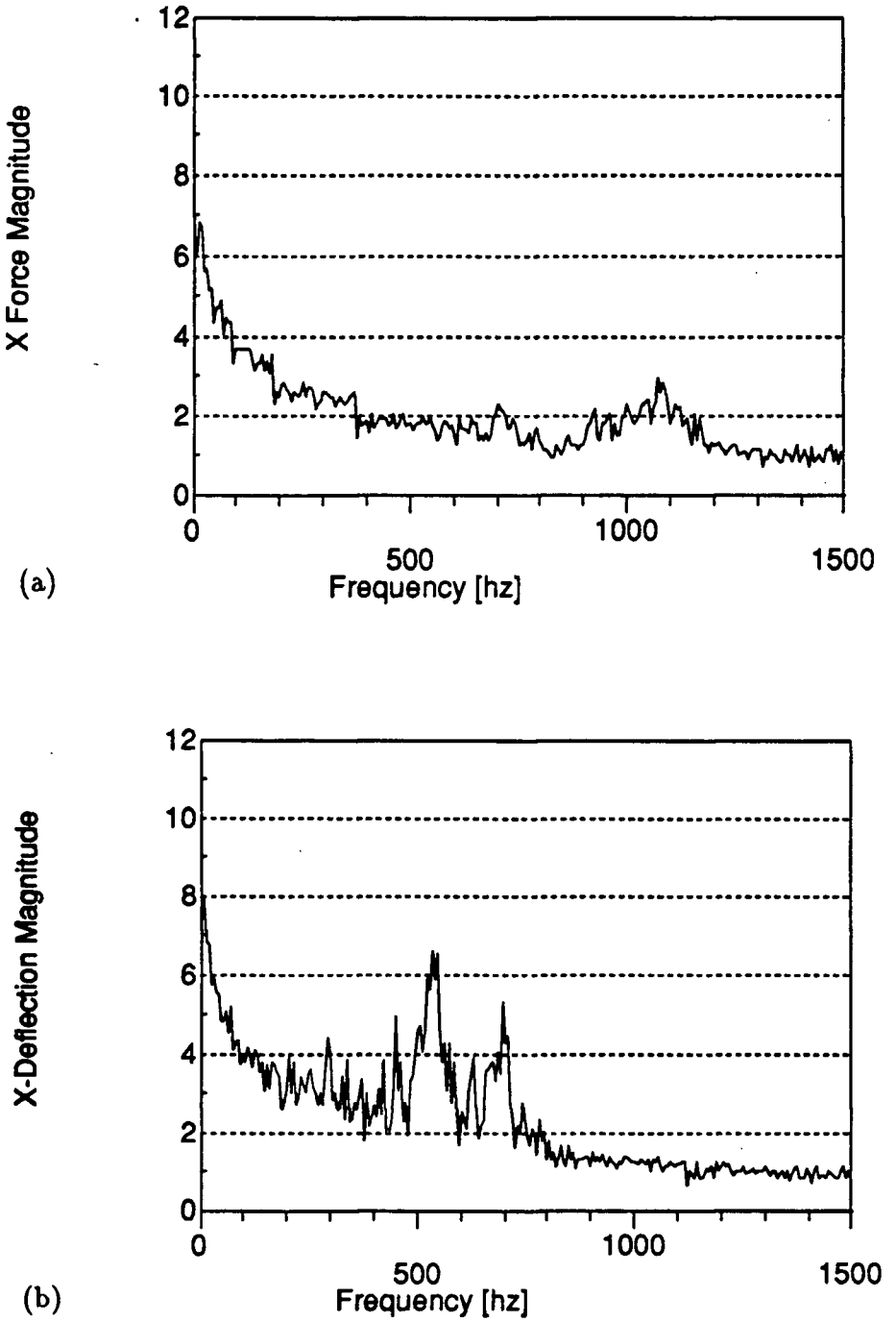


Figure 4.36: Experimental (X) cutting force and filtered (X) cutter deflection spectrums. $n = 600$ [rpm], $\phi = 90^0$, $a = 1.27$ [mm].

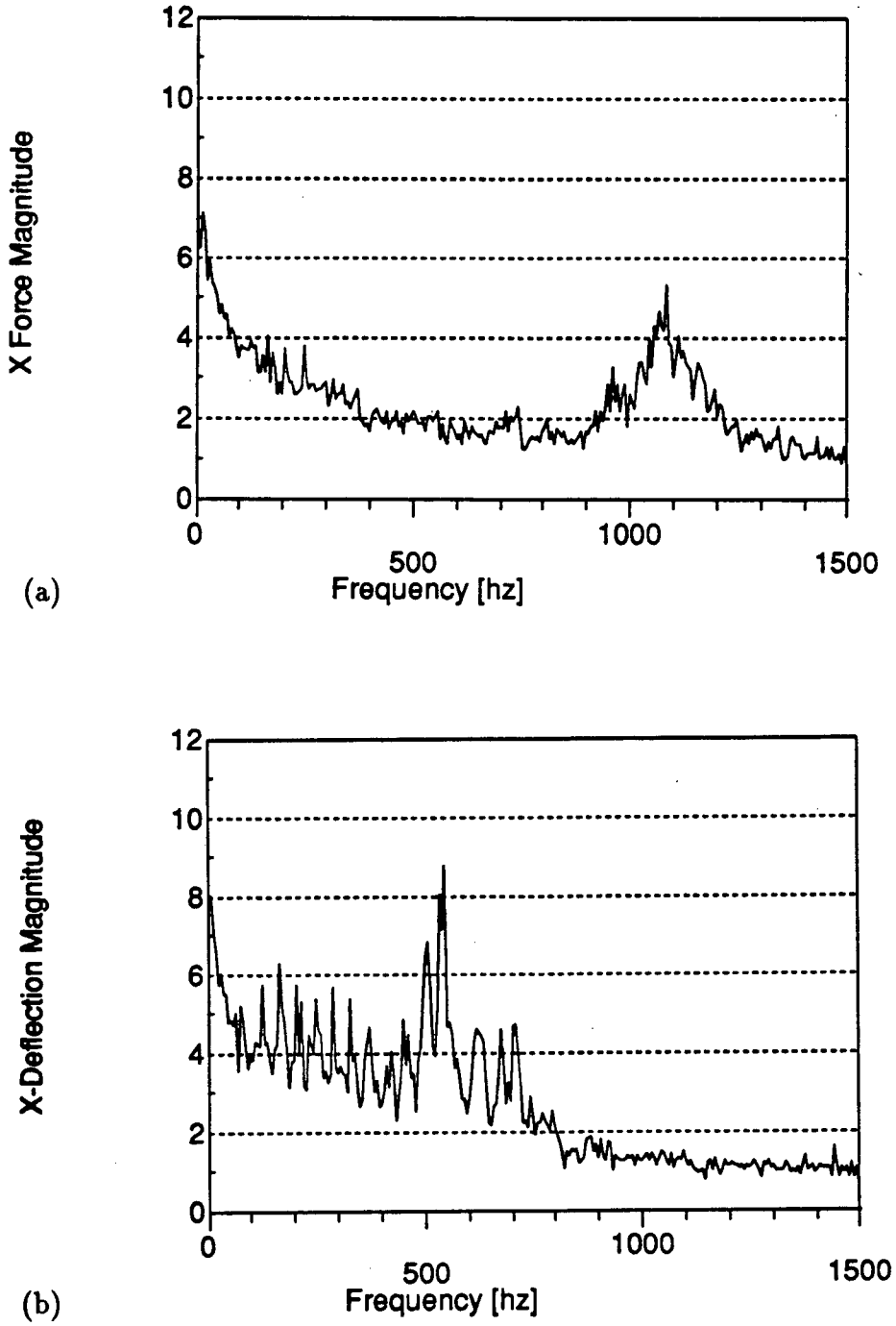
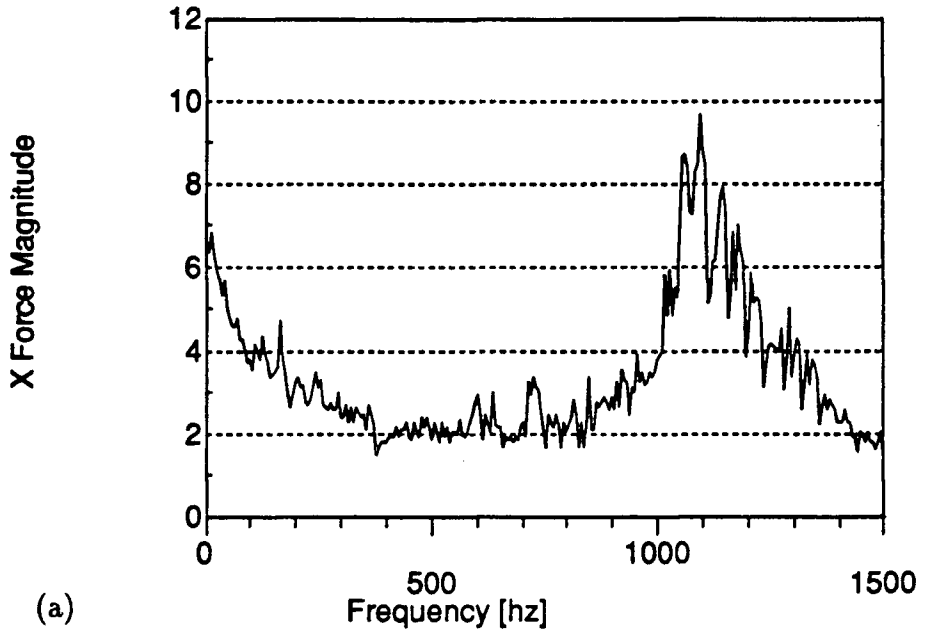
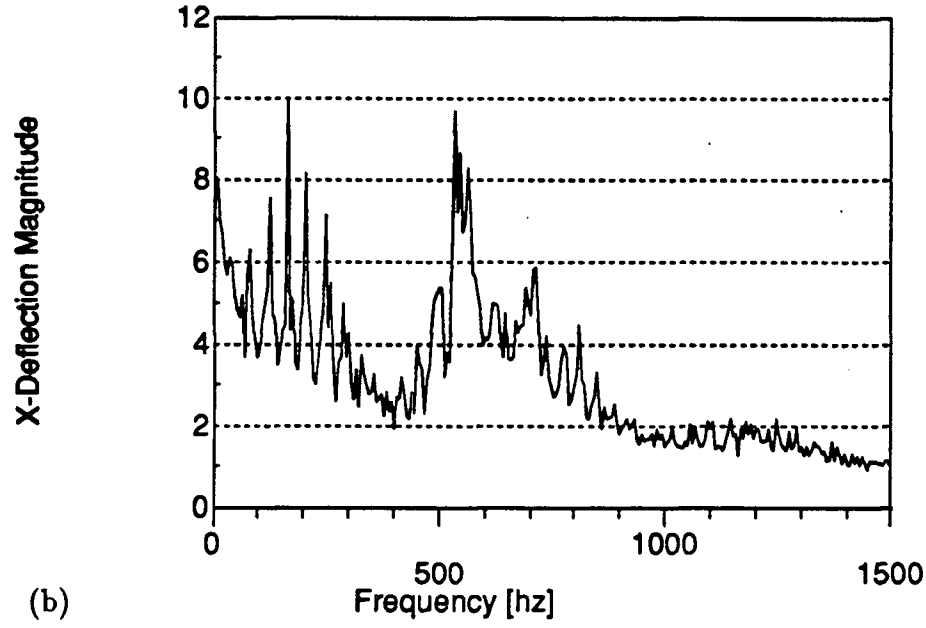


Figure 4.37: Experimental (X) cutting force and filtered (X) cutter deflection spectrums. $n = 600$ [rpm], $\phi = 90^0$, $a = 2.54$ [mm].

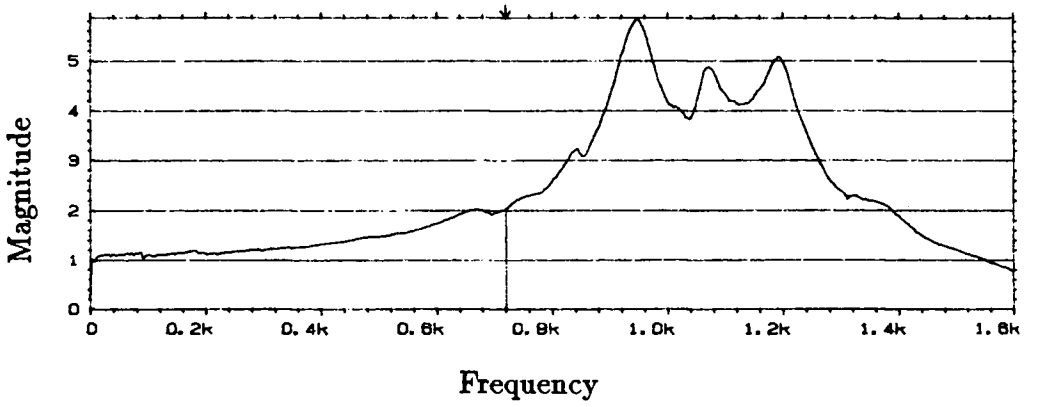


(a)

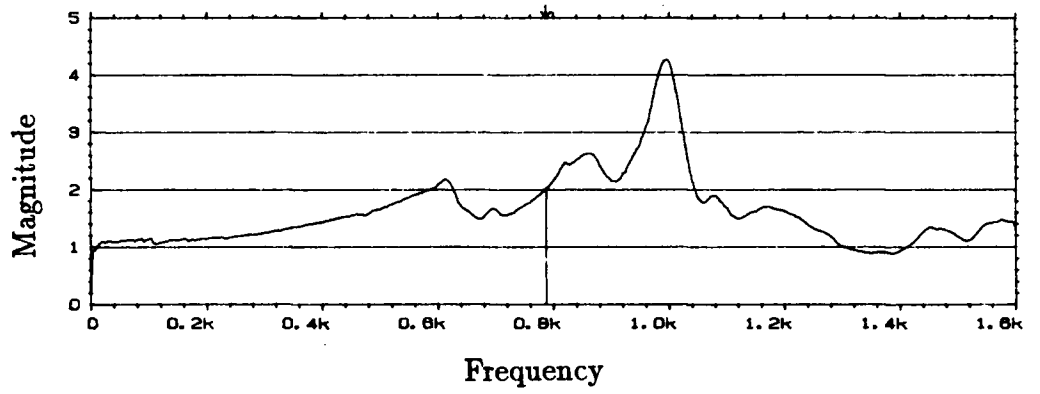


(b)

Figure 4.38: Experimental (X) cutting force and filtered (X) cutter deflection spectrums. $n = 600$ [rpm], $\phi = 90^0$, $a = 5.08$ [mm].



(a)



(b)

Figure 4.39: Measured transfer function of table dynamometer. (a) (X) direction, (b) (Y) direction.

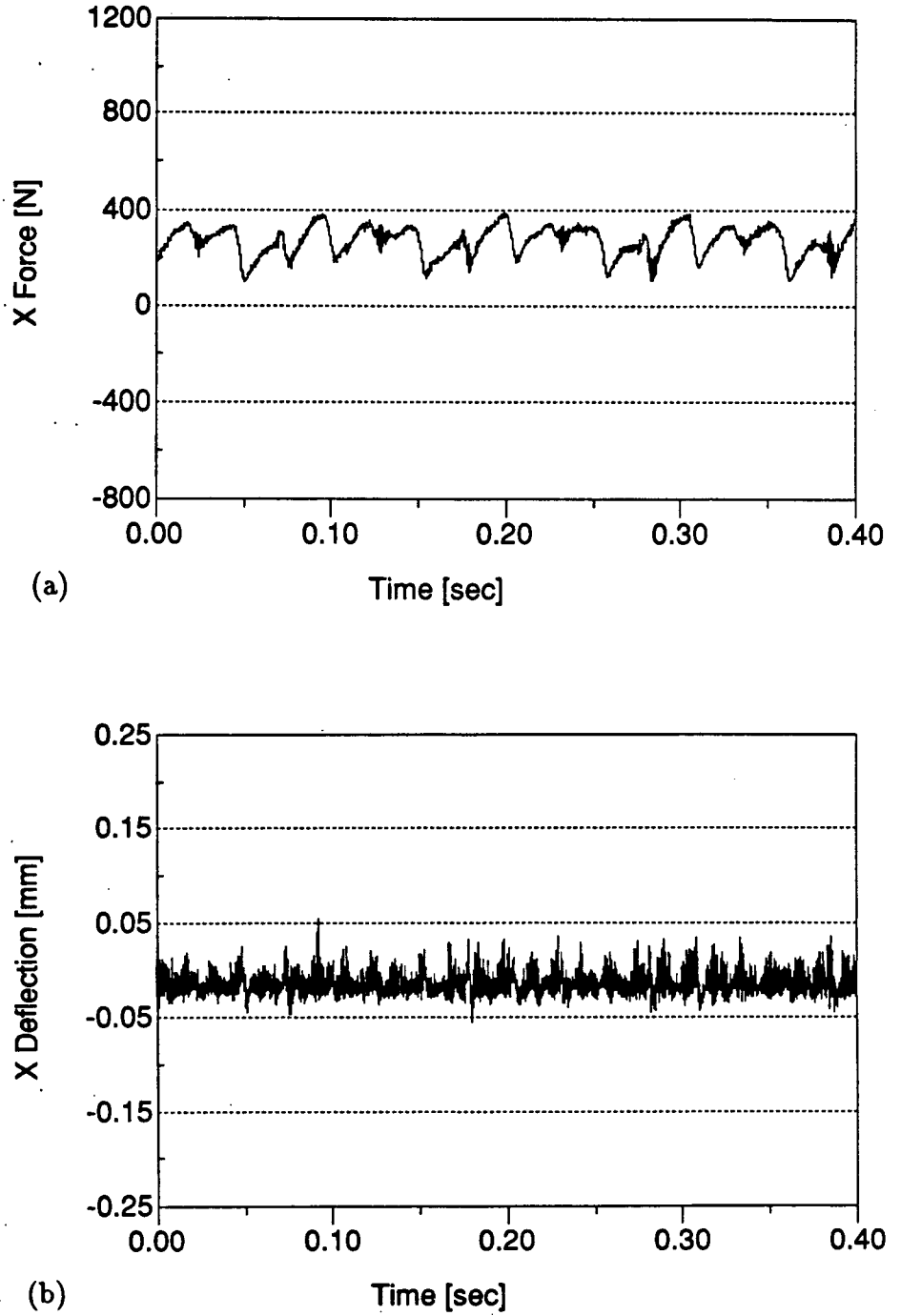


Figure 4.40: Experimental (X) cutting force and filtered (X) cutter deflection. $n = 600$ [rpm], $\phi = 180^0$, $a = 1.27$ [mm].

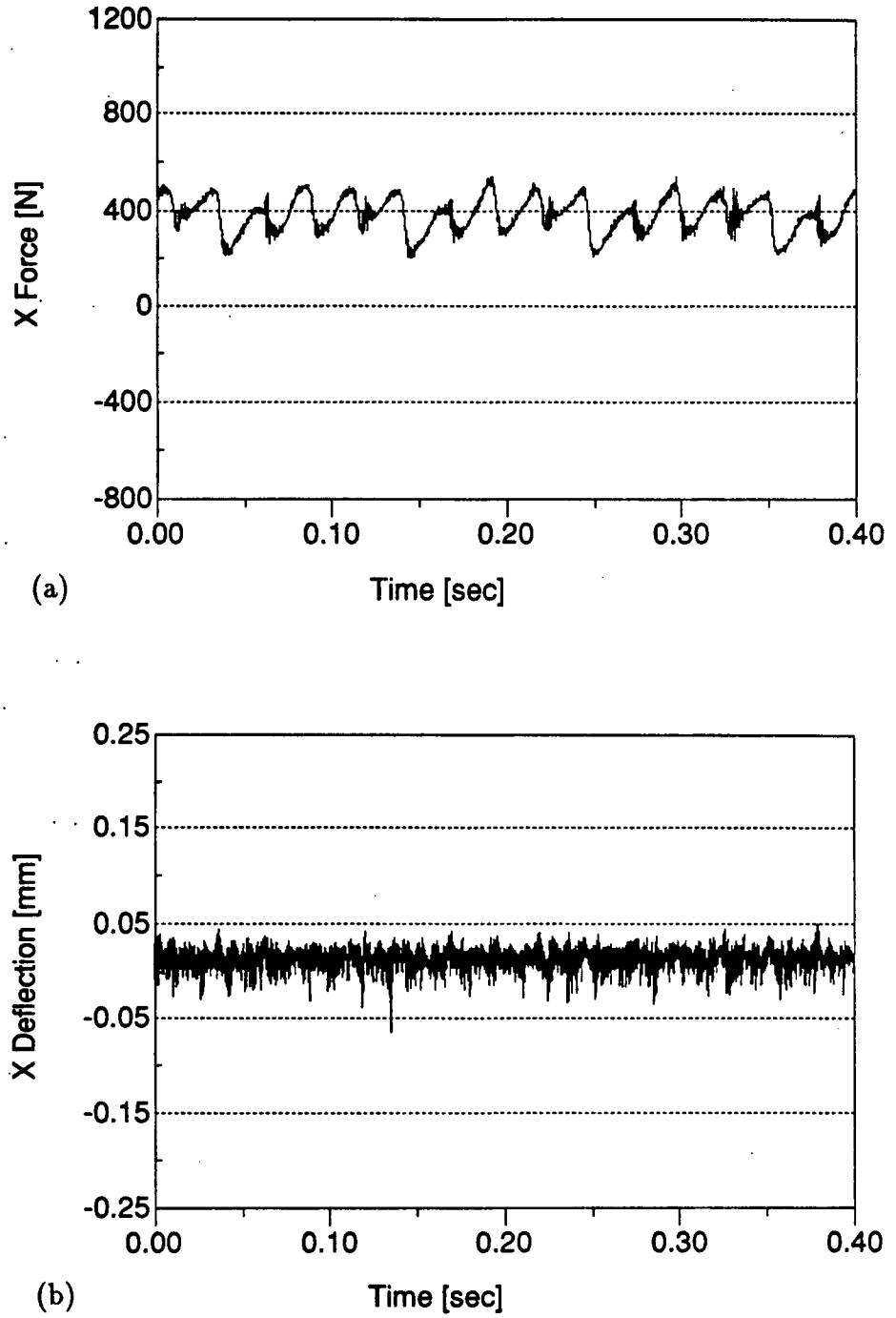


Figure 4.41: Experimental (X) cutting force and filtered (X) cutter deflection. $n = 600$ [rpm], $\phi = 180^0$, $a = 2.54$ [mm].

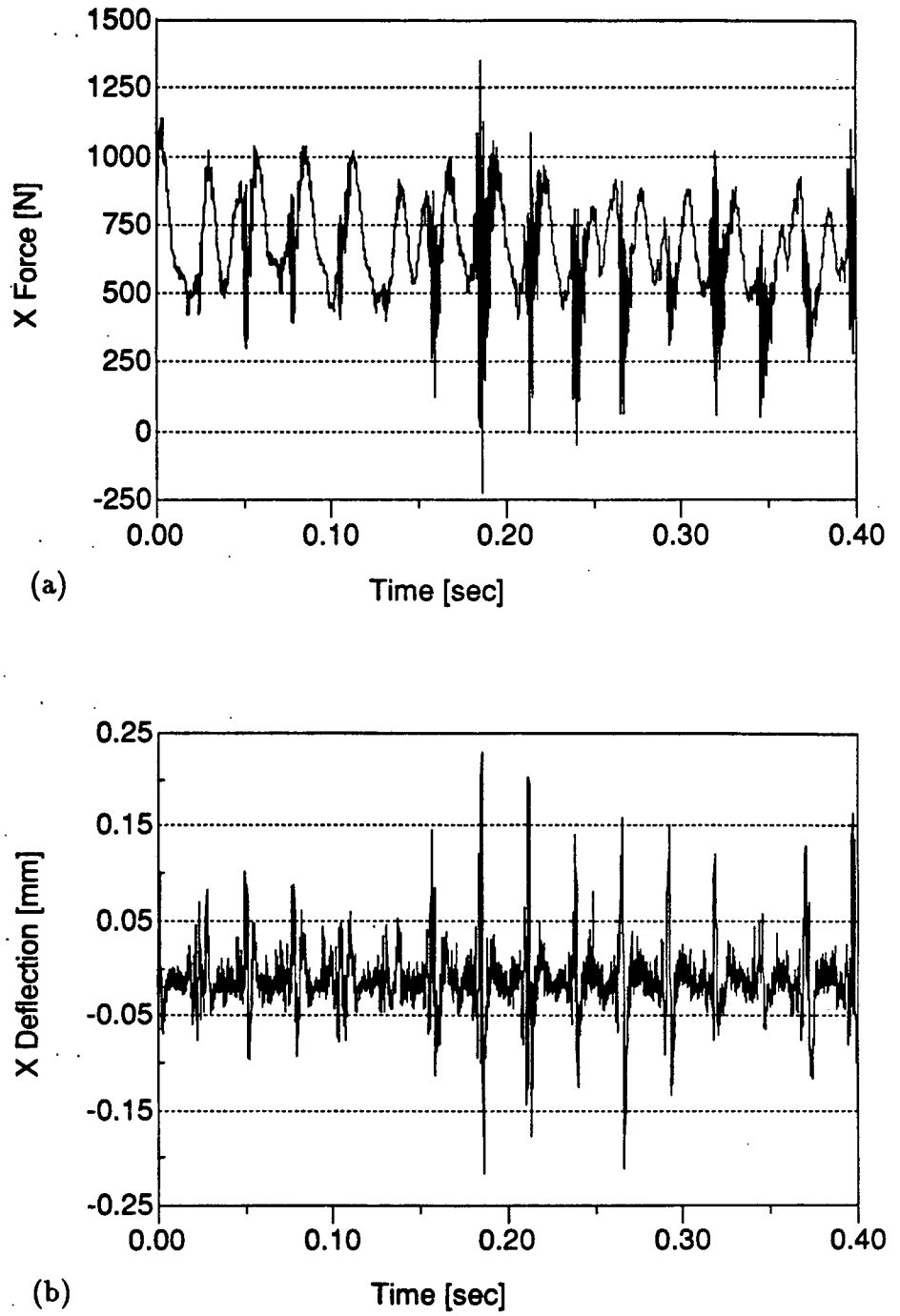


Figure 4.42: Experimental (X) cutting force and filtered (X) cutter deflection. $n = 600$ [rpm], $\phi = 180^0$, $a = 3.81$ [mm].

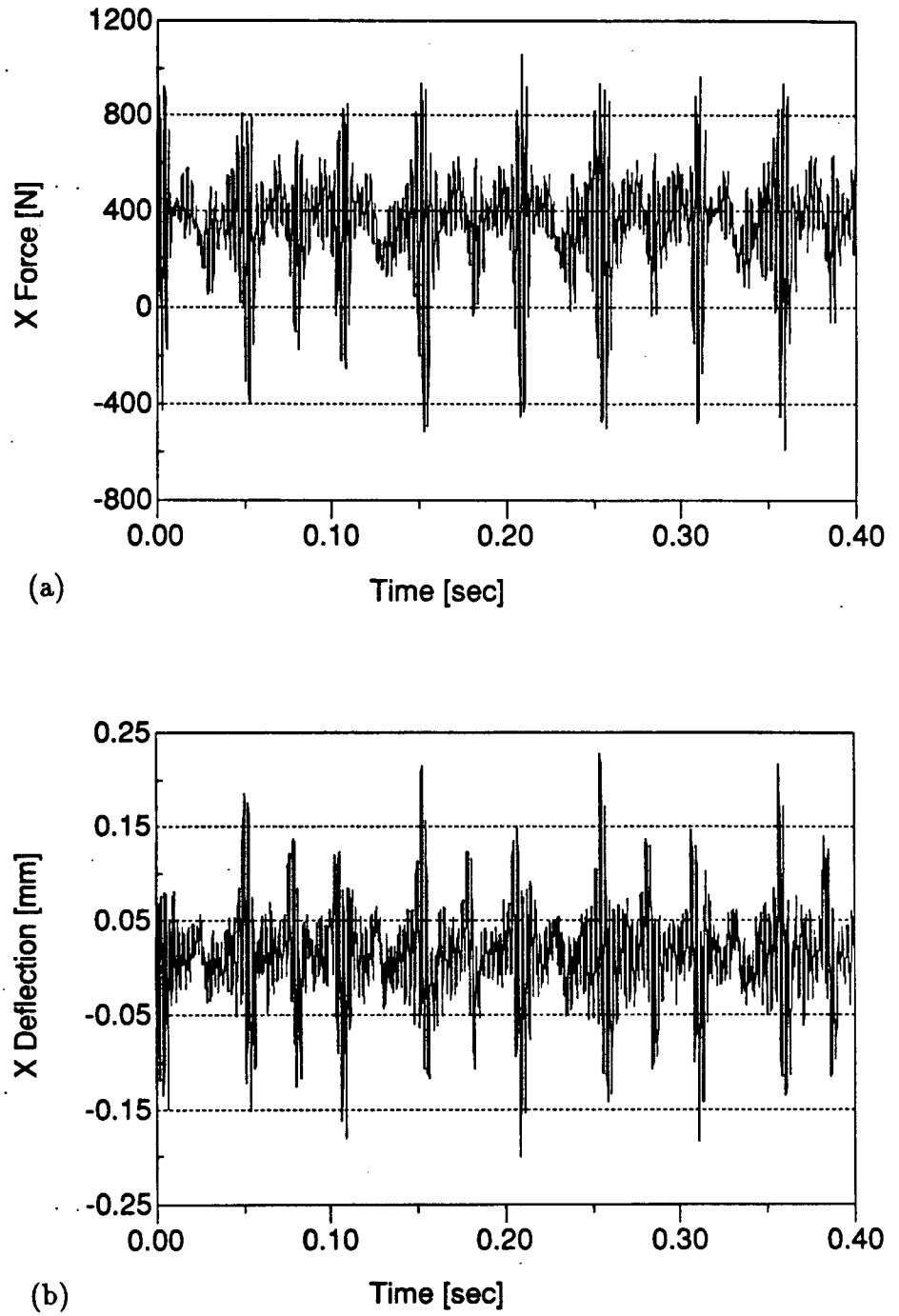


Figure 4.43: Experimental (X) cutting force and filtered (X) cutter deflection. $n = 1200$ [rpm], $\phi = 180^0$, $a = 2.54$ [mm].

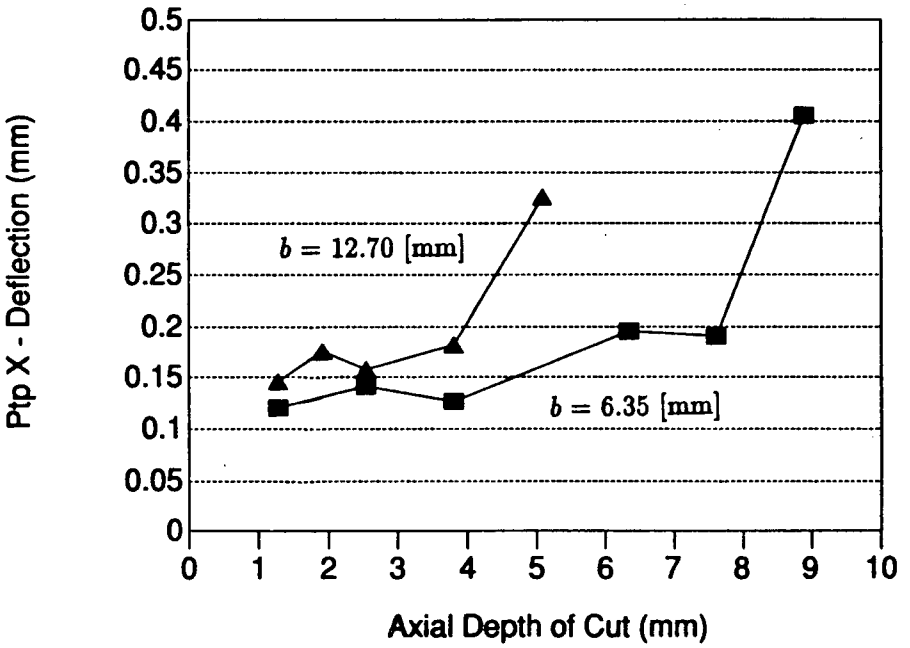


Figure 4.44: Experimental (X) cutter deflection versus depth of cut at (1/4) and (1/2) immersion. n = 600 [rpm], $\phi = 60^0$ and 90^0.

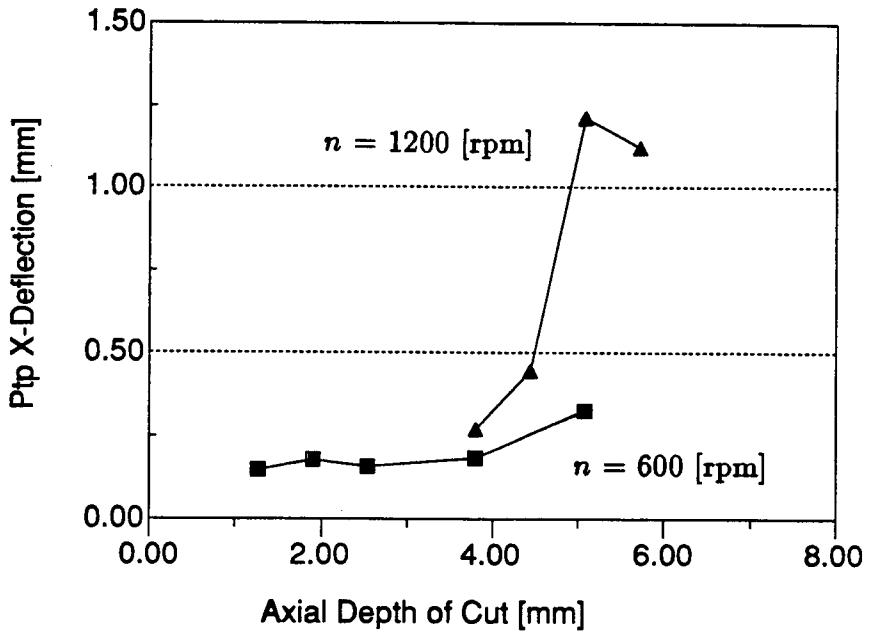


Figure 4.45: Experimental (X) cutter deflection versus depth of cut at 600 and 1200 [rpm]. n = 600 and 1200 [rpm], $\phi = 90^0$.

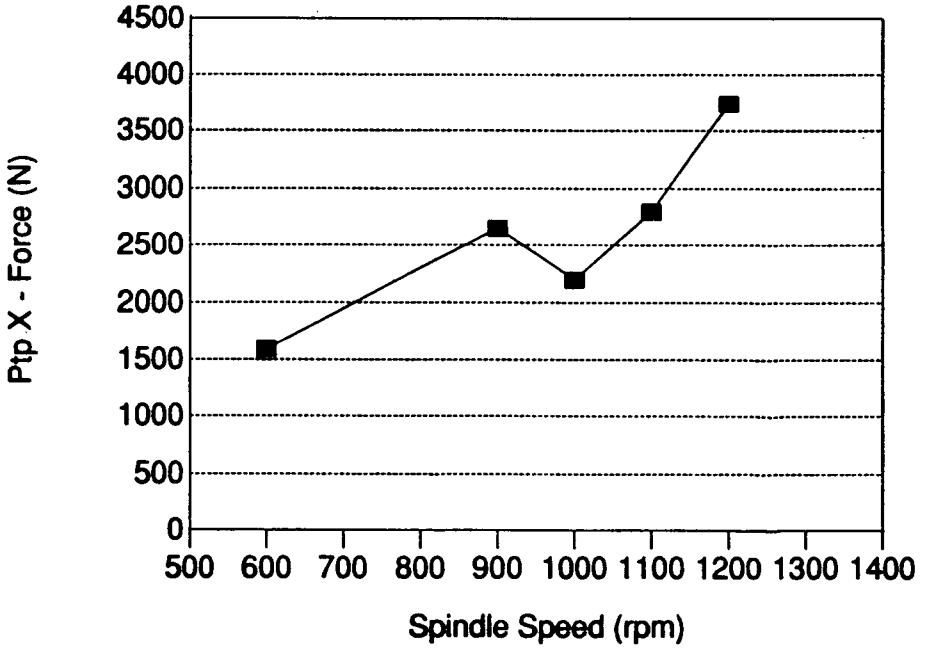


Figure 4.46: Experimental (X) cutting force versus spindle speed. $\phi = 112^0$, $a = 3.81$ [mm].

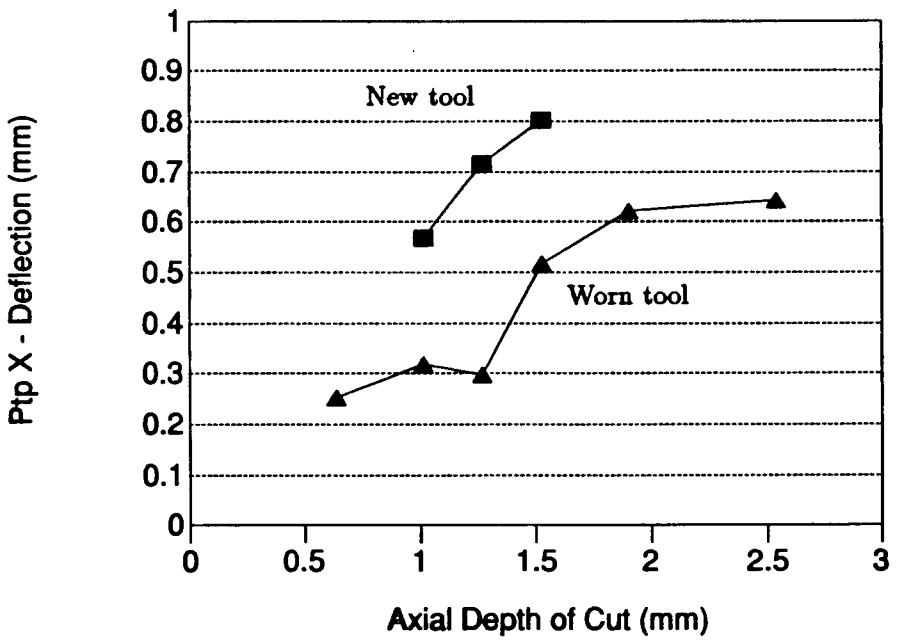


Figure 4.47: Experimental (X) cutter deflection for new and worn tool. $n = 1200$ [rpm], $\phi = 180^0$.

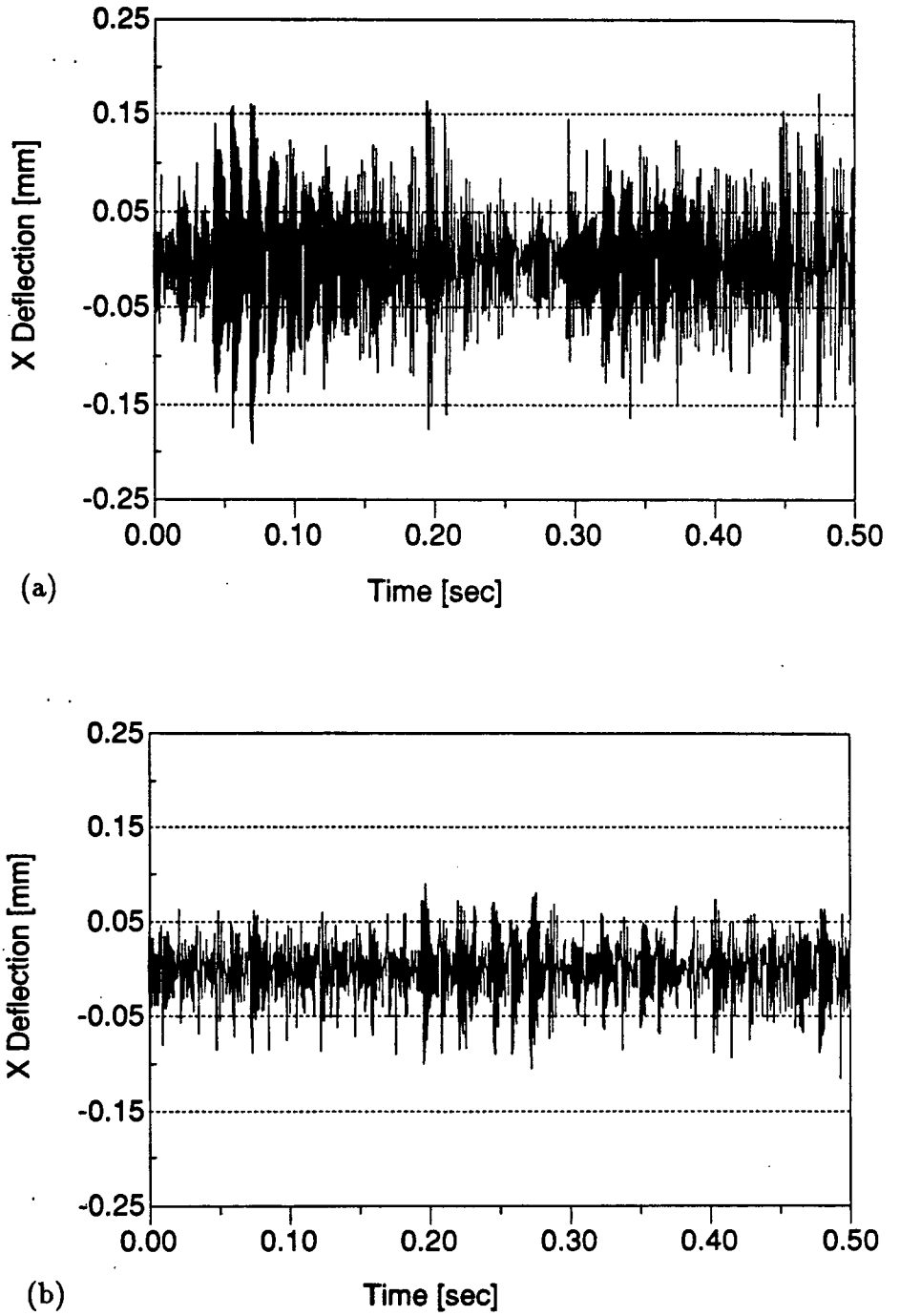
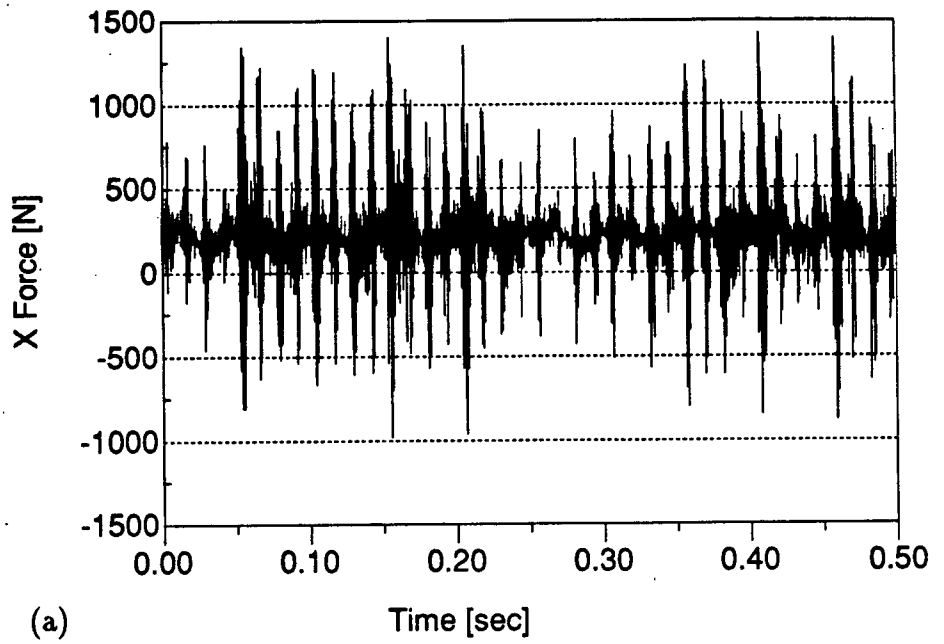


Figure 4.48: Experimental (X) cutter deflection comparison between fixed and variable speed. $n_o = 1200$ [rpm], $\phi = 104^0$, $a = 1.27$ [mm].



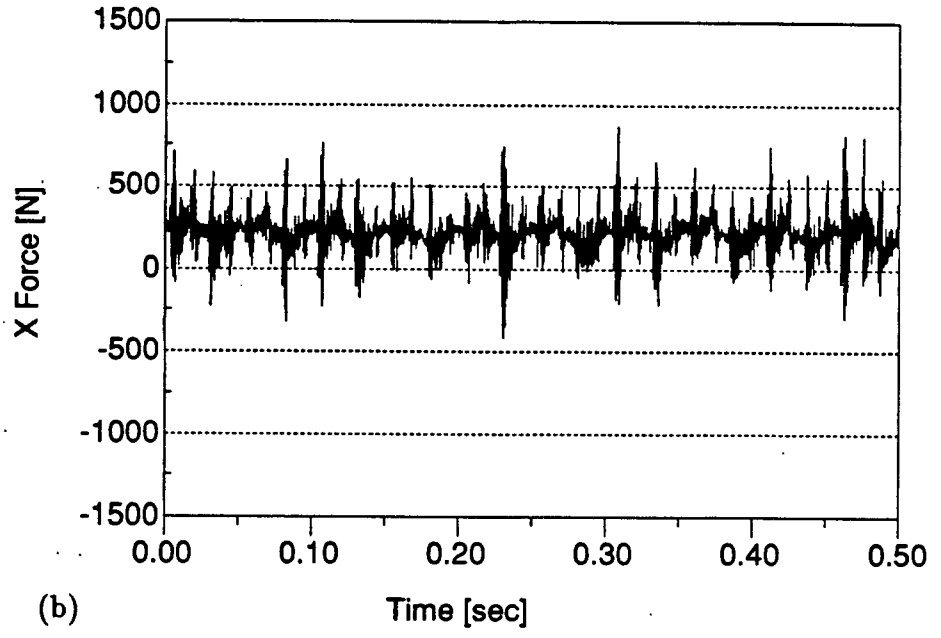
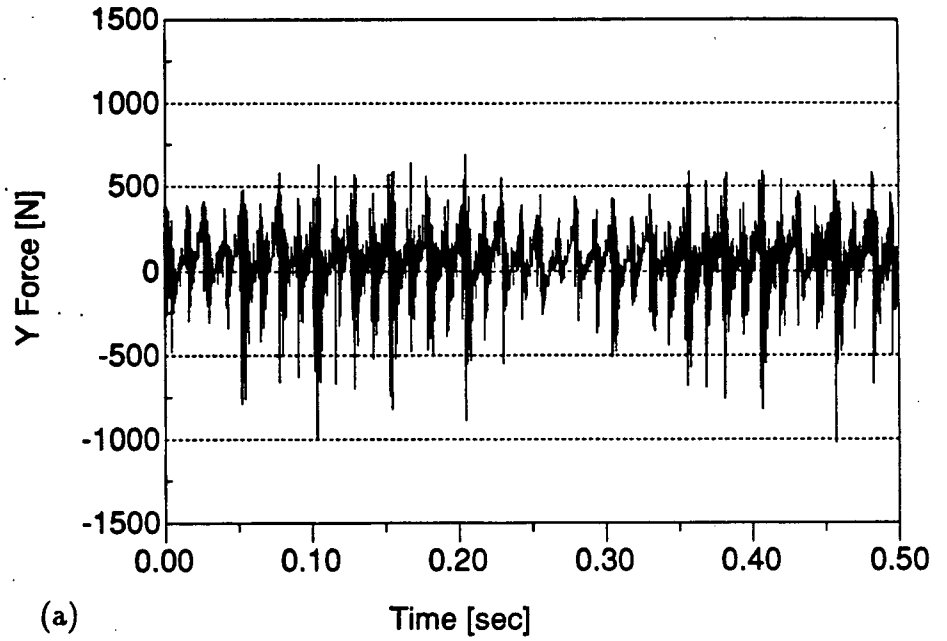


Figure 4.49: Experimental (X) cutting force comparison between fixed and variable speed. $n_o = 1200$ [rpm], $\phi = 104^0$, $a = 1.27$ [mm].



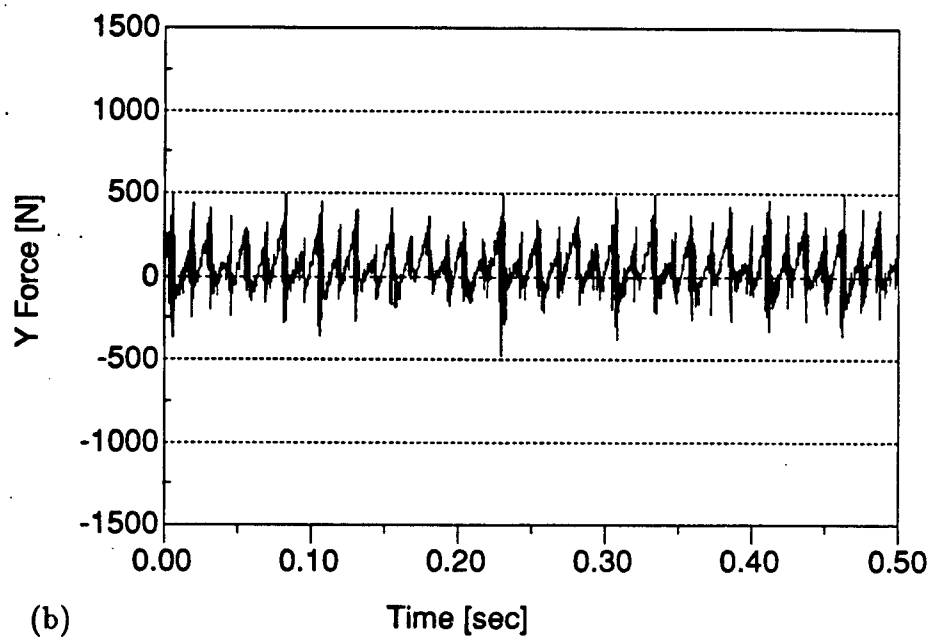


Figure 4.50: Experimental (Y) cutting force comparison between fixed and variable speed. $n_o = 1200$ [rpm], $\phi = 104^0$, $a = 1.27$ [mm].

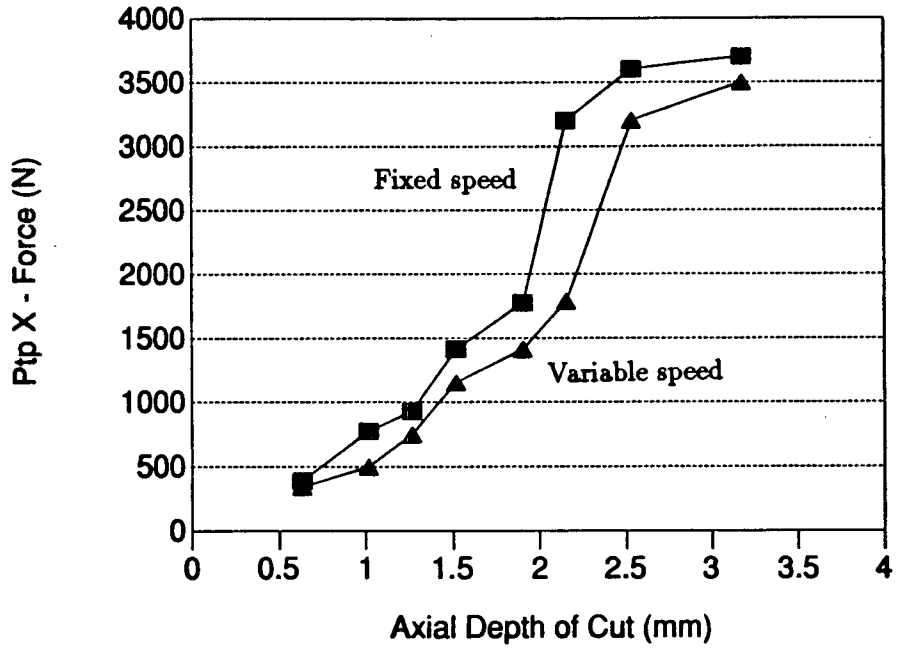


Figure 4.51: Experimental (X) cutting force versus depth of cut for fixed and variable speed. $n_o = 1200$ [rpm], $\phi = 180^0$.

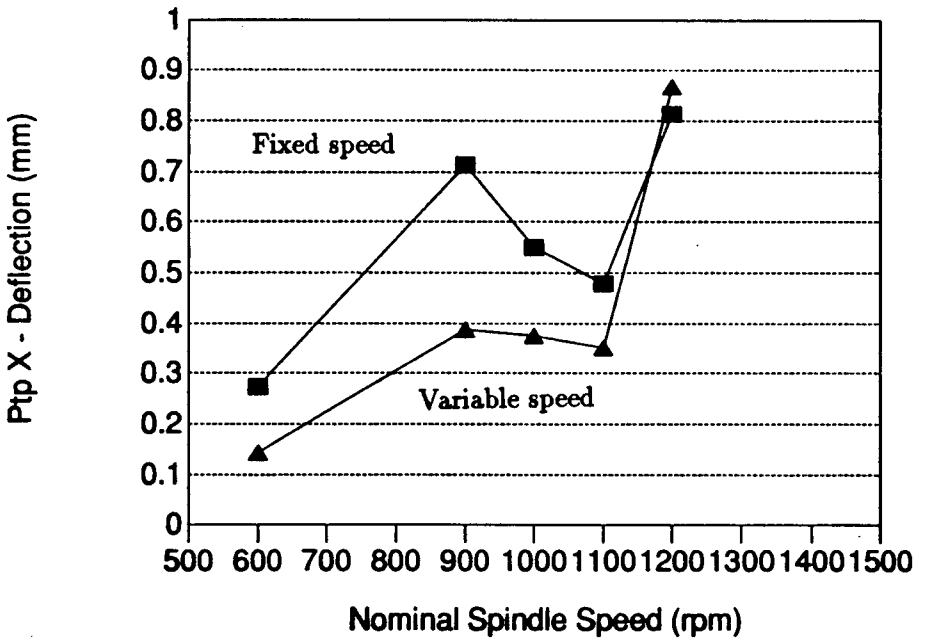


Figure 4.52: Experimental (X) cutter deflection versus speed for fixed and variable speed. $\phi = 112^0$, $a = 3.81$ [mm].

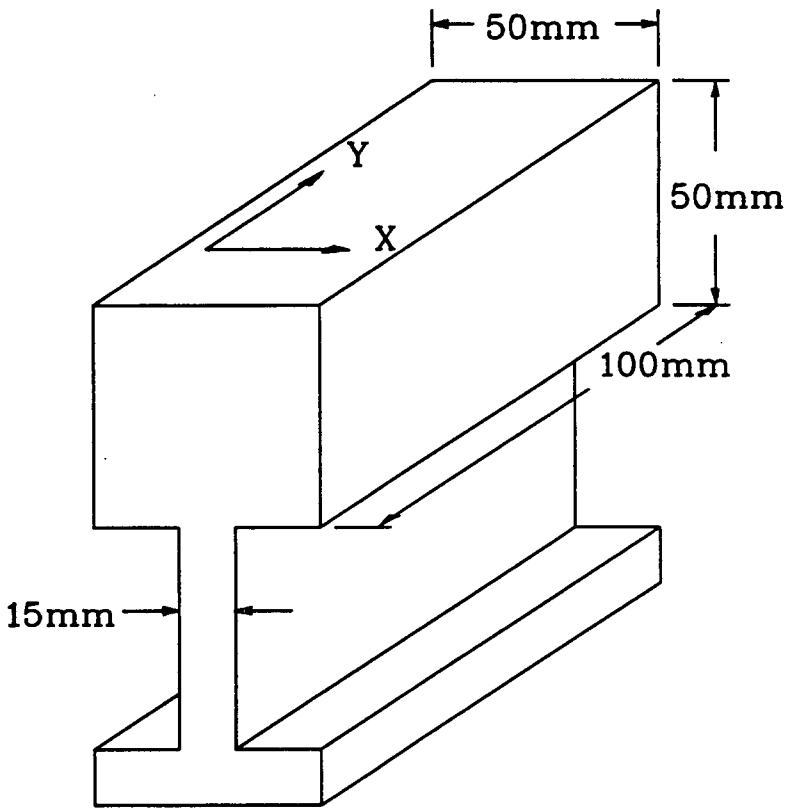
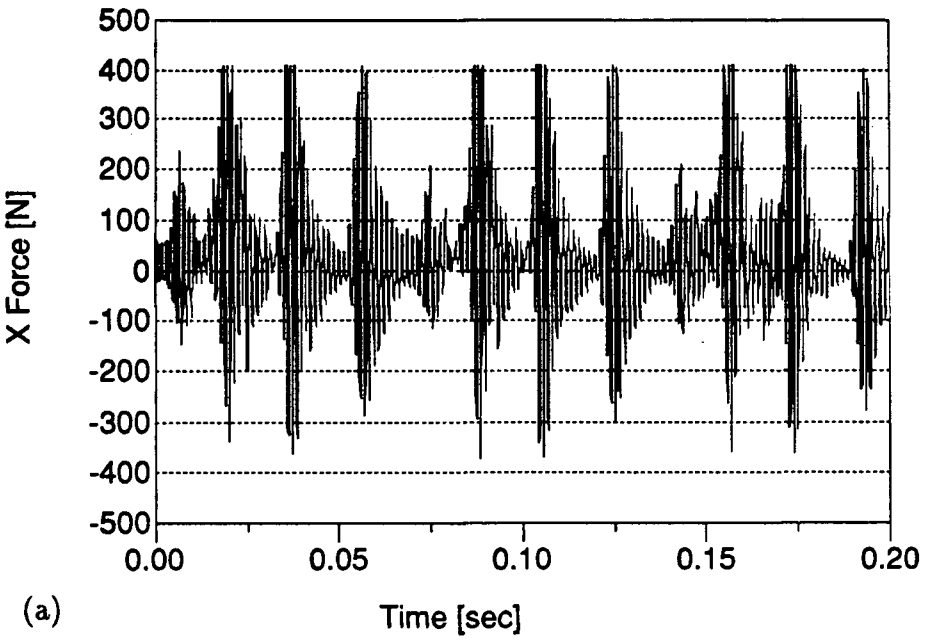
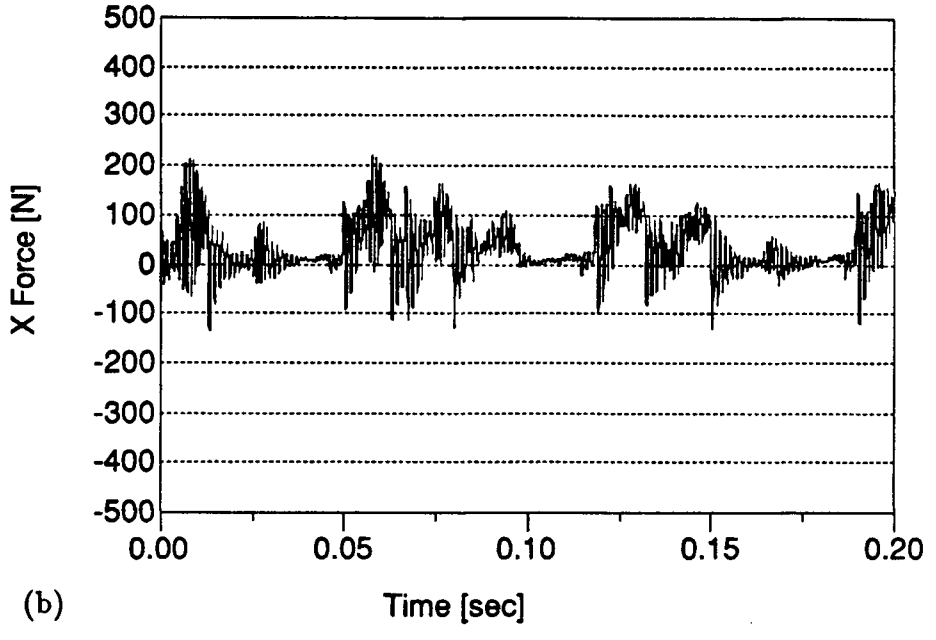


Figure 4.53: Single degree of freedom workpiece.

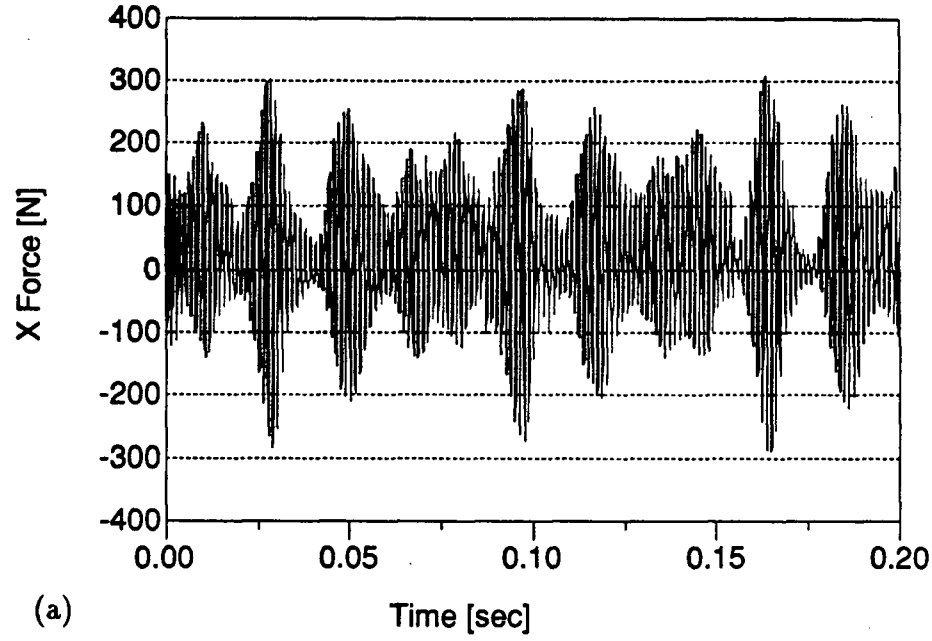


(a)

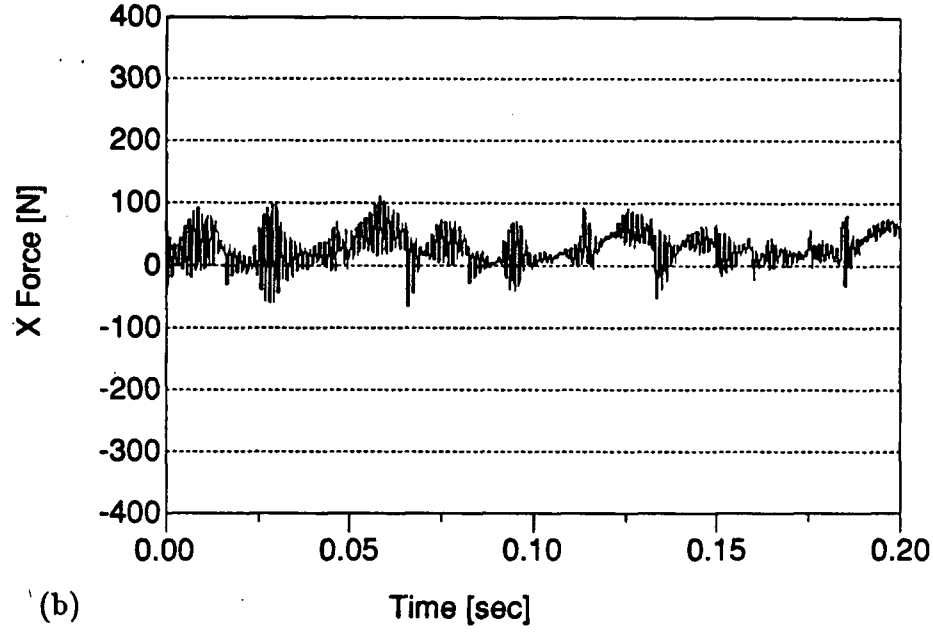


(b)

Figure 4.54: Experimental (X) cutting force for fixed and variable speed. $n_o = 900$ [rpm], $\phi = 60^0$, $a = 6.35$ [mm].

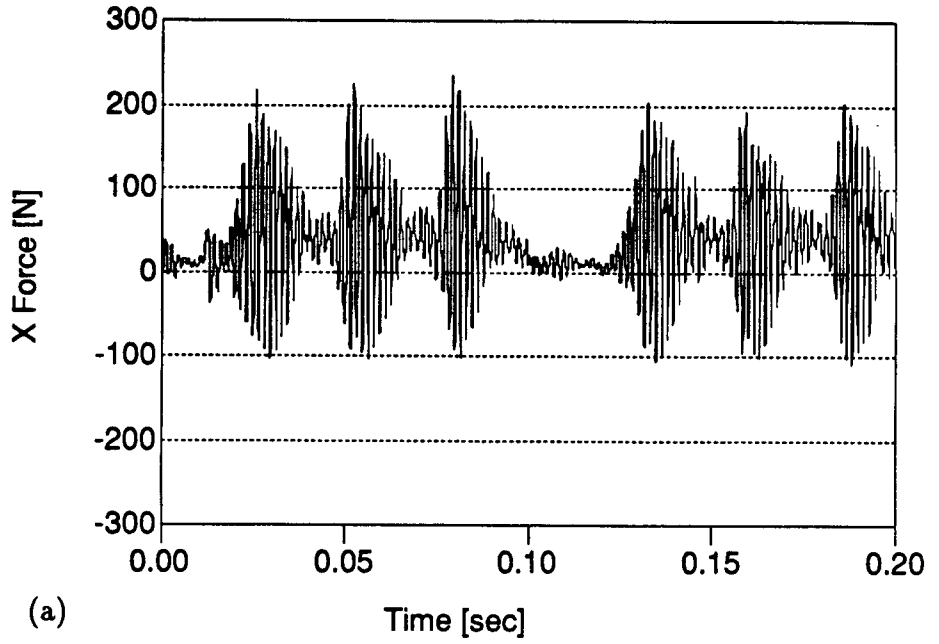


(a)



(b)

Figure 4.55: Experimental (X) cutting force for fixed and variable speed. $n_o = 900$ [rpm], $\phi = 90^0$, $a = 3.18$ [mm].



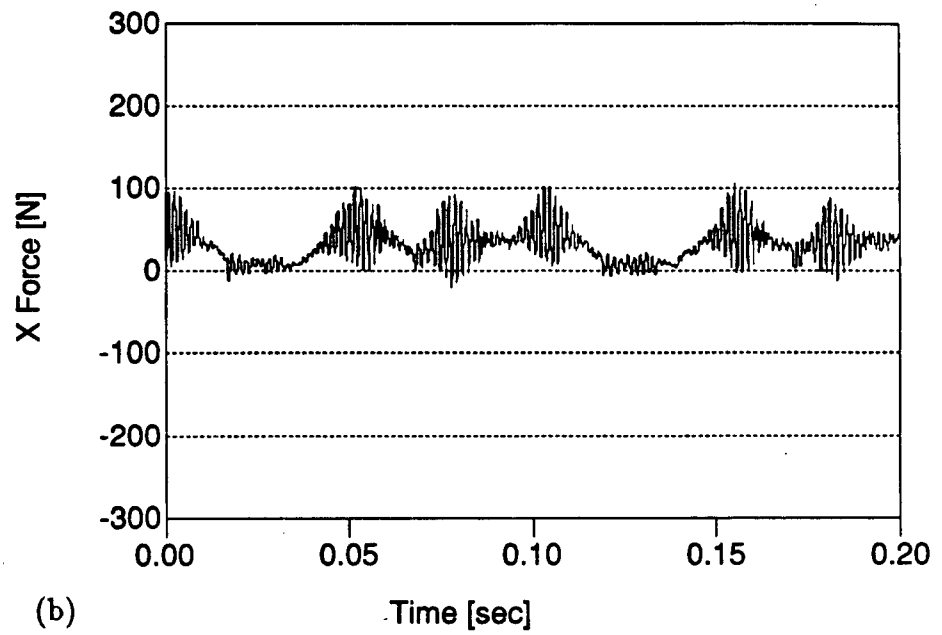


Figure 4.56: Experimental (X) cutting force for fixed and variable speed. $n_o = 600$ [rpm], $\phi = 105^0$, $a = 2.54$ [mm].

Chapter 5

In Process Chatter Detection and Avoidance

5.1 Introduction

This chapter describes the application of the proposed in process detection and avoidance of chatter. Results from previous chapters are used to develop the complete chatter avoidance algorithm. Experimental findings for face milling and end milling are presented after a brief discussion on chatter detection and the chatter avoidance algorithm.

5.2 In Process Chatter Detection

The first problem that has to be solved in this final experimental stage is chatter detection. The most direct way is to use the time or frequency domain signal of the force or deflection to indicate the presence of chatter. The possibility of which signal to use depends largely on which cutter-workpiece combination is used. The combination of a flexible end mill machining a rigid workpiece is considered first.

The use of time domain signal requires some knowledge of the width and depth of cut. The maximum static chip load and resulting force and deflection can be calculated from this data. Then, a chatter threshold value can be established as a certain percentage of the maximum static value of the cutting forces and cutter displacements. This method is satisfactory except for a couple of practical problems. First, the dynamometer's measurement bandwidth is not always reliable with the chatter frequencies produced by this cutter-workpiece combination, forcing the need to use the deflection signal. Dynamic

cutter deflections are measured using a noncontact proximity sensor, aligned stationary but parallel to the sleeve mounted on the cutter shank. The sensor has a measurement bandwidth of approximately 2000 [Hz]. The deflection measurement, in this case, has to be fed through a highpass filter because the end mill sleeve is not perfectly concentric. Although the runout of the sleeve is not that (about 0.1 [mm]), it is significant relative to end mill deflections. Once the signal is highpass filtered, the static deflection diminishes. Therefore, a chatter threshold must be based on dynamic deflections; this is still possible if not for the second problem, cutter runout. Cutter runout results in spikes in the force and deflection, which make objective correlation with stability difficult.

The other alternative is to make use of the frequency spectrum of the deflection. Since the available hardware is capable of on-line Fourier transform computations, long processing time is not a problem. The problem becomes the recognition of chatter from the frequency spectrums alone. Once again, runout of the sleeve and runout of the cutter make chatter detection more difficult. In the absence of runout and flank interference, chatter is always characterized by a large peak close to the tool natural frequencies. Figure 5.1(a) shows the (X) deflection spectrum for a 60 degree immersion up milling cut at 600 [rpm] and a depth of 6.35 [mm]. The process was stable from observations during cutting. Although there are peaks near the tool natural frequencies at 500 [Hz] to 600 [Hz], their magnitudes are only comparable to the tooth harmonic magnitudes. On the other hand, Figure 5.1(b) shows a cut with the same conditions but an axial depth of cut of 8.90 [mm]. The cut was unstable and it is evident from the spectrum. The chatter peak has a much larger magnitude than the magnitudes of the tooth harmonics around it. Therefore, a chatter threshold can be based on a comparison of the strength of the chatter frequency with the strength of neighboring frequencies.

The same method can still be used with the presence of runouts and flank interference, but there is greater difficulty in determining a proper threshold. The strength of tooth

harmonics is largely influenced by the runouts, flank interference, and also the highpass filter characteristics. Since the sleeve runout and the analog filter remains unchanged, they do not represent a major problem. The influence of cutter runout and process damping, however, differ between cuts. Figure 5.2 illustrates this problem. Figure 5.2 contains the (X) deflection spectrum for three cuts made at 1200 [rpm] with Tool 2. They are all full immersion cuts with 0.1 [mm] feed per tooth. Figure 5.2(a) shows the spectrum for a cutting depth of 1.27 [mm] . From observations during machining, this cut was stable and had peak-to-peak (X) forces of about 930 [N]. The spectrum of the cut when the depth is increased to 1.9 [mm] is shown in Figure 5.2(b). The spectrum has a fairly sharp peak at approximately 750 [Hz], a sign usually indicative of chatter. However, peak-to-peak (X) forces recorded during machining was only about 1700 [N], and the cut was observed to be stable. Chatter was not actually reached until a cutting depth of 2.54 [mm], which is shown in Figure 5.2(c). The spectrum in Figure 5.2(c) has very strong tooth harmonics, but no significant peaks near the tool natural frequencies (500 [Hz] to 700 [Hz]). However, chatter was evident during cutting as peak-to-peak (X) forces reached magnitudes of 3600 [N]. Although there are some uncertainties with this method, it requires little information from the cut, and is still more viable than using time domain signals.

Using a rigid face mill with a flexible workpiece alleviates the situation since runout and flank interference no longer dominate the dynamics. In addition, dynamic cutting tests performed with this cutter-workpiece combination showed that the force signal was reliable. Following the same reasoning as above, the frequency domain signal is the preferred one to use. The force signal is the most convenient one to use, and experiments showed that very distinct changes in the spectrum are visible as chatter develops. Figure 5.3 contains the spectrums from full immersion cuts at 900 [rpm] using Tool 3. Figure 5.3(a) shows a stable cut where the axial depth is 1.27 [mm]. The high frequency peak

in this case has approximately the same magnitude as the tooth frequency. However, when the depth is increased to 1.40 [mm], chatter develops and the resulting spectrum (in Figure 5.3(b)) is indicative of the instability. The growth of a dominant peak close to the tool natural frequencies is easily distinguishable.

Another way of detecting chatter is to analyze the sound pressure, a method recently applied by Smith et al. [16]. The sound emitted from cutting is normally indicative of the vibration present. As with the deflection or force signal, the frequency spectrum of the sound pressure is more appropriate to use than the time domain signal. The advantage of using a microphone to detect chatter is that it is not in the way of actual cutting. The dynamometer and proximity sensor has to be in the cutting environment. In the case of using force signals, the workpiece must be mounted on the dynamometer. Using deflection signals requires the proximity sensor to be very close to the end mill. With the available sensor technology, it is difficult to use the force or the deflection methods in a practical cutting situation. Thus the remote nature of the microphone makes it more practical to use in an actual shop environment. Significant background noise, however, is a problem that is associated with sound and not with the other two signals. A microphone is typically equally sensitive in all directions; thus ambient sounds are picked up as well. Although directional sensitivity can be achieved through proper processing of signals from two microphones, the narrowed range of sensitivity may still pick up noise from the spindle. In the present experimental set-up, the majority of the background noise comes from the spindle bearings which are in close proximity to the cutting area.

The recognition of chatter from sound pressure has similar difficulties as from deflection, since sound is propagated from vibrations. Thus significant cutter runout coupled with excessive flank interference can also cause accurate detection to be difficult. Nevertheless, chatter usually can be recognized from large peaks in the spectrum close to the natural frequencies. The typical contrast between stable and unstable cutting for

a cutter with minimal runout and flank interference is shown in Figure 5.4. The plots show the sound pressure spectrum for full immersion cuts at 900 [rpm] using Tool 3. Figure 5.4(a) shows the sound spectrum for a stable cut with a cutting depth of 0.76 [mm]. The low frequency signal is characterized by a large offset. Otherwise, the signal smoothly drops in magnitude until about 400 [Hz]. At that frequency, noise from the spindle bearing and cutting dynamics appear. The magnitude of the largest peak in this high frequency range, however, does not exceed the low frequency offset. Also notice the lack of any tooth frequencies or harmonics in the spectrum. Figure 5.4(b) contains the sound spectrum for an unstable cut when the axial depth of cut is increased to 1.02 [mm]. The low frequency signal up to 400 [Hz] is practically identical to that of Figure 5.4(a). However, chatter produces a large peak at about 450 [Hz]. The second harmonic of the vibration frequency can also be seen at 900 [Hz]. Since the low frequency signal (from 0 [Hz] to 300 [Hz]) is so similar between the stable and unstable spectrums, the average low frequency magnitude is an ideal reference value for automatic chatter detection. The peak magnitude in the high frequency portion of the spectrum (300 [Hz] and higher) can be compared with the average magnitude of the low frequency range. The low frequency average magnitude can be multiplied by a *threshold factor*, which is predetermined from cutting tests. If the magnitude of the chatter peak exceeds the low frequency average by a factor greater than the threshold factor, then chatter can be assumed to be present.

Thus the signal used for chatter detection in the following experiments is sound pressure. Although previous experiments by Rahman [12] showed that the vibration signal is a better predictor of cutting instability than force or sound, a remotely positioned microphone is more practically convenient. In addition, the lack of tooth harmonics and the resulting similarity in low frequency signal makes chatter detection more accurate. Before discussing results from the chatter detection and avoidance system, it is first necessary to briefly describe the chatter avoidance algorithm.

5.3 In Process Chatter Avoidance Algorithm

The chatter avoidance algorithm is executed by the IBM-compatible PC in the experimental set-up described in Section 4.4. The algorithm requires the user inputs of threshold factor, nominal spindle speed, and maximum speed variation parameters. It is responsible for controlling the data acquisition, signal processing, and variable speed spindle.

The proposed algorithm is outlined in Figure 5.5, and the program code can be found in Appendix A. The first step in the algorithm is execution of the data acquisition routine on the digital signal processing board. At this point, the sound pressure is sampled at a frequency which is approximately five times higher than the possible chatter frequency. While the signal indicative of milling stability is sampled, software processes the frequency spectrum from the previous iteration. The maximum magnitude and its frequency in the specified high frequency (i.e. chatter) range are determined. In addition, the average magnitude in the specified low frequency (i.e. chatter free) range is computed. The algorithm compares the average magnitude of the low frequency spectrum with the peak magnitude of the high frequency spectrum. Chatter is assumed to be present when the maximum magnitude exceeds the low frequency spectrum average by a factor greater than the threshold factor.

The next step begins as soon as the data acquisition initiated at the beginning of the algorithm is completed. The newly acquired time domain signal of the sound spectrum is multiplied by the Hanning window to minimize end effects during subsequent Fourier transform processing. While fast Fourier transform computations are taking place on the present sound signal, necessary control signals are sent to the variable spindle drive. If chatter was detected from the earlier signal processing, then speed variation signals are sent to the spindle speed controller, otherwise a constant speed signal is sent. Upon completion of fast Fourier transform computations, the algorithm initiates data acquisition

on the digital signal processing hardware once again, and the cycle repeats.

The total execution time of one iteration depends primarily on the desired frequency resolution of the resulting spectrum. The sampling frequency is related to the frequency resolution by,

$$\begin{aligned} \Delta f &= \frac{f_s}{N} \\ \text{where } \Delta f &= \text{frequency resolution of frequency spectrum} \\ f_s &= \text{sampling frequency} \\ N &= \text{the number of samples, usually a power of 2.} \end{aligned}$$

For example, in order to obtain 5 Hz resolution from a 512 point spectrum, the sampling frequency has to be no greater than 2560 Hz. Acquiring 512 points at this frequency requires 200 milliseconds. Therefore, a single iteration cannot be accomplished in less than this time. Of course, there are also additional computational times that must be included in the total execution time of one iteration.

5.4 Experimental Chatter Avoidance Results

The milling tests shown in this section are for the purpose of evaluating the chatter avoidance system. Figures are usually organized so that cuts performed with the guidance of the system can be compared with fixed speed cuts on the same page. The signal shown for all cuts is the (X) cutting force time history. All cuts taken in this section are either up milling or full immersion step cuts. The axial depth of cut starts at one value, but abruptly increases part way through the cut. Thus the ability of the chatter detection algorithm can be tested. The responses of the chatter avoidance milling tests of course show both fixed speed and variable speed cutting. During stable cutting, the system sends only constant speed signals to the spindle. However, once chatter is detected by

the algorithm, variable speed cutting commences in order to suppress the instability. The speed variation used in most of these cutting tests is 50 [rpm] variation amplitude and 2.5 [Hz] variation frequency. The sampling frequency of the cutting tests is 3600 [Hz], which results in a 1800 [Hz] frequency spectrum. The low frequency average is taken between 0 [Hz] and 300 [Hz], while the chatter peak is searched from 300 [Hz] to 1800 [Hz].

The following experimental results shown in Figure 5.6 were obtained using Tool 2. The full immersion cuts were taken with a nominal speed of 1200 [rpm]. The axial depth of cut started at 1.27 [mm] and changed abruptly to 2.54 [mm] part way through the cut. The resulting (X) force history for the fixed speed cut is shown in Figure 5.6(a). Since a long length of time signal is contained in the plot, the details of the dynamic signal is not distinguishable. However, the relative stability of the cut is evident from peak-to-peak force amplitudes. There is stable cutting up to approximately 1.0 [sec] when the depth is still 1.27 [mm]. The force fluctuations here are primarily a result of cutter runout. Once the cutting depth increases to 2.54 [mm], however, instability develops and chatter is present. The peak-to-peak forces of approximately 1200 [N] present after about 1.2 [sec] are the result of self-excited vibration.

Figure 5.6(b) shows the same cut performed under the chatter detection and avoidance system with the threshold factor set at 6.0. The (X) force time data in Figure 5.6(b) show that more stability than the fixed speed cut (in Figure 5.6(a)) is maintained throughout the cut. The peak-to-peak force during the 1.27 [mm] depth of cut has the same magnitude as the previous cut since they are performed at the same fixed speed. However, the algorithm detects chatter when the depth increases to 2.54 [mm] (at 1.5 [sec] in Figure 5.6(b)). (It should be pointed out that the time reference of the plots presented in the following discussion is just for relative comparison within the plot. They do not correspond to any particular reference point). Spindle speed oscillation is initiated at

this time and chatter is suppressed. The maximum peak-to-peak force during the axial depth of 2.54 [mm] is no more than 800 [N], considerably less than the fixed speed cut. Actually, nominal force fluctuations are only about 300 [N] (i.e. primarily due to runout) with only a few spikes greater than 600 [N].

The previous two plots show the success of the chatter avoidance system based on spindle speed oscillation. However, they do not suggest that the system succeeds on every occasion. The problems described in Section 4.5.2 are all still present here. The effect of runout, excessive process damping, tool wear, and spindle speed variation limitations have not diminished in these tests. The additional problem of establishing a proper chatter threshold is present here as well. To assess the system more properly, chatter avoidance tests were carried out using rigid cutters and single degree of freedom workpieces as well. The cutter-workpiece combinations used are listed under Tool 3 and Tool 4 in Table 4.2. Some satisfactory results were obtained, but the experimental set-up had significant limitations and thus full potential of variable speed milling was not achieved. Problems associated with the experimental equipment will be discussed later. First, some results of this series of tests are presented, and some observations are made.

Figure 5.7 shows the extended (X) force time history of two stepped cuts, using the 50.0 [mm] face mill and workpiece combination listed under Tool 3 in Table 4.2. The depth of cut changes from 1.27 [mm] to 2.54 [mm], while the immersion angle is 112 degrees. Figure 5.7(a) shows the fixed cutting speed response at the nominal speed of 600 [rpm]. Although it is difficult to determine from Figure 5.7(a), the axial depth of cut changes at 1.2 [sec]. Chatter, however, does not start to develop until 1.8 [sec]. The lack of chatter immediately after the step change may be a result of workpiece dynamics. The cantilevered cutting surface of the workpiece produces additional deflection in the axial direction instead of just in the feed direction. These axial deflections may change the actual axial depth of cut, and influence stability at certain points on the workpiece.

After chatter develops for this fixed speed cut, peak-to-peak forces grow to almost 3200 [N] (between 2.0 [sec] and 2.5 [sec]) before levelling off to 2500 [N] peak-to-peak.

The same cut performed under the chatter avoidance system is shown in Figure 5.7(b). The threshold factor is set at 10.0 for this test. The first part of the plot is very similar to the fixed speed plot in Figure 5.7(a). The step change in cutting depth is encountered at 1.5 [sec], but instability does not develop until 2.0 [sec]. At this time, however, chatter is detected from the sound spectrum and speed oscillations are imposed. Stabilization is not immediate in this case as it was in the case shown in Figure 5.6. Dynamic forces continue to grow between 2.0 [sec] and 2.6 [sec]. However, the increase is not as rapid as the fixed speed cut. In addition, the maximum peak-to-peak force during the interval of increasing forces is approximately 2700 [N], about 500 [N] less than the fixed speed cut. Figure 5.7(b) shows that stability is completely restored (at 3.5 [sec]) after about 2.6 [sec] of variable speed cutting. Cutting after 3.5 [sec] for the cutting depth of 2.54 [mm] is even more stable than at 1.27 [mm].

Figure 5.8 provides a detailed comparison of the (X) forces before chatter develops for the two cases shown in Figure 5.7. Figure 5.8(a) shows a 0.2 [sec] segment of data from Figure 5.7(a) starting at about 1.5 [sec], while Figure 5.8(b) shows a 0.2 [sec] span of data from Figure 5.7(b) starting at approximately 1.5 [sec]. The two plots in Figure 5.8 are very similar since they are cuts of the same depth and with the same speed. There is noticeable dynamic forces, but the static modulation with the tooth period is still visible. Figure 5.9 compares the (X) forces from Figure 5.7 after chatter has developed. Figure 5.9(a) is taken from Figure 5.7(a) at about 3.5 [sec], and Figure 5.9(b) is taken from Figure 5.7(b) at about 3.0 [sec]. Figure 5.9(a) shows fully developed chatter for the fixed speed case. Figure 5.9(b), on the other hand, shows stable cutting as a result of variable speed. There is minimal dynamics and force fluctuations are primarily a result of tooth entry and exit transients.

The next figure illustrates the effect of early chatter detection. Figure 5.10 shows three full immersion cuts made by a 25.4 [mm] end mill and flexible workpiece as listed under Tool 4 in Table 4.2. The cutting depth increases from 0.64 [mm] to 1.27 [mm]. The fixed speed cut with a nominal speed of 1200 [rpm] is shown in Figure 5.10(a). The step change is encountered at approximately 2.7 [sec]. Chatter develops soon after the step change and force fluctuations grow to 900 [N] peak-to-peak. Figure 5.10(b) shows the chatter avoidance response when the threshold factor is set at 4.0 and variable speed is subsequently triggered when chatter develops. The step change in depth to 1.27 [mm] is reached at 3.3 [sec], when chatter immediately develops. Speed oscillation is initiated at about the same time (3.3 [sec] in Figure 5.10(b)). Forces are brought back down to a reasonable level of 300 [N] peak-to-peak after about 2.5 [sec] (at 6.0 [sec]). However, before full stabilization is achieved, peak-to-peak forces jump to 1000 [N]. Figure 5.10(c) shows a similar cut, but with speed oscillation triggered early at approximately 1.8 [sec], about 1.0 [sec] before the step change occurred. The threshold factor was set too low at a value of 3.0; thus chatter was assumed after the relatively large force and resulting noise at 1.8 [sec]. Early triggering results in the most stable response amongst the three cases. Forces rarely exceed 300 [N] peak-to-peak. Early initialization of speed oscillation in Figure 5.10(c) prevents wave regeneration from developing chatter, as opposed to stabilizing vibrations after it has become unstable (in Figure 5.10(b)). Figure 5.11 shows short time segments of the forces shown in Figure 5.10. Figure 5.11(a) shows 0.2 [sec] of force data taken from the fixed speed case in Figure 5.10(a) at 7.0 [sec]. Besides having large peak-to-peak amplitudes, the force is also very vibratory. A 0.2 [sec] segment taken from Figure 5.10(b) at 7.0 [sec] is shown in Figure 5.11(b). The cut is stable, and periodic modulation due to runout can be seen. Figure 5.11(c) shows a similar segment of (X) force data taken from Figure 5.10(c) at 7.0 [sec]. The response of Figure 5.11(c) is very similar to the one shown in Figure 5.11(b). This shows that despite immediate

stabilization in the early detection case, stability of the two cuts shown in Figure 5.10(b) and Figure 5.10(c) are eventually the same.

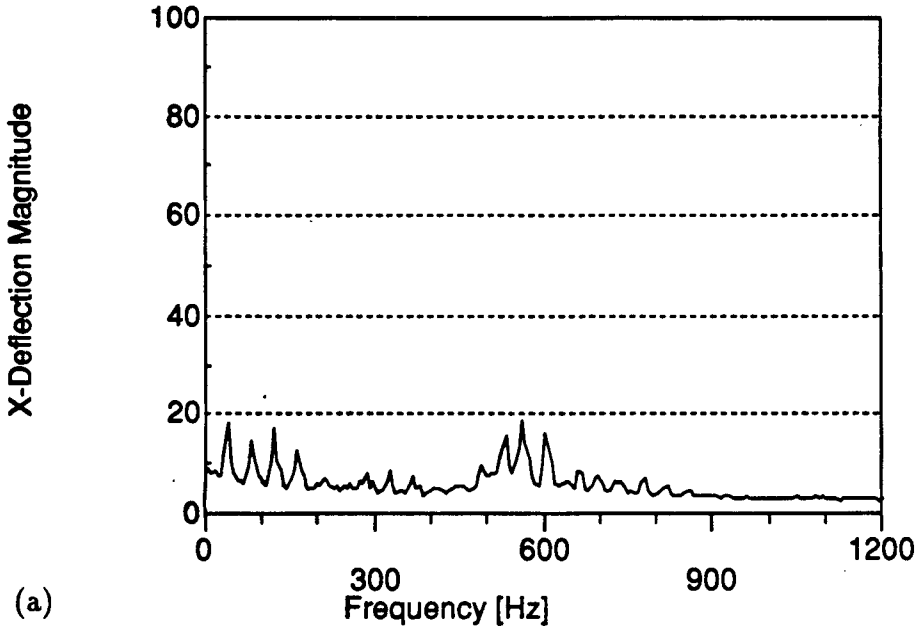
The last set of plots shows the effect of large speed variations. The figure compares 120 degree immersion responses for a step cut of 0.64 [mm] and 1.27 [mm] using Tool 4. Figure 5.12(a) shows the results of a fixed speed cut with a nominal spindle speed of 900 [rpm]. The step change in depth of cut is encountered at 0.4 [sec]. The force immediately becomes unstable and peak-to-peak forces reach 1500 [N]. Figure 5.12(b) shows the same cut performed under the control of the chatter avoidance system with variation amplitude of 50 [rpm] and variation frequency of 2.5 [Hz]. The threshold factor for this and the next two cuts is set at 5.0. Although total stabilization is not achieved, the cut with speed variation imposed once chatter develops is a little more stable than the fixed speed cut in Figure 5.12(a). Force fluctuations are approximately 1200 [N] peak-to-peak. Figure 5.12(c) shows the response for a cut performed under the chatter avoidance system with variation amplitude of 50 [rpm] and variation frequency of 3.0 [Hz]. The step is again encountered at 0.4 [sec], and variable speed is triggered at that time. Comparison of Figure 5.12(c) with Figure 5.12(b) shows that the increase in variation frequency does result in a decrease in peak-to-peak forces. Although the improvement is small, it is noticeable; forces have dropped to about 1000 [N] peak-to-peak in Figure 5.12(c). The response in Figure 5.12(d) is for a similar cut, but with speed variation of 60 [rpm] and 3.0 [Hz]. Once again, there is a decrease in peak-to-peak forces, especially from 0.5 [sec] to 2.0 [sec] where force fluctuations are approximately 600 [N]. Forces grow slightly after 2.0 [sec] to approximately 800 [N] peak-to-peak. In summary, there is a 200 [N] decrease from Figure 5.12(b) to Figure 5.12(c), and another 200 [N] decrease from Figure 5.12(c) to Figure 5.12(d) . Although increases in stability are subtle, they are significant since the differences in speed variation are small as well. The difference in variation frequency is only 0.5 [Hz] between the cases in Figure 5.12(b) and Figure 5.12(c), and the difference

in variation amplitude is only 10 [rpm] between the cases in Figure 5.12(c) and Figure 5.12(d). Larger speed variations are sure to increase stability even more and even faster. However, the torque limit of the spindle motor was almost reached with the variation of 3.0 [Hz] and 60 [rpm], and further increases stalled the spindle motor.

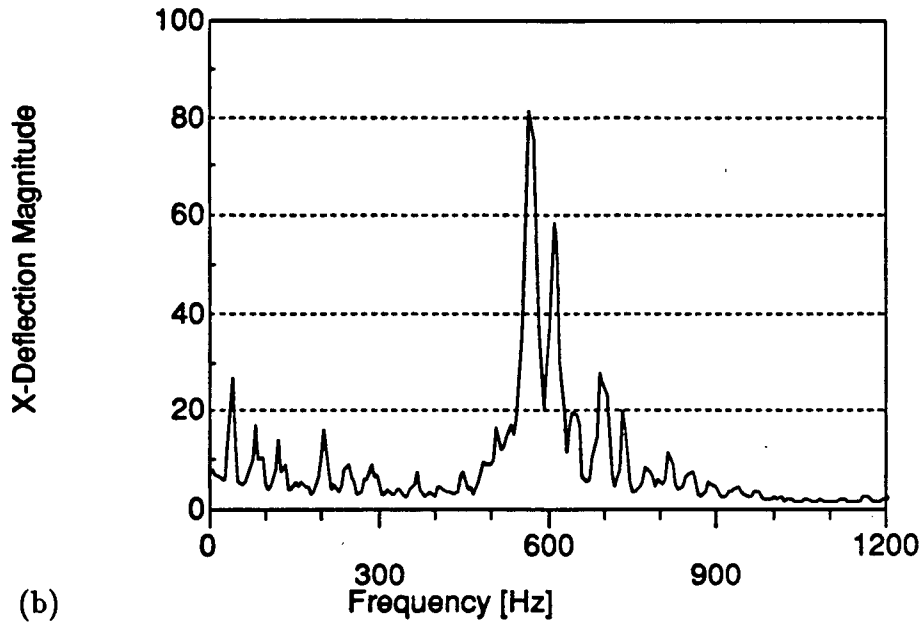
With all the variable speed cutting tests performed in this and the previous chapter, the low torque limit of the variable spindle proved to be the most restricting limitation. The effect of this limitation was not as apparent in controlled variable speed tests in Chapter 4. Cutting tests in that chapter were performed with variable speed from start to finish of a cut; thus the process often did not have an opportunity to become unstable. However, chatter avoidance experiments in this chapter necessitated variable speed to stabilize cutting after the fixed speed cut had become unstable. In order to bring back stability in a reasonably short time, larger speed variations than the maximum limit are often necessary.

5.5 Conclusions

In whole, the chatter detection and avoidance tests did produce some favorable comparisons with fixed speed cuts. Results, however, were hampered by the torque limit, and the large time constant of the spindle drive system. With the speed variations allowed by the motor, stabilization often took 1.0 [sec] to 2.0 [sec], although immediate stabilization was seen as well. Early chatter detection proved to be beneficial for variable speed chatter suppression. The benefits of large speed variations surfaced in these tests as observed from the simulation and experimental results presented in Chapter 3 and Chapter 4.



(a)



(b)

Figure 5.1: Experimental (X) cutter deflection spectrum of stable and unstable cut. n = 600 [rpm], $\phi = 60^0$, a = (a) 6.35, (b) 8.90 [mm].

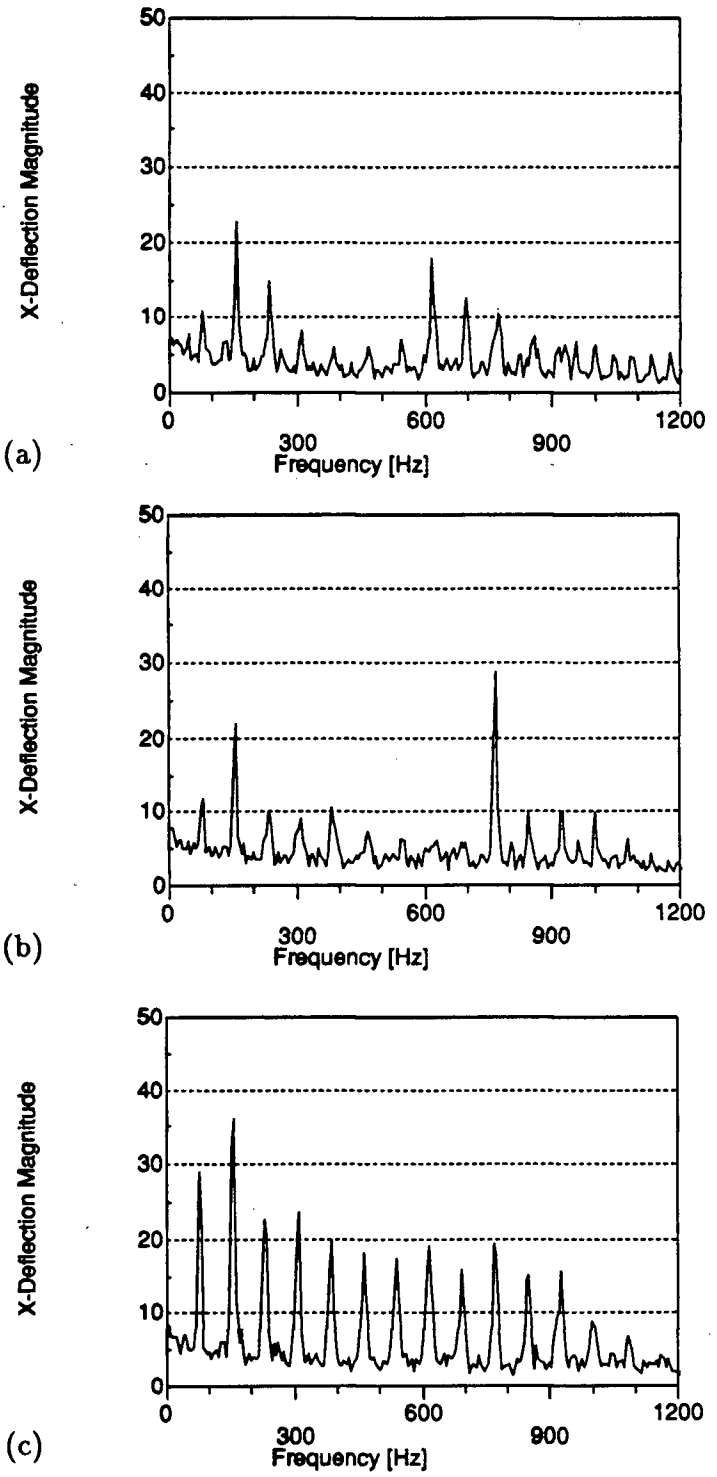
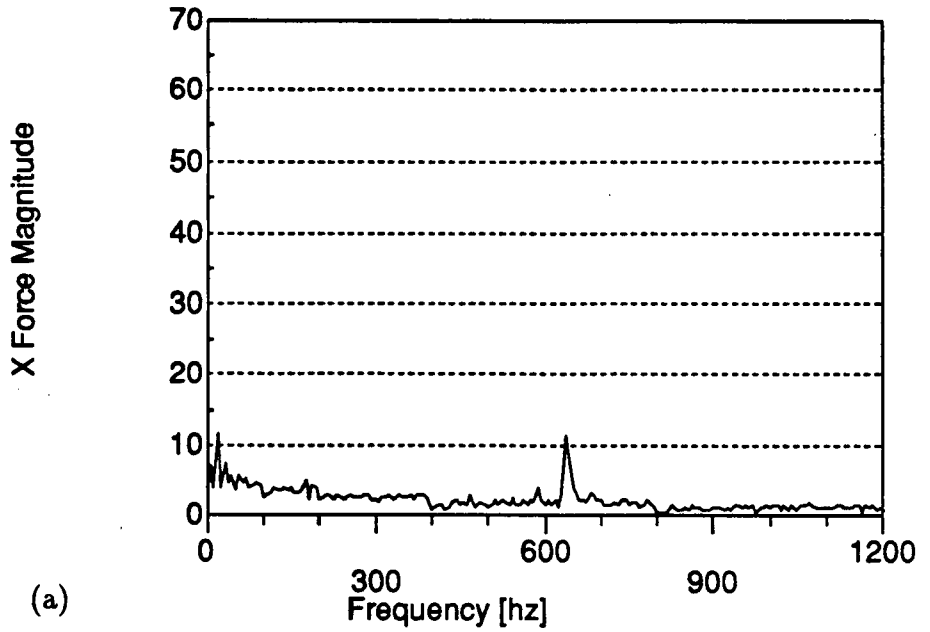
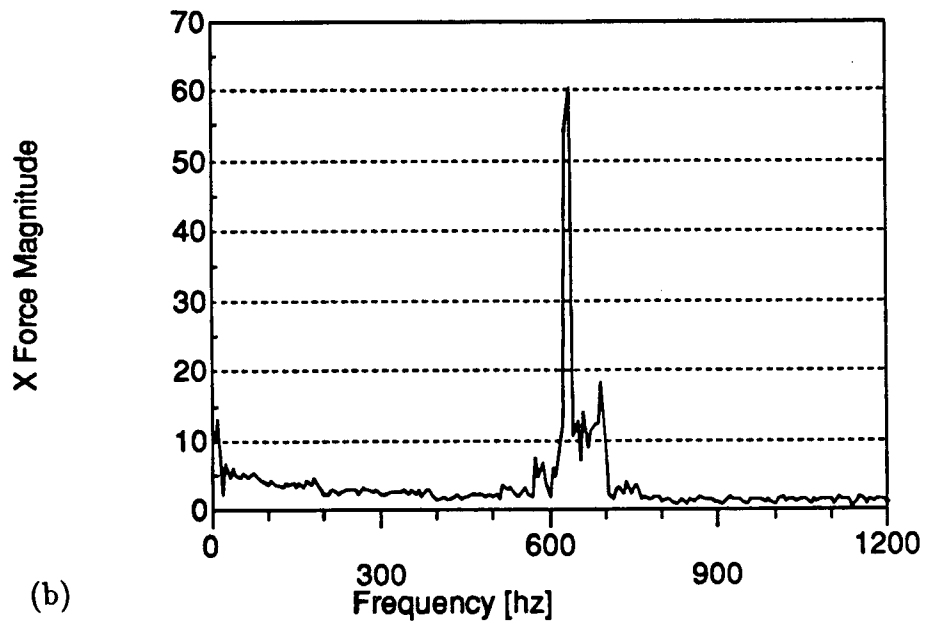


Figure 5.2: Experimental (X) cutter deflection for three depths of cut. $n = 1200$ [rpm], $\phi = 180^0$, $a =$ (a) 1.27, (b) 1.90, (c) 2.54 [mm].



(a)



(b)

Figure 5.3: Experimental (X) cutting force spectrum of stable and unstable cut. n = 900 [rpm], $\phi = 180^0$, a = (a) 1.27, (b) 1.40 [mm].

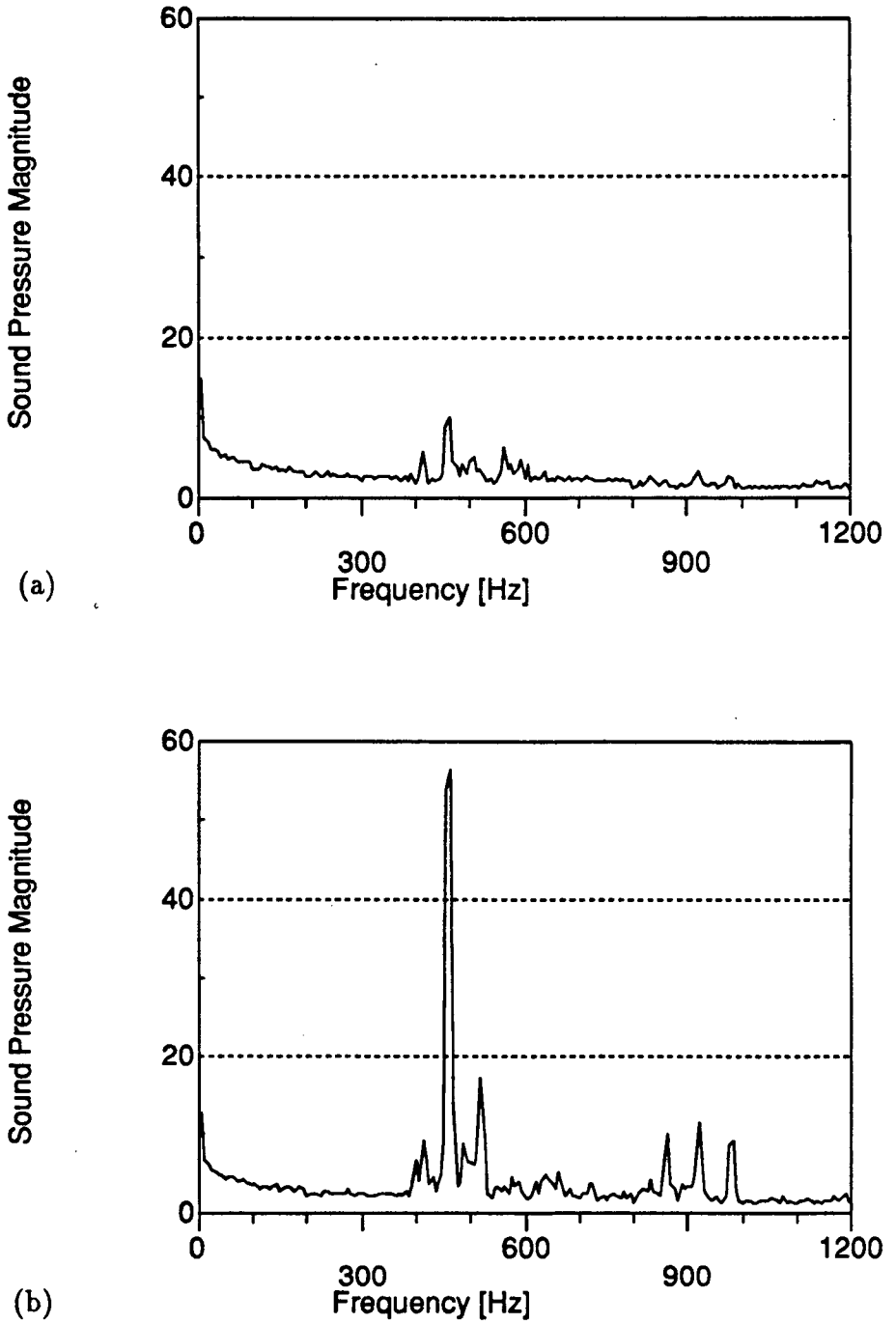


Figure 5.4: Experimental sound pressure spectrum of stable and unstable cut. $n = 900$ [rpm], $\phi = 180^0$, $a =$ (a) 0.76, (b) 1.02 [mm].

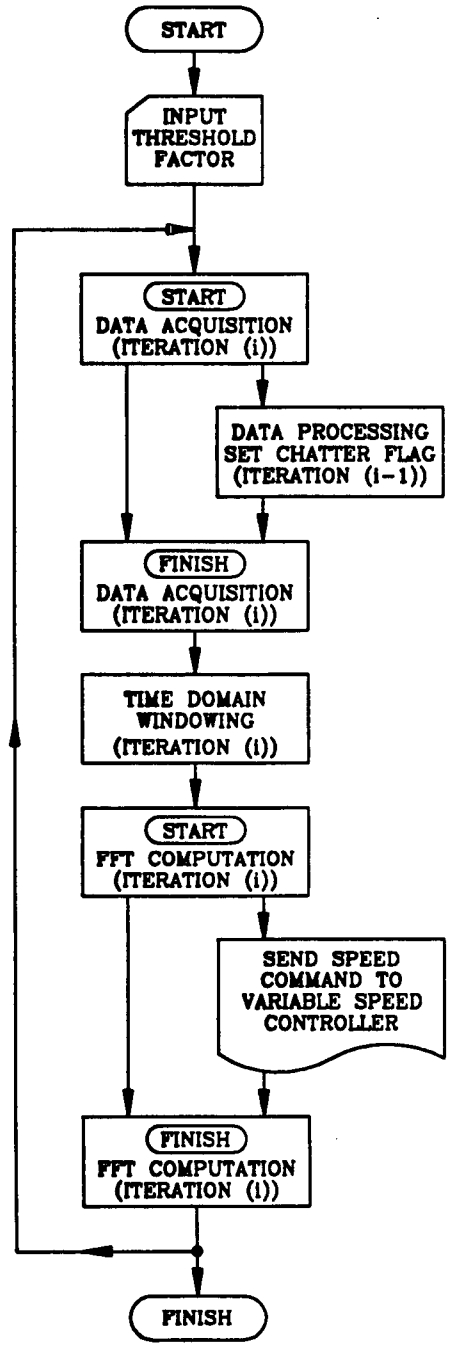
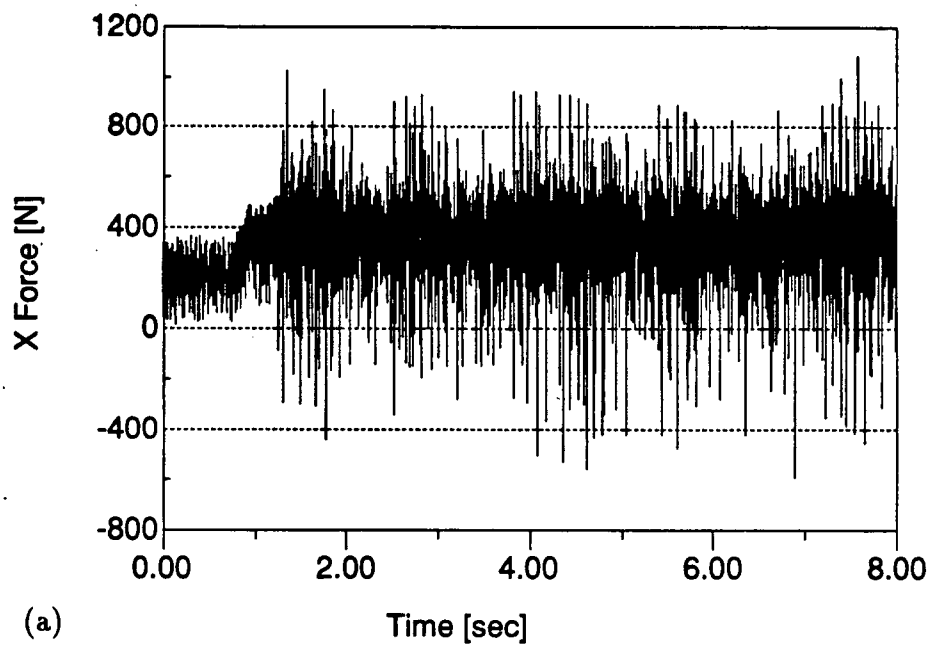


Figure 5.5: Chatter avoidance algorithm.



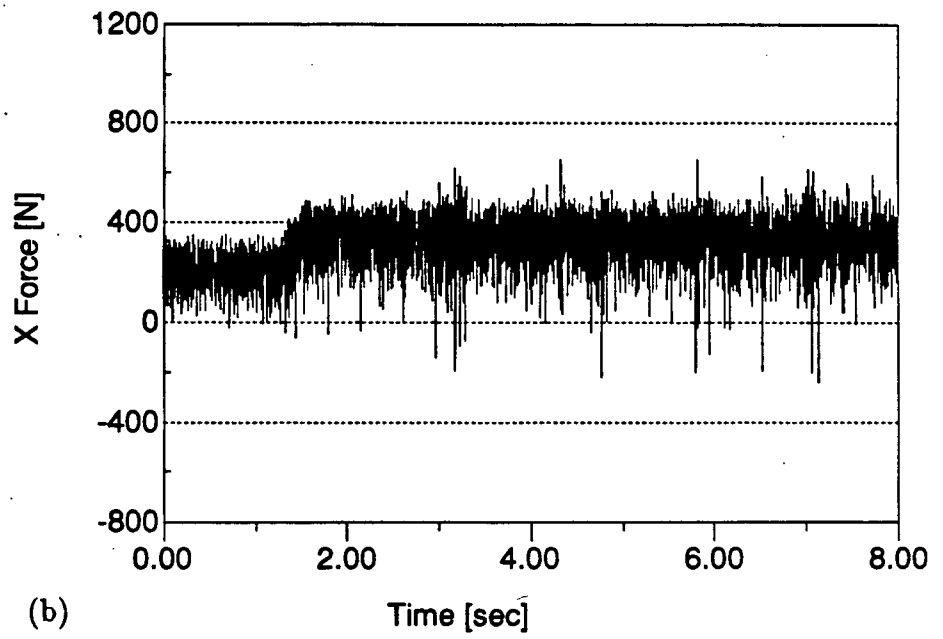
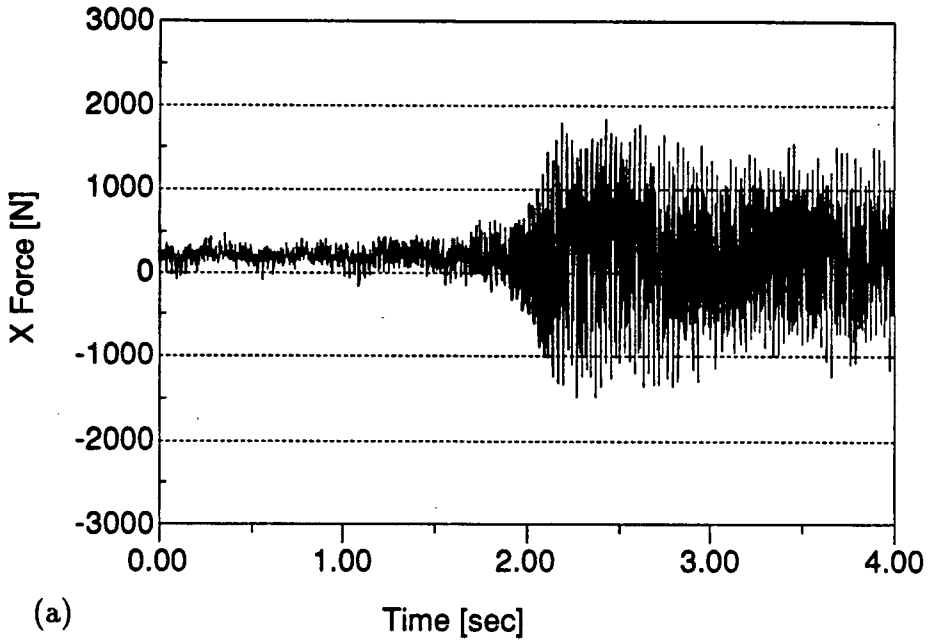


Figure 5.6: Comparison of (X) cutting force between fixed speed and chatter avoidance system. $n = 1200$ [rpm], $\phi = 180^0$, $a = 1.27$ to 2.54 [mm].



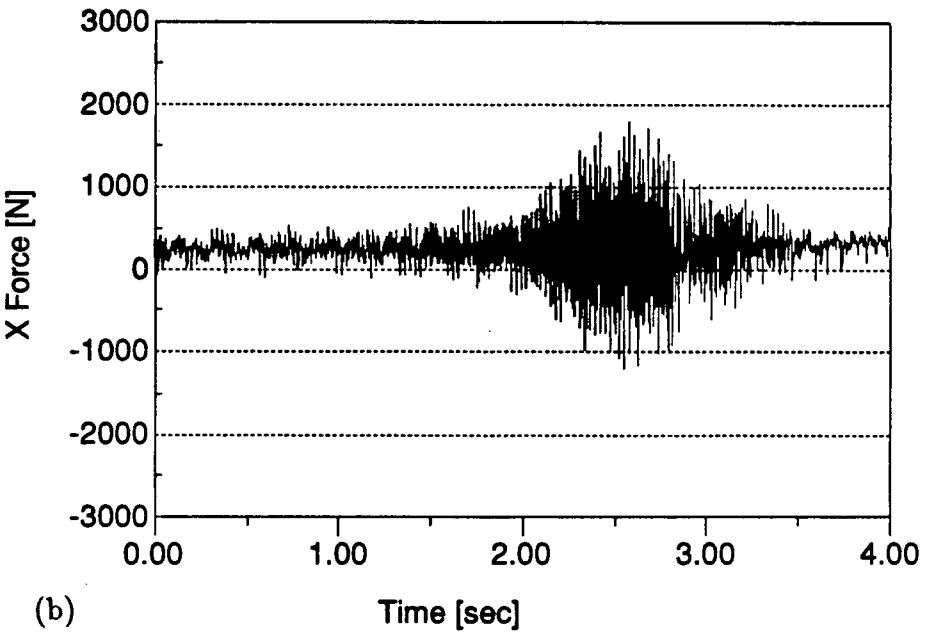


Figure 5.7: Chatter detection and avoidance in face milling. n = 600 [rpm], $\phi = 112^0$, a = 1.27 to 2.54 [mm].

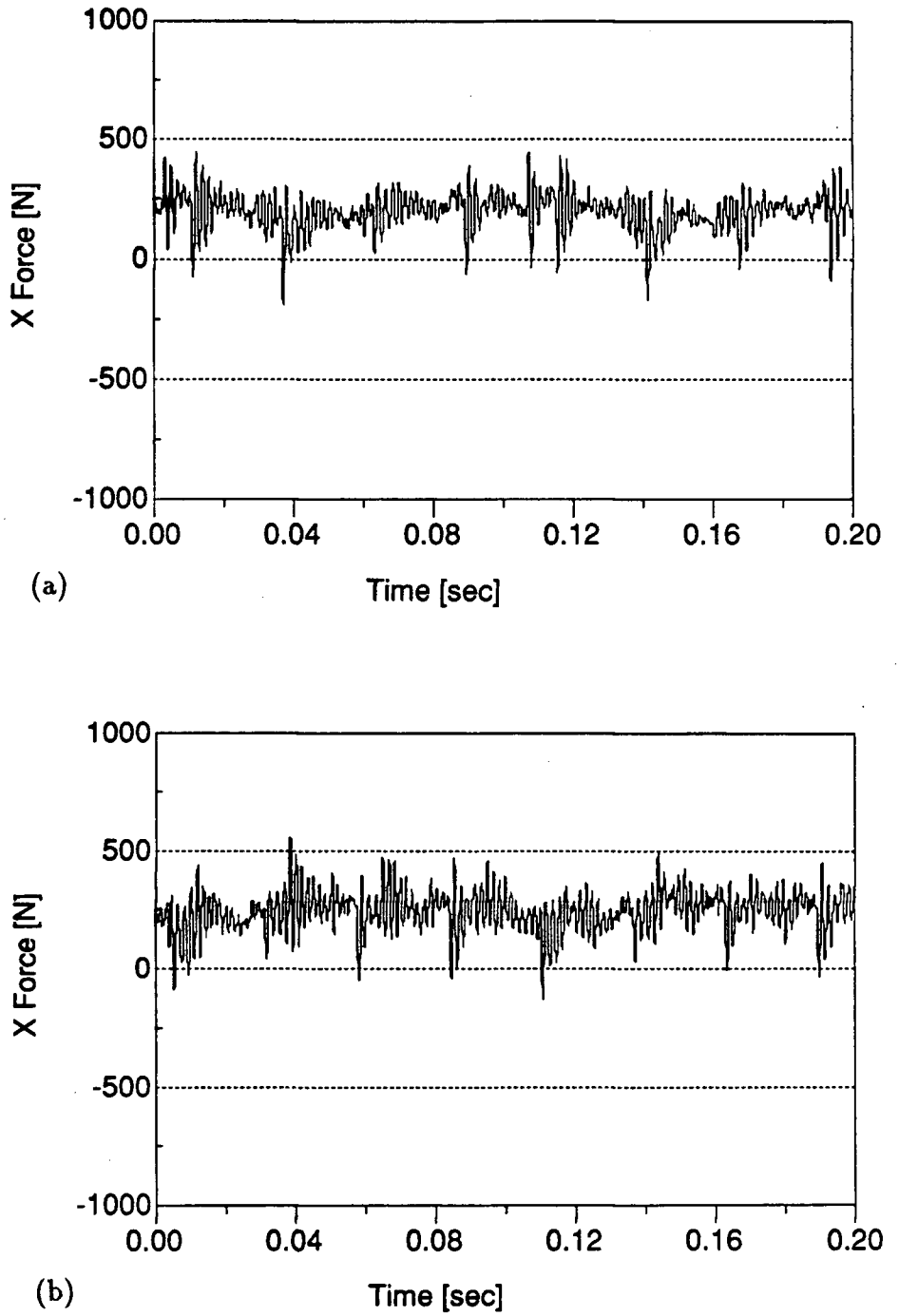
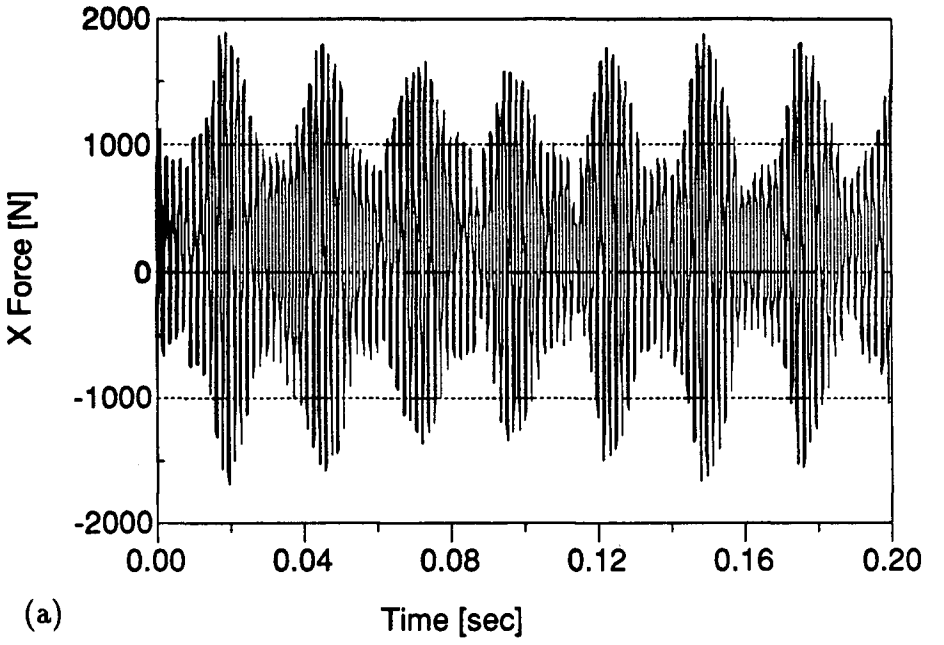


Figure 5.8: Close-up comparison of X) cutting force before chatter. $n = 600$ [rpm], $\phi = 112^0$, $a = 2.54$ [mm].



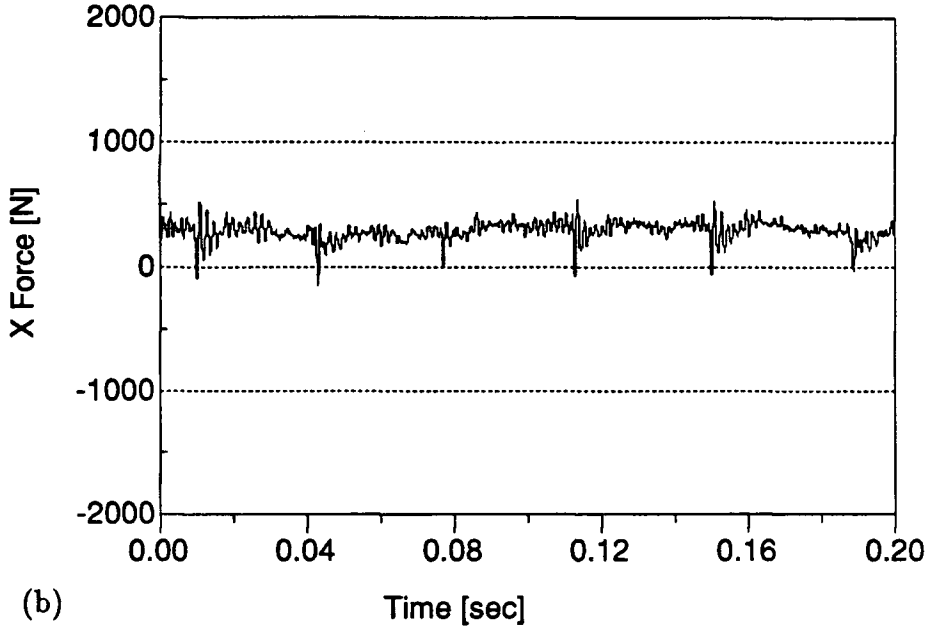


Figure 5.9: Close-up comparison of (X) cutting force after chatter. $n = 600$ [rpm], $\phi = 112^0$, $a = 2.54$ [mm].

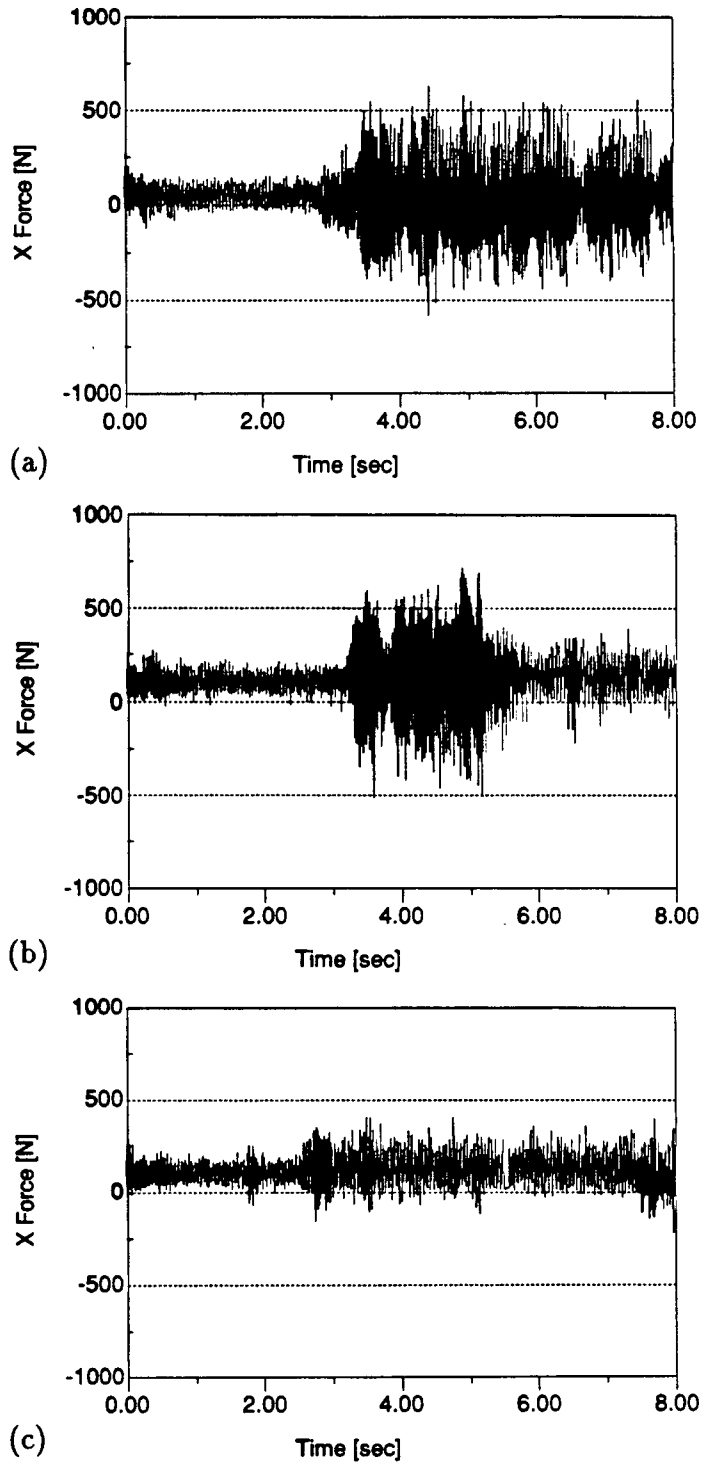


Figure 5.10: Comparison of responses from different triggering time. n = 1200 [rpm], ϕ = 180^0, a = 0.64 to 1.27 [mm].

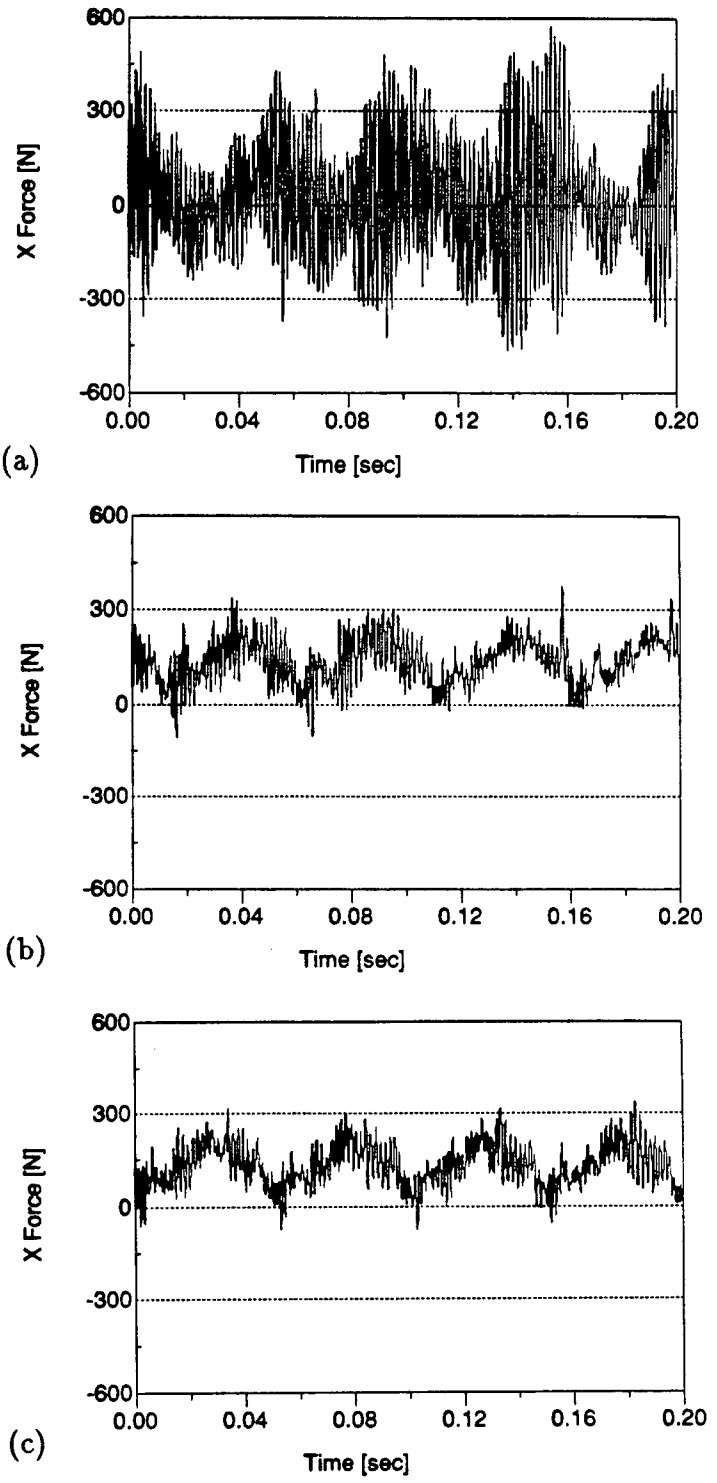
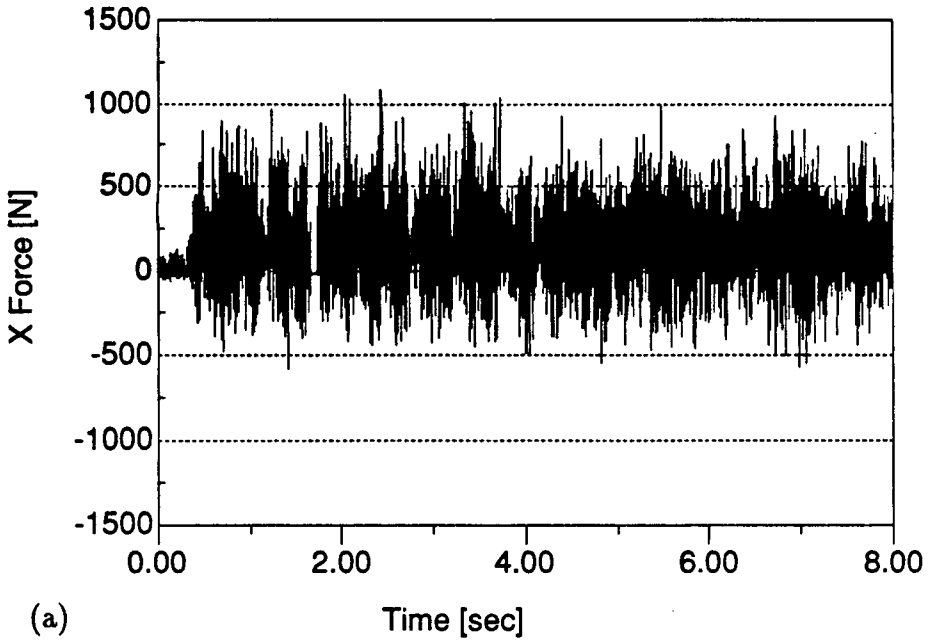
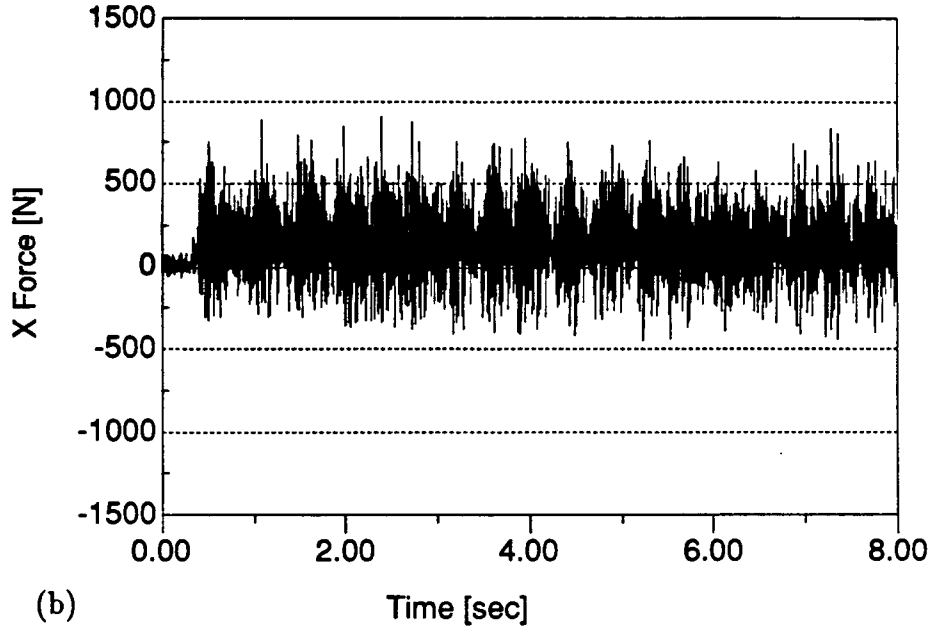


Figure 5.11: Close-up comparison of (X) cutting force after chatter. n = 1200 [rpm], ϕ = 180^0, a = 0.64 to 1.27 [mm].

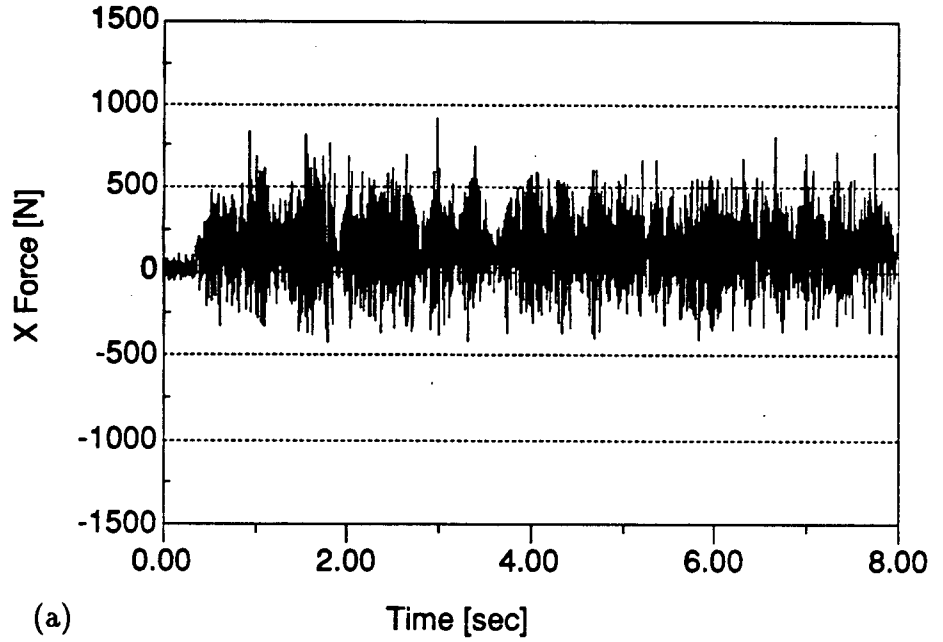


(a)

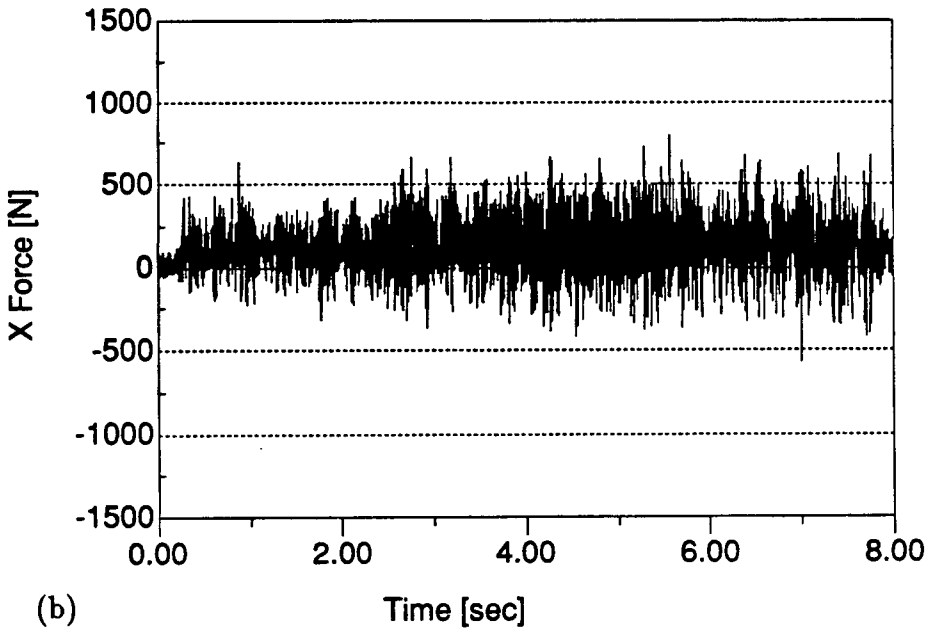


(b)

Figure 5.12: Effect of speed variation magnitude on chatter suppression. $n = 900$ [rpm], $\phi = 120^0$, $a = 0.64$ to 1.27 [mm].



(a)



(b)

Figure 5.12: Effect of speed variation magnitude on chatter suppression. $n = 900$ [rpm], $\phi = 120^0$, $a = 0.64$ to 1.27 [mm].

Chapter 6

Concluding Remarks

6.1 Conclusions

The use of a continuously variable spindle speed as a means of suppressing chatter in milling has been investigated. The method was implemented as part of an automatic chatter detection and avoidance system.

Digital turning simulations and frequency domain analysis determined that speed oscillations stabilize cutting due to two reasons. The first is the prevention of the most favorable (and therefore most unstable) wave regeneration mechanism from developing. The most unstable wave regeneration process for a certain spindle speed and depth of cut has a fixed phase shift and chatter frequency associated with it. Speed oscillations cause the phase shift between the inner and outer modulation to fluctuate continuously, not settling down to a fixed value. The second reason is the shift of the cutting process to spindle speeds that are inherently more stable than the nominal spindle speed.

A digital milling simulation gave more insight into the effectiveness of variable speed milling. The simulation included various features that could influence cutting stability, such as runout, mode coupling, flank interference, tool wear, and amplitude and frequency of speed variation. Simulation results showed the possible stability improvement with speed oscillations. Variable speed milling is most successful at increasing the chatter limit when there are large speed variations and minimal mode coupling activity.

Cutting tests were performed to verify the findings from digital simulation. Most

trends introduced from simulations were witnessed in actual cutting tests. Experimental data showed that variable cutting speed is indeed capable of suppressing chatter. However, there were factors that were detrimental to its stabilizing effect, such as runout, excessive flank interference, and low variation amplitude and frequency limits.

Chatter, however, can never be eliminated unless the phase shift between the present and previous vibrations is fixed at zero. Zero phase shift occurs if the chatter frequency is equal to the natural frequency. The chatter frequency approaches the natural frequency only if the cutting frequency is equal to or much greater than the natural frequency (as suggested by Tlusty and Smith [14]).

Automatic chatter detection and avoidance was implemented on a vertical knee type milling machine. The complete algorithm was controlled by a personal computer with a resident digital signal processing board. The chatter signal came from a three axis dynamometer, a proximity sensor, or an audio microphone. Experiments performed with the microphone produced some successful results and important observations. Large speed variations and early detection were both beneficial to quick chatter suppression. The experimental set-up, however, had limitations and as a result the full potential of variable speed cutting was not achieved.

6.2 Future Work

Development of this research lies in two areas: 1) modelling of dynamic cutting, and 2) experimental testing.

The accuracy of the dynamic cutting model used in this thesis needs improvement. The discrepancy between simulation and experimental results is primarily caused by deficient modelling of ploughing forces. The concern and necessity to develop a reliable process damping model is by no means restricted to the area of variable spindle speed

milling. The problem is one of the most important issues in dynamic cutting research. Anyhow, a better cutting force prediction model may improve the variable speed milling simulation so that it predicts stability characteristics instead of just shows trends as it is doing now.

Improvement in force prediction should be concurrent with extensive experimental testing. The present research equipment lacks only a high power variable speed spindle for more conclusive experimental verification. Testing could be much better controlled with a high power spindle.

Bibliography

[1] Armarego, E. J. A. and Brown, R. H., *The Machining of Metals* , Prentice-Hall Inc., 1969.

[2] Fu, H. J., A Dynamic Modelling Approach to the Optimal Design of Nonuniform Chip Loading in Face Milling, *Ph.D. Thesis, University of Illinois*, 1985.

[3] Halling, J. (ed.), *Principles of Tribology*, The Macmillan Press Ltd., pp.72–91, 1983.

[4] Ismail, F., Identification, Modelling, and Modification of Mechanical Structures From Modal Analysis Testing, *Ph.D. Thesis, McMaster University*, 1982.

[5] Jemielniak, K. and Widota, A., Suppression of Self-Excited Vibration by the Spindle Speed Variation Method, *Int. J. Mach. Tool Des. Res.*, Vol. 23, No. 3, pp. 207–214, 1984.

[6] King, Robert I.(ed.), *Handbood of High Speed Machining Technology*, Chapman and Hall, pp.48–153, 1985.

[7] Lee, E. H. and Sheffer, W. B., *J. Appl. Math* 18, 1951.

[8] Lin, S. C., DeVor, R. E. and Kapoor, S.G., The Effects of Variable Speed Cutting on Vibration Control in Face Milling, *Proc. ASME 1988 Winter Annual Meeting*, Book PED-Vol. 33, pp. 41–51, 1988.

[9] Merchant, M.E., Mechanics of the Cutting Process, *J. Appl. Phys.* 16, 1945.

[10] Montgomery, D. T. and Altintas, Y., Mechanism of Cutting Force and Surface Generation in Dynamic Milling, *Proc. ASME 1990 Winter Annual Meeting*, Book PED-Vol. 40, pp. 65–74, 1990.

[11] Palmer, W. B. and Oxley, P. L. B., *Proc. Instn. Mech. Engrs.* 173, 1959.

[12] Rahman, M., In-Process Detection of Chatter Threshold, *ASME, J. of Eng. Ind.*, Vol. 110, pp. 44–50, 1988.

[13] Sisson, T. R., Kegg, R. L., An Explanation of Low Speed Chatter Effects, *ASME, J. of Eng. Ind.*, Vol. 91, Nov. 1969, pp. 951–955.

[14] Smith, S. and Tlusty, J., Update on High-Speed Milling Dynamics, *ASME, J. of Eng. Ind.*, Vol. 112, May 1990, pp. 142–149.

[15] Smith, S., Chatter, Forced Vibrations, and Accuracy in High-Speed Milling, *Master's Thesis, University of Florida*, 1986.

[16] Smith, S. and Delio, T., Sensor-Based Control for Chatter-Free Milling by Spindle Speed Selection, *Proc. ASME 1989 WAM*, Book DSC-Vol. 18, pp. 107–114.

[17] Takemura, T. et al., Active Suppression of Chatter by Programmed Variation of Spindle Speed, *Annals of the CIRP*, Vol. 23/1/1974, pp. 121–122, 1974.

[18] Tlusty, J., Analysis of the State of Research in Cutting Dynamics, *Annals of CIRP*, Vol.27/2/1978, pp. 583–589, 1978.

[19] Tobias, S. A., *Machine-Tool Vibration*, Blackie and Son Ltd. London, pp. 143–179, 1965.

[20] Vanherck, P., Increasing Milling Machine Productivity by Use of Cutter with Non-Constant Cutting-Edge Pitch, *Proc. Adv. MTDR Conf.*, No. 8, pp. 947–960, 1967.

[21] Vold, Harvard, Insight, Not Numbers, *Proceedings of the 7th International Modal Analysis Conference*, Vol. 1, pp. 16–19, 1989.

[22] Wu, D. W., A New Approach of Formulating the Transfer Function for Dynamic Cutting Processes, *ASME, J. of Eng. Ind.*, Vol. 111, Feb. 1989, pp. 37–47.

Appendix A

Listing of Programs

- Simulation of variable speed plunge turning.
- Simulation of variable speed milling.
- On line chatter detection and avoidance.

The code for the above programs can be found in the following pages.

```
/*                          DIGITAL SIMULATION OF CHATTER AVOIDANCE          */
/*                          WITH ONLINE PHASE SHIFT APPROXIMATION            */
/*                          FOR TWO DEGREE OF FREEDOM                        */
/*                                 (DSOC2.C)                                 */
/*                                                                           */
/*                          AUTHOR : PHILIP CHAN                             */
/*                          COMPILER: APOLLO VERSION                         */
/*                          FOR : THE SHEAR JOY OF IT                        */
/*                          UPDATED : JUNE 11, 1990                          */

#include <math.h>
#include <stdio.h>

#define     Ks      2000000000.0
#define     PI       3.1415927

FILE *ff1, *ff2, *ff3;
char fname[20];

int  count, n, N, pvf, nrev, vcs, pc, fileop, npi;
float  skippy, skip;
float ndt, dt, t, h, f, f1, f2,  m1, m2, c1, c2, k1, k2, beta, alpha1, alpha2, b;
float x1, x1d, x1dd, x1_1, x1d_1, x2, x2d, x2dd, x2_1, x2d_1;
float y[88888], c, feedper, rpm, drpm, frpm, speed, rps, ymax1, ymax2;
float rpmmin, rpmmax, rpmstep, drpmmin, drpmmax, ddrpm, frpmmin, frpmmax, dfrpm;

float fn1, wn1, eta1, fn2, wn2, eta2, bmin, bmax, db, yfax1, yfax2, ystatic;
int   ws1, we1, ws2, we2;

float peak, valley, phase[88888], freq, epsilon, nrevr1, npir1;
int eps, magn;
float tallyep[361], totalep, ev1, ev2, var1, var2, magev, magvar;

/*  Important variables :

    N       -number or iterations to simulate
    pvf     -if equal to 0, then searching for valley
            -if equal to 1, then searching for peak
    vcs     -revolution at which to start variable speed
    pc      -revolution to start phase shift calculations
            -must be at least 2
    fileop  -file option    : 1 -summary
                            : 2 -phase distribution
                            : 3 -time history

    skippy  -sampling frequency for time histroy plot
    ndt     -nominal time increment
    dt      -time increment
    t       -time
    h       -chip thickness
    f       -resultant cutting force
    b       -width of cut
    f1..k2  -orhtogonal mode parameters
    x1..x2dd-present modal deflections and derivatives
    x1_1... -previous modal deflections and derivatives
    y[i]    -normal deflection at increment i
    feedper -nominal feed per tooth
    c       -feed per tooth
    rpm     -nominal spindle speed in [rpm]
    drpm    -speed variation amplitude
    frpm    -speed variation frequency [hz]

    ev1     -expected value of phase shift
    var1    -variance of phase shift

    ymax1   -maximum y during first window
    yfax1   -maximum y divided by static y during first window

    Input file:

    Example                 Explanation
```

```
    tsm13.dat           output file name
    2                   file option 2 (phase distribution)
    500.0               plotting frequency of 500 hz
    0.0027              minimum b at 2.7 mm
    0.0037              maximum at 3.7 mm
    0.0005              increment b at 0.5 mm
    600.0               minimum speed at 600 rpm
    700.0               maximum speed at 700 rpm
    50.0                increment speed at 50 rpm
    50.0                minimum speed variation at amplitude 50 rpm
    60.0                maximum speed variation amplitude at 60 rpm
    10.0                increment speed variation amplitude at 10 rpm
    5.0                 min variation frequency
    6.0                 max variation frequency
    1.0                 variation freqeuncy increment
    30.0                angle of modal direction 1
    100.0               natural frequency of modal direction 1
    0.05                damping factor of modal direction 1
    40000000.0          stiffness of modal direction 1
    120.0                angle of modal direction 2
    120.0               natural frequency of modal direction 2
    0.05                damping factor of modal direction 2
    56000000.0          stiffness of modal direction 2
    80000               number of points to simulate
    4000                number of points per revolution
    12                  variable speed starts at revolution 12
    2                   phase calculations start at second revolution
    44000               window 1 start
    48000               window 1 end
    76000               window 2 start
    80000               wondow 2 end
*/

main ()
{

   file_spec();
   fprintf(ff1, "    rpm       drpm      frpm      ymax1      ymax2      yfax1
                                         yfax2        ave-eps     std-eps \n");

   for (b = bmin; b <= bmax; b+= db)
   {
      for (rpm = rpmmin; rpm <= rpmmax; rpm += rpmstep)
          {
         for (drpm = drpmmin; drpm <= drpmmax ; drpm += ddrpm)
             {
                for (frpm = frpmmin; frpm <= frpmmax ; frpm += dfrpm)
                    {
                      set_constants();
               while (n <= N)
               {
                  n++;
                  if (n < (vcs*nrev))
                  {
                     dt = ndt;
                     c = feedper;
                  }
                  else
                  {
                     speed = rpm + drpm*sin(2.0*PI*frpm*t);
                     rps = speed/60.0;
                     dt = 1/(rps*nrevrl);
                     c = feedper*rpm/speed;
                  }
                  t += dt;
                  y[n] = cos(alpha1) * x1 + cos(alpha2) * x2;
/* start of phase shift calculations */
                  if (n > nrev)
                  {
                     if ((pvf == 1) && (y[n] < y[n-1]))
                     {
                        peak = t;
```

```
                        pvf = 0;
                    }
                    if ((pvf == 0) && (y[n] > y[n-1]))
                    {
                        valley = t;
                        pvf = 1;
                    }
                    if ( peak != valley)
                        freq = 1.0/(2.0*fabs(peak - valley));
                    phase[n] = (t - peak)*freq*360.0;
                }
                if ( n > (pc*nrev))
                {
                    epsilon = phase[n] - phase[n-nrev];
                    npirl = epsilon/(360.0);
                    npi = (int)npirl;
                    epsilon -= ((float)npi*360.0);
                    if (epsilon < 0.0) epsilon +=360.0;
                    eps = epsilon;
                    tallyep[eps] += (dt/ndt);
                    totalep += (dt/ndt);
                }

/* end of phase thing */
                if ( n<= nrev)
                {
                    h = c - y[n];
                }
                else
                {
                    h = c + y[n-nrev] - y[n];
                }
                if ( h > 0.0)
                {
                    f = Ks * b * h ;
                }
                else
                {
                    f = 0.0;
                }
                f1 = f * cos(beta - alpha1);
                f2 = f * cos(beta - alpha2);
                x1dd = (f1 - c1 * x1d_1 - k1 * x1_1)/m1;
                x1d = x1d_1 + x1dd * dt;
                x1 = x1_1 + x1d * dt;
                x2dd = (f2 - c2 * x2d_1 - k2 * x2_1)/m2;
                x2d = x2d_1 + x2dd * dt;
                x2 = x2_1 + x2d * dt;
                if ((((fileop ==3) || (fileop == 4)) && (t >= skip))
                {
                    fprintf(ff1, "%f %f %f %f %f \n",t, 1000000.0 * y[n],
                         f, freq, epsilon);
                    skip += skippy;
                }
                if ((n > ws1) && (n <= we1) && (y[n] > ymax1)) ymax1 = y[n];
                if ((n > ws2) && (n <= we2) && (y[n] > ymax2)) ymax2 = y[n];
                x1_1 = x1;
                x1d_1 = x1d;
                x2_1 = x2;
                x2d_1 = x2d;
            }
            stats();
        }
    }
  }
}

stats()
{
    if (fileop == 2)
    {
```

```
      fprintf(ff3, "%f \n", rpm);
      fprintf(ff3, "%f \n", drpm);
      fprintf(ff3, "%f \n\n", frpm);
   }
   for (n = 0; n < 360; n++)
   {
      if ((fileop == 2) || (fileop == 4))
      {
         fprintf(ff3, " %d %f \n", n, tallyep[n]);
      }
          ev1 += tallyep[n]*(float)n/totalep;
          if (tallyep[n] > 0.0)
          {
         magev += tallyep[n];
                magn++;
          }
      if (n < 180)
      {
            ev2 = ev1;
      }
      else
      {
                ev2 += tallyep[n]*((float)n - 180.0)/totalep;
      }
   }
   magev /= (float)magn;
   for (n=0; n < 360; n++)
   {
      var1 += tallyep[n]*((float)n-ev1)*((float)n-ev1)/totalep;
          if (tallyep[n] > 0.0)
          {
             magvar += (tallyep[n] - magev)*(tallyep[n] - magev)/(float)magn;
          }
          if (n < 180)
          {
             var2 = var1;
          }
          else
          {
             var2 += tallyep[n]*((float)n-180.0-ev2)*((float)n-180.0-ev2)/totalep;
      }
   }
   if ((fileop == 2) || (fileop == 4))
   {
      fprintf(ff3, "\n%f \n", ev1);
      fprintf(ff3, "%f \n", sqrt(var1));
          fprintf(ff3, "%f \n", magev);
          fprintf(ff3, "%f \n \n ",sqrt(magvar));
      fprintf(ff3, "%f %f %f %f %f %f %f %f %f \n", rpm, drpm, frpm, ev1, sqrt(var1),
                                          ev2, sqrt(var2), magev, sqrt(magvar));
   }
   yfax1 = ymax1/ystatic;
   yfax2 = ymax2/ystatic;
   fprintf(ff1, "%5.2f %5.2f %5.2f %5.3f %5.3f %5.4f %5.4f %5.3f %5.3f \n", rpm,
      drpm, frpm, 1000000.0*ymax1, 1000000.0*ymax2, yfxx1, yfax2, ev1, sqrt(var1));
   printf("%5.2f %5.2f %5.2f %5.3f %5.3f %5.4f %5.4f %5.3f %5.3f \n", rpm, drpm,
      frpm, 1000000.0*ymax1, 1000000.0*ymax2, yfax1, yfax2, ev1, sqrt(var1));

}


set_constants()
{
   feedper = .00015;    /*  nominal feed per [m/rev] */

   beta = 70.0 / 180.0 * PI;

   wn1 = fn1*(2.0*PI);
   m1 = k1/(wn1*wn1);
   c1 = 2.0*eta1*m1*wn1;
   wn2 = fn2*(2.0*PI);
```

```
    m2 = k2/(wn2*wn2);
    c2 = 2.0*eta2*m2*wn2;
    ystatic = b*feedper*Ks*((cos(beta-alpha1)*cos(alpha1)/k1) +
                            (cos(beta-alpha2)*cos(alpha2)/k2));

    skip = 0;

    pvf = 1;
    epsilon = 180.0;
    freq  = 100.0;
    peak = valley = 0.0;

    nrevrl = nrev;
    ndt = 1.0/((rpm/60.0)*nrevrl);

    printf("%f \n", b);

    x1 = x1_1 = x1d_1 = 0.0;
    x2 = x2_1 = x2d_1 = 0.0;
    t = 0.0;
    ymax1 = ymax2 = 0.0;

    ev1 = ev2 = 0.0;
    var1 = var2 = 0.0;
    magev = magvar = 0.0;
    magn = 0;
    totalep = 0.0;
    for(n = 0; n <= 360; n++)
    {
       tallyep[n] = 0.0;
    }
    n = 0;

}


file_spec()
{
    ff2 = fopen("d2spec.dat", "r");
    printf(" Enter the complete summary or time output file name : \n");
    fgets(fname, 10, ff2);
    ff1 = fopen(fname, "w");
    ff3 = ff1;
    printf(" Enter the file option (1 = summary, 2 = phase, 3 = time, 4 = both : \n");
    fscanf(ff2, "%d", &fileop);

    printf(" Enter the number of plotting frequency  (~1000)  : \n");
    fscanf(ff2, "%f", &skippy);
    skippy = 1.0/skippy;
    printf(" Enter the minimum depth of cut (m) : \n");
    fscanf(ff2, "%f", &bmin);
    printf(" Enter the maximum depth of cut (m) : \n");
    fscanf(ff2, "%f", &bmax);
    printf(" Enter the depth of cut increment (m): \n");
    fscanf(ff2, "%f", &db);
    printf(" Enter the minimum nominal spindle speed (rpm) : \n");
    fscanf(ff2, "%f", &rpmmin);
    printf(" Enter the maximum nominal spindle speed (rpm) : \n");
    fscanf(ff2, "%f", &rpmmax);
    printf(" Enter the nominal spindle speed step(rpm) : \n");
    fscanf(ff2, "%f", &rpmstep);
    printf(" Enter the minimum spindle speed variation (rpm) : \n");
    fscanf(ff2, "%f", &drpmmin);
    printf(" Enter the maximum spindle speed variation (rpm) : \n");
    fscanf(ff2, "%f", &drpmmax);
    printf(" Enter the spindle speed variation step (rpm) : \n");
    fscanf(ff2, "%f", &ddrpm);
    printf(" Enter the minimum spindle speed freq(hz) : \n");
    fscanf(ff2, "%f", &frpmmin);
    printf(" Enter the maximum spindle speed freq(hz) : \n");
    fscanf(ff2, "%f", &frpmmax);
    printf(" Enter the spindle speed freq step (hz) : \n");
```

```
    fscanf(ff2, "%f", &dfrpm);

    printf(" Enter the alpha for mode 1 (~30.0) : \n");
    fscanf(ff2, "%f", &alpha1);
    alpha1 = alpha1 / 180.0 * PI;
    printf(" Enter the system1 frequency (hz) : \n");
    fscanf(ff2, "%f", &fn1);
    printf(" Enter the damping factor 1 (~0.05) : \n");
    fscanf(ff2, "%f", &eta1);
    printf(" Enter the stiffness for 1(n/m) : \n");
    fscanf(ff2, "%f", &k1);
    printf(" Enter the alpha for mode 2 (~120.0) : \n");
    fscanf(ff2, "%f", &alpha2);
    alpha2 = alpha2 / 180.0 * PI;
    printf(" Enter the system1 frequency (hz) : \n");
    fscanf(ff2, "%f", &fn2);
    printf(" Enter the damping factor 2 (~0.05) : \n");
    fscanf(ff2, "%f", &eta2);
    printf(" Enter the stiffness for 2 (n/m) : \n");
    fscanf(ff2, "%f", &k2);

    printf(" Enter the number of points (<= 20000) : \n");
    fscanf(ff2, "%d", &N);
    printf(" Enter the number of points per rev (~500-1000) : \n");
    fscanf(ff2, "%d", &nrev);
    printf(" Enter at what revolution does vcs kick in (~4) : \n");
    fscanf(ff2, "%d", &vcs);
    printf(" Enter when the phase calculations kick in (~3) : \n");
    fscanf(ff2, "%d", &pc);

    printf(" Enter start time for ymax1  (~4500) : \n");
    fscanf(ff2, "%d", &ws1);
    printf(" Enter end time for ymax1  (~5000) : \n");
    fscanf(ff2, "%d", &we1);
    printf(" Enter start time for ymax2  (~9500) : \n");
    fscanf(ff2, "%d", &ws2);
    printf(" Enter start time for ymax1  (~10000) : \n");
    fscanf(ff2, "%d", &we2);

}
```

```
c
c                    PROGRAM: STABILITY LOBES FROM TIME SIMULATION
c                             (SLFTS8.FTN)
c                    COMPILEE: PHILIP CHAN
c                    COMPILER: SOME APOLLO FORTRAN THING
c                    COMPILED: APRIL 14, 1990
c
c
c Declare variables
c
      real*8   feed, teeth, b, Ks, rl, phis, phie, dphi, tphi, helix,
     *         t, h, phi, phi1, phi2, dt, dft, dfr, dfy, dfx, fx, fy,
     *         x(3), xd(3), xdd(3), y(3), yd(3), ydd(3), zmin(1800,15),
     *         kx, cx, mx, etax, fnx, wnx, ky, cy, my, etay, fny, wny,
     *         rpm, drpm, frpm, speed, pi, c, fperth, ms(15),
     *         tlength, tincr, fs, tt1, tt2, ts1, te1, ts2, te2
c
      real*8   gamma, gammae, lpl, lplmax, lpar, lpen,
     *         dfplr, dfplt, po, vb, gammas,
     *         vf, vyz(2), vyzlim, dphi2, zminl(15),
     *         ptimes, ppercent, itotal, ro(12), tro, aro(12)
c
      logical  cross, plough, justpl, skip
c
      real*8   ymax, tymax, ylim, ylimf, ymaxf, ymxphi,
     *         xmax, txmax, xlim, xlimf, xmaxf, xmxphi,
     *         ppxmax, xmx, xmn, ppymax, ymx, ymn,
     *         blim, bmax, db, bstart, bf,
     *         rpmst, rpmmax, rpmskp
c
      real*8   alpha, xx, yy, fxp, fyp, fxmax, fymax, tfxmax, tfymax,
     *         xxd, yyd, kappa, tkappa, xc, dxc, dy, dymax, tdymax,
     *         xcl, dymn, dymx, ppdymax, surfphi
c
      integer  i, imax, j, k, p, r, rmax, revs, tth, fchoice, backup,
     *         m, m2, n, nmax, rev, tz, numofb, s, pcount,
     *         plf, tzmin, tzmax
      real*8   imaxrl, krl, prl, rrl, nobrl, tzrl,
     *         revrl, revsrl, plfrl, pcrl
c
      character*16 fname
c
c
c   Input file example
c
c       Example                 Explanation
cc
c     vstbf224.dat          summary file name
c     vstbs224.dat          surface file name
c     vstbd224.dat          complete time history file name (not used)
c     vstbp224.dat          time history plot file name
c     vstbt224.dat          miscellaneous file name (not used)
c     1                     file option 1: summary (2: + surface, 3: + plot)
c     2560.0                plot file sampling frequency
c     0.4                   plot file data window 1 start time
c     0.808                 plot file data window 1 end time
c     0.4                   plot file data window 2 start time
c     0.808                 plot file data window 2 end time
c     0.000781              minimum axial depth of cut (m)
c     0.001000              maximum axial depth of cut (m)
c     0.00009               (not used)
c     1.25                  depth of cut increment multiplication factor
c     11                    (not used)
c     0.00005               feed per tooth
c     4.0                   number of teeth
c     0.0                   runout of tooth 1
c     0.0000                runout of tooth 2
c     0.000                 runout of tooth 3
c     0.0000                runout of tooth 4
c     0.44                  cutting force ratio
c     1                     number of axial slices
c     1.00                  mode shape coefficient for slice 1 (not used)
```

```
c     700000000.0          specific cutting force coefficient
c     15.0                 (not used)
c     0.808                total simulation time
c     0.0                  start immersion angle
c     90.0                 end immersion angle
c     0.00004              nominal time increment
c     0.0                  helix angle
c     10.0                 clearance angle
c     0.0007               flank face length
c     0.00002              flank wear flat length
c     500000000.0          material yiel stress
c     50                   decrease in time increment for ploughing
c     0.0254               diameter of cutter
c     -15.0                rotation of orthogonal modes
c     2200000.0            stiffness in X ortho mode
c     2100000.0            stiffness in Y ortho mode
c     0.04                 damping in X ortho mode
c     0.04                 damping in Y ortho mode
c     800.0                natural frequency of X orhto mode
c     760.0                natural frequency of Y orhto mode
c     300.0                minimum spindle speed
c     1500.0               maximum spindle speed
c     50.0                 spindle speed increment
c     50.0                 speed variation amplitude
c     3.0                  speed variation frequency
c
c
c
c Get file names for input and output
c
      open (unit=4, file='simspec8', status = 'unknown')
      print *, ' Enter the output file name for full stability data:  '
      read (4, 01) fname
      open (unit=2, file=fname, status='unknown')
      print *, ' Enter the output file name for stability lobe data:  '
      read (4, 01) fname
      open (unit=3, file=fname, status='unknown')
      print *, ' Enter the output file name for digital time  data:  '
      read (4, 01) fname
      open (unit=7, file=fname, status='unknown')
      print *, ' Enter the output file name for plot data:  '
      read (4, 01) fname
      open (unit=8, file=fname, status='unknown')
      print *, ' Enter the output file name for test data:  '
      read (4, 01) fname
      open (unit=9, file=fname, status='unknown')
01    format (a16)
      print *,' Enter the file options (0 = f; 1 = f & s; 2 = f, s, & p;
     *; 3 = f, s, p, & d) :  '
      read (4, *) fchoice
c
      pi = 3.1415926
c
c Get necessary inputs
c
c
      print *, ' Enter the plot file sampling freq (Hz) : '
      read (4, *) fs
      print *, ' Enter the plot/digital data window 1 start time : '
      read (4, *) ts1
      print *, ' Enter the plot/digital data window 1 end time : '
      read (4, *) te1
      print *, ' Enter the plot/digital data window 2 start time : '
      read (4, *) ts2
      print *, ' Enter the plot/digital data window 2 start time : '
      read (4, *) te2
      print *, ' Enter the minimum axial depth of cut (m) : '
      read (4, *) bstart
      print *, ' Enter the maximum axial depth of cut (m) :  '
      read (4, *) bmax
      print *, ' Enter the critical axial depth of cut (m) :  '
      read (4, *) blim
```

```
      print *, ' Enter axial depth of cut multiplication factor
     *(approx. 1.15):  '
      read (4, *) bf
      print *, ' Enter the number of  adc steps to backup (~2) : '
      read (4, *) backup
      print *, ' Enter the feed per tooth : '
      read (4, *) fperth
      print *, ' Enter the number of teeth :  '
      read (4, *) teeth
      tth = teeth
      print *, ' Enter the measured runout of each tooth (m) : '
      tro = 0.0
      do 777 s = 1, tth
         read (4, *) ro(s)
         tro = tro + ro(s)
777   continue
      print *, ' Enter the cutting force ratio , rl :  '
      read (4, *) rl
      print *, ' Enter the number of slices per axial depth of
     *cut (approx. 10):  '
      read (4, *) numofb
      print *, ' Enter the mode shape coefficients (from bottom) :'
      do 111 s = 1, numofb
         read (4, *) ms(s)
111   continue
      print *, ' Enter the specific cutting pressure (N/m^2):  '
      read (4, *) Ks
      print *, ' Enter the chatter factor (~2.0):  '
      read (4, *) ylimf
      print *, ' Enter the the length of time to simulate (s) : '
      read (4, *) tlength
      print *, ' Enter the entry immersion angle (degrees) :  '
      read (4, *) phis
      print *, ' Enter the exit immersion angle (degrees) :  '
      read (4, *) phie
      print *, ' Enter the time increment (~0.000050):  '
      read (4, *) tincr
      print *, ' Enter the helix angle (degrees) (~30.0):  '
      read (4, *) helix
      print *, ' Enter the clearance angle (degrees) (~4.0):  '
      read (4, *) gamma
      print *, ' Enter the maximum ploughing length (~0.001) : '
      read (4, *) lplmax
      print *, ' Enter the wearland length, VB (~0.0 - 0.0005) : '
      read (4, *) vb
      print *, ' Enter the material yield strength : '
      read (4, *) po
      print *, ' Enter the time step factor for ploughing (~20) : '
      read (4, *) plf
      print *, ' Enter the diameter of the cutter (m):  '
      read (4, *) d
      print *, ' Enter the orthogonal mode direction (~ -15):  '
      read (4, *) alpha
      print *, ' Enter the X stiffness (N/m) :  '
      read (4, *) kx
      print *, ' Enter the Y stiffness (N/m) :  '
      read (4, *) ky
      print *, ' Enter the X damping ratio (~0.05) :  '
      read (4, *) etax
      print *, ' Enter the Y damping ratio (~0.05) :  '
      read (4, *) etay
      print *, ' Enter the X natural frequency (Hz) :  '
      read (4, *) fnx
      print *, ' Enter the Y natural frequency (Hz) :  '
      read (4, *) fny
      print *, ' Enter the starting spindle speed (rpm) :  '
      read (4, *) rpmst
      print *, ' Enter the maximum spindle speed (rpm) :  '
      read (4, *) rpmmax
      print *, ' Enter the spindle speed increment (rpm) :  '
      read (4, *) rpmskp
      print *, ' Enter amplitude of spindle speed variation (rpm) :  '
```

```
      read (4, *) drpm
      print *, ' Enter frequency of spindle speed variation (Hz) : '
      read (4, *) frpm
c
c Write necessary inputs to output file
c
      write (2, 13) fchoice
c
      write (2, 121) fs
      write (2, 127) ts1
      write (2, 124) te1
      write (2, 125) ts2
      write (2, 126) te2
c
      write (2, 12) bstart
      write (2, 23) bmax
      write (2, 34) blim
      write (2, 45) bf
      write (2, 455) backup
      write (2, 456) fperth
      write (2, 457) teeth
      do 177 s = 1, tth
         aro(s) = ro(s) - tro/tth
         write (2, 4578) s, ro(s), aro(s)
177   continue
      write (2, 459) rl
      write (2, 56) numofb
      do 122 s = 1, numofb
         write (2, 567) s, ms(s)
122   continue
      write (2, 67) Ks
      write (2, 78) ylimf
      write (2, 89) tlength
      write (2, 101) phis
      write (2, 112) phie
      write (2, 123) tincr
      write (2, 1234) helix
      write (2, 12345) gamma
      write (2, 12346) lplmax
      write (2, 12349) vb
      write (2, 12347) po
      write (2, 12348) plf
      write (2, 1235) d
      write (2, 12356) alpha
      write (2, 1236) kx
      write (2, 1237) ky
      write (2, 1238) etax
      write (2, 1239) etay
      write (2, 1240) fnx
      write (2, 1241) fny
      write (2, 134) rpmst
      write (2, 145) rpmmax
      write (2, 156) rpmskp
      write (2, 167) drpm
      write (2, 178) frpm
c
13    format (' file option = ', i2)
c
121   format (' plot file sampling frequency = ', f10.4)
127   format (' window 1 start time = ', f12.6)
124   format (' window 1 end  time = ', f12.6)
125   format (' window 2 start time = ', f12.6)
126   format (' window 2 end time = ', f12.6)
c
12    format (' starting b = ', f12.6)
23    format (' maximum b = ', f12.6)
34    format (' critical b = ', f12.6)
45    format (' bf = ', f12.6)
455   format (' adc steps to backup = ', i2)
456   format (' feed per tooth = ', f12.6)
457   format (' teeth = ', f12.6)
4578  format (' tooth#', i2, '-measured runout = ', f10.6,
```

```
     *        '  normalized runout = ', f10.6)
459   format (' r1 = ', f12.6)
56    format (' number of slices of b = ', i2)
567   format (' mode shape coefficient ', i2, ' = ', f12.6)
67    format (' specific cutting pressure = ', f17.6)
78    format (' chatter limit factor = ', f12.6)
89    format (' total simulation time = ', f12.6)
101   format (' entry angle = ', f12.6)
112   format (' exit angle = ', f12.6)
123   format (' time increment = ', f12.6)
1234  format (' helix angle = ', f12.6)
12345 format (' clearance angle = ', f12.6)
12346 format (' max plough edge length = ', f12.6)
12349 format (' wearland length  = ', f12.6)
12347 format (' material yield strength = ', f20.2)
12348 format (' ploughing time factor = ', i3)
1235  format (' cutter diameter = ', f12.6)
12356 format (' angle alpha = ', f16.6)
1236  format (' X stiffness = ', f16.6)
1237  format (' Y stiffness = ', f16.6)
1238  format (' X damping constant = ', f12.6)
1239  format (' Y damping constant = ', f12.6)
1240  format (' X natural frequency = ', f12.6)
1241  format (' Y natural frequency = ', f12.6)
134   format (' starting spindle speed = ', f12.6)
145   format (' maximum spindle speed = ', f12.6)
156   format (' spindle speed increment = ', f12.6)
167   format (' variation amplitude = ', f12.6)
178   format (' variation frequency = ', f12.6)
c
      write (2, *) '   rpm     b    ppdymax   dymax   tfxmax    fxmax  p
     *pxmax    xmax   xmxlim   tfymax    fymax  ppymax    ymax   ymxlim
     *  pprcnt'
c
c
c Set global constants
c
      phis = phis * (pi/180.0)
      phie = phie * (pi/180.0)
      helix = helix * (pi/180.0)
      gamma = gamma * (pi/180.0)
      alpha = alpha * (pi/180.0)
c
      tphi = 2.0*pi/teeth
      kappa = asin(fperth/d)
      tkappa = tphi - kappa
c
      wnx = 2*pi*fnx
      wny = 2*pi*fny
      mx = kx/(wnx*wnx)
      my = ky/(wny*wny)
      cx = 2*etax*mx*wnx
      cy = 2*etay*my*wny
c
      nmax = log(bmax/bstart)/log(bf)+1
      b = bstart*(bf**(backup + 1))
      n = backup + 1
c
c ***************************************
c
c Loop for incrementing spindle speed
c
c ***************************************
      rmax = (rpmmax - rpmst)/rpmskp + 1
      do 11 r = 1, rmax
         rrl = r
c
c Set constants of for each spindle speed
c
         rpm = rpmst + rpmskp*(rrl - 1)
         cross = .false.
         b = b/(bf**(backup + 1))
```

```
            n = n - (backup + 1)
            if (n .lt. 1) then
               n = 1
               b = bstart
            end if
c
            revs = tlength*(rpm /60.0)
            dphi = tincr*(2*pi*rpm/60.0)
c
            if (phis .le. 0.0) then
               surfphi = 0.0
            else
               surfphi = pi - dphi
            end if
c
            revsrl = revs
            imax = 2.0*pi/dphi
c
            feed = fperth*(teeth*rpm/60.0)
c
c ****************************************
c
c Loop for incrementing axial depth of cut
c
c ****************************************
            do while (n .le. nmax)
c
c Set constants of axial depth of cut
c
               xmax = 0.0
               txmax = 0.0
               ymax = 0.0
               tymax = 0.0
c
               dymn = 1.0
               dymx = 0.0
               ppdymax = 0.0
               xmn = 1.0
               xmx = -1.0
               ppxmax = 0.0
               ymn = 1.0
               ymx = -1.0
               ppymax = 0.0

               dymax = 0.0
               tdymax = 0.0
               fxmax = 0.0
               tfxmax = 0.0
               fymax = 0.0
               tfymax = 0.0
               nobrl = numofb
               db = b/nobrl
c
               ymxphi = (0.5*(atan(rl)) + (pi/2.0))
               if (phie .lt. ymxphi) then
                  ymaxf = rl*sin(phie)*cos(phie) - (sin(phie))**2
               else
                  ymaxf = rl*sin(ymxphi)*cos(ymxphi) - (sin(ymxphi))**2
               end if
               ylim = ylimf*(-ymaxf)*(Ks*fperth/ky) * b
               xlimf = ylimf
               xmxphi = (0.5*(atan(rl)) + (pi/4.0))
               if (phie .lt. xmxphi) then
                  xmaxf = sin(phie)*cos(phie) + rl*(sin(phie))**2
               else
                  xmaxf = sin(xmxphi)*cos(xmxphi) + rl*(sin(xmxphi))**2
               end if
               xlim = xlimf*xmaxf*(Ks*fperth/kx) * b
               print 189,  rpm, b
189            format (' rpm = ', f12.6, ' b =', f12.6)
c
c Initialize variables for each cut
```

```
c
            xc = 0.0
c
            xx = 0.0
            xxd = 0.0
            yy = 0.0
            yyd = 0.0
c
            x(1) = 0.0
            x(2) = 0.0
            xd(2) = 0.0
            xdd(2) = 0.0
            y(1) = 0.0
            y(2) = 0.0
            yd(2) = 0.0
            ydd(2) = 0.0
            t = 0.0
            tt1 = ts1
            tt2 = ts2
            dt = 0.0
            dft = 0.0
            dfr = 0.0
            dfy = 0.0
            dfx = 0.0
            do 33 m = 1, 1800
               do 133 m2 = 1, 15
                  zmin(m, m2) = 0.0
133            continue
33          continue
c
            dphi2 = dphi
            phi = phis - dphi2
c
            ptimes = 0.0
            plough = .false.
            skip = .false.
            justpl = .false.
c
            p = 1
c
c
c ************************************
c
c Loop for each revolution of spindle
c
c ************************************
c
            do while (p .le. revs)
               prl = p
c
               i = 1
c
c ************************************
c
c Loop for each time step
c
c ************************************
c
c
               do while (i .le. imax)
                  phi = phi + dphi2
c
                  fx = 0.0
                  fy = 0.0
c
c Determine spindle speed, feed/tooth, and cutting speed
c
                  speed = rpm + drpm*sin(2*pi*frpm*t)
                  c = feed/(teeth*speed/60.0)
                  vf = (speed/60.0)*(2.0*pi)*(d/2.0)
                  vyzlim = vf*tan(gamma)
c
```

```
                  tzmin = 9999
                  tzmax = 0
c
c ************************************
c
c Loop for each tooth
c
c ************************************
c
                  do 66 k = 1, tth
                     krl = k
                     phi1 = phi + (krl - 1.0)*tphi
                     revrl = phi1/(2*pi)
                     rev = revrl
                     revrl = rev
                     phi1 = phi1 - (revrl*2*pi)
                     phi2 = phi1
c
c ************************************
c
c Loop for each slice of depth of cut
c
c ************************************
c
                     do 77 j = 1, numofb
                        xi = 2.0*db*tan(helix)/d
c
c Determine if tooth is in cut
c
                        if ((phi2 .gt. phis).and.(phi2 .lt. phie)) then
c
c Determine chip thickness for each slice
c
                           if (.not. plough) then
                              tzrl = phi2/dphi
                              tz = tzrl + 0.01
                              if (tz .lt. 1) then
                                 tz = 1
                              end if
                              zminl(j) = zmin(tz,j)
                           else
                              tzrl = (phi2 + (plfrl - pcrl)*dphi2)/dphi
                              tz = tzrl + 0.01
                              if (tz .lt. 1) then
                                 tz = 1
                              end if
                              if (.not. justpl) then
                                 zmin(tz,j) = zminl(j)
                              end if
                           end if
c
                           z = xx*sin(phi2) + yy*cos(phi2)*ms(j)
                           h = c*sin(phi2) - (z -aro(k)) + zmin(tz, j)
c
c update surface at normal dphi increments only
c
                           if ((.not. plough).or.(pcount .eq. plf)) then
                              if (h .le. 0.0) then
                                 zmin(tz, j) = zmin(tz, j) + c*sin(phi2)
                              else
                                 zmin(tz, j) = z - aro(k)
                              end if
c
c Surface finish approximation
c
                              if ((phi2 .ge. surfphi)
     *                           .and.(phi2 .le. (surfphi+dphi))) then
                                 dy = -z
                                 xc = fperth*(phi/(pi/2.0))
                                 if ((xc - xcl) .ge. (fperth/2.0)) then
c
```

```
                                    if (fchoice .ge. 1 ) then
                                       write(3, 234) (1000.0*xc),
     *                                (1000.0*dy)
                                    end if
234                                 format (f12.6, f12.6)
                                    if (t .ge. ts1) then
                                       if (abs(dy) .gt. dymax) then
                                          dymax = abs(dy)
                                          tdymax = t
                                       end if
                                       if (dy .lt. dymn) then
                                          dymn = dy
                                          ppdymax = dymx - dymn
                                       end  if
                                       if (dy .gt. dymx) then
                                          dymx = dy
                                          ppdymax = dymx - dymn
                                       end if
                                    end if
                                 end if
                                 xcl = xc
                              end if
c
c
                           end if
                           if (h .le. 0.0) then
                              h = 0.0
                              dfx = 0.0
                              dfy = 0.0
                           else
                              dft = db*Ks*h
                              dfr = rl*dft
c
c Ploughing model using friction theory
c
                              vyz(1) = -(yyd*cos(phi2)*ms(j) +
     *                                  xxd*sin(phi2))
                              if (((vyz(1).gt.0.0).and.(vb .gt. 0.0))
     *                           .or. (vyz(1) .gt. vyzlim)) then
                                 gammae = atan(vyz(1)/vf)
                                 lpl = h/sin(gamma)
                                 if (lpl .gt. lplmax) then
                                    lpl = lplmax
                                 end if
                                 if (vyz(1) .gt. vyzlim) then
                                    lpar = lpl*cos(gammae - gamma)
     *                                    + vb*cos(gammae)
                                    lpen = lpl*sin(gammae - gamma)
     *                                    + vb*sin(gammae)
                                 else
                                    lpar = vb*cos(gammae)
                                    lpen = vb*sin(gammae)
                                 end if
                                 dfplr = po*db*((lpar*sin(gammae))+
     *                                         (lpen*cos(gammae)))
                                 dfplt = po*db*((lpar*cos(gammae))-
     *                                         (lpen*sin(gammae)))
                                 dfr = dfr + dfplr
                                 dft = dft + dfplt
                              end if
                              dfx = dft*cos(phi2) + dfr*sin(phi2)
                              dfy = -dft*sin(phi2) + dfr*cos(phi2)
                              dfy = dfy * ms(j)
                           end if
                           fy = fy + dfy
                           fx = fx + dfx
                        end if
                        phi2 = phi2 - xi
77                   continue
                     if (tz .lt. tzmin) then tzmin = tz
                     if (tz .gt. tzmax) then tzmax = tz
66                continue
```

```
c
c Dynamic response
c
                  dt = dphi2/(2*pi*speed/60.0)
                  fxp = fx*cos(alpha) - fy*sin(alpha)
                  xdd(1) = (fxp - cx*xd(2) - kx*x(2))/mx
                  xd(1) = xd(2) + xdd(1)*dt
                  x(1) = x(2) + xd(1)*dt
c
                  fyp = fx*sin(alpha) + fy*cos(alpha)
                  ydd(1) = (fyp - cy*yd(2) - ky*y(2))/my
                  yd(1) = yd(2) + ydd(1)*dt
                  y(1) = y(2) + yd(1)*dt
c
                  if (pcount .ge. plf) then
                     plough = .false.
                     pcount = 0
                     dphi2 = dphi
                     i = i + 1
                     justpl = .true.
                  end if
c
c Check for ploughing by velocity at top of cut
c
c
                  if (.not. plough) then
c
                     xxd = xd(1)*cos(alpha) + yd(1)*sin(alpha)
                     yyd = -xd(1)*sin(alpha) + yd(1)*cos(alpha)
c
                     vyz(1) = -(yyd*cos(tzrl*dphi)*ms(numofb) +
     *                           xxd*sin(tzrl*dphi))
c
c Ploughing criteria
c
                     if ((((vyz(1) .gt. 0.0) .and. (vb .gt. 0.0))
     *                    .or. (vyz(1) .gt. vyzlim))
     *                    .or.((tzmin .eq. 1).or.(tzmax.ge.imax))) then
                        plough = .true.
                        ptimes = ptimes + 1.0
                        plfrl = plf
                        dphi2 = dphi/plfrl
                        pcount = 0
                        if (.not. justpl) then
c
                           x(1) = x(2)
                           xd(1) = xd(2)
                           xdd(1) = xdd(2)
c
                           y(1) = y(2)
                           yd(1) = yd(2)
                           ydd(1) = ydd(2)
c
                           x(2) = x(3)
                           xd(2) = xd(3)
                           xdd(2) = xdd(3)
c
                           y(2) = y(3)
                           yd(2) = yd(3)
                           ydd(2) = ydd(3)
c
c
                           phi = phi - dphi
                           t = t - dt
                           skip = .true.
                        end if
                     end if
                  end if
c
                  if (plough) then
                     pcount = pcount + 1
                  else
```

```
                  if (.not. justpl) then
                     i = i + 1
                  end if
                  justpl = .false.
               end if
c
               pcrl = pcount
               t = t + dt
c
c Tracking maximum forces, deflection, etc.
c
               xx = x(1)*cos(alpha) + y(1)*sin(alpha)
               yy = -x(1)*sin(alpha) + y(1)*cos(alpha)
c
               xxd = xd(1)*cos(alpha) + yd(1)*sin(alpha)
               yyd = -xd(1)*sin(alpha) + yd(1)*cos(alpha)
c
               if (t .ge. ts1) then
                  if (abs(xx) .gt. xmax) then
                     xmax = abs(xx)
                     txmax = t
                  end if
c
                  if (xx .lt. xmn) then
                     xmn = xx
                     ppxmax = xmx - xmn
                  end  if
                  if (xx .gt. xmx) then
                     xmx = xx
                     ppxmax = xmx - xmn
                  end if
c
                  if (abs(fx) .gt. fxmax) then
                     fxmax = abs(fx)
                     tfxmax = t
                  end if
c
                  if (abs(yy) .gt. ymax) then
                     ymax = abs(yy)
                     tymax = t
                  end if
c
                  if (yy .lt. ymn) then
                     ymn = yy
                     ppymax = ymx - ymn
                  end  if
                  if (yy .gt. ymx) then
                     ymx = yy
                     ppymax = ymx - ymn
                  end if
c
                  if (abs(fy) .gt. fymax) then
                     fymax = abs(fy)
                     tfymax = t
                  end if
               end if
c
c
               if ((abs(xx) .gt. 0.01) .or.
     *              (abs(yy) .gt. 0.01)) then
                  i = imax + 1
                  p = revs
               end if
c
c
               x(3) = x(2)
               xd(3) = xd(2)
               xdd(3) = xdd(2)
c
               y(3) = y(2)
               yd(3) = yd(2)
               ydd(3) = ydd(2)
```

```
c
                  x(2) = x(1)
                  xd(2) = xd(1)
                  xdd(2) = xdd(1)
c
                  y(2) = y(1)
                  yd(2) = yd(1)
                  ydd(2) = ydd(1)
c
c
c
c Writing to output files as necessary
c
                  if ((fchoice .ge. 2) .and. (.not. skip)) then
                     if (((t .ge. ts1) .and. (t .le. te1)) .or.
     *                   ((t .ge. ts2) .and. (t .le. te2))) then
                        if ((t .ge. tt1) .or. (t .ge. tt2)) then
                           tt1 = tt1 + (1.0/fs)
                           tt2 = tt2 + (1.0/fs)
                           write (8, 2012) t, fx, (1000.0*xx),
     *                                     fy, (1000.0*yy)
2012                       format (' ',f8.6,f14.4, f12.6, f14.4, f12.6)
                        end if
                        if (fchoice .eq. 3) then
                           write (7, 201) t, tz, h, dphi2,
     *                        fx, (1000.0*xx),
     *                        fy, (1000.0*yy), (1000.0*vyz(1)),
     *                        (1000.0*vyzlim), (gammas*180.0/pi)
201                        format (' ', f8.6, i4, f9.6, f9.6,f10.4,
     *                        f12.6, f14.4, f12.6, f14.4, f12.6, f10.4)
                        end if
                     end if
                  end if
c
                  skip = .false.
               end do
               p = p + 1
            end do
c
c
            imaxrl = imax
            itotal = prl * imaxrl
            ppercent = ptimes / itotal
            write (2, 212) rpm, (1000.0*b), (1000.0*ppdymax),
     *         (1000.0* dymax), tfxmax, fxmax, (1000.0*ppxmax),
     *         (1000.0*xmax),(1000.0*xlim/xlimf),tfymax,fymax, (1000.0*
     *         ppymax), (1000.0*ymax),(1000.0*ylim/ylimf), ppercent
            print 212, rpm , (1000.0*b), (1000.0*ppdymax),
     *         (1000.0* dymax), tfxmax, fxmax, (1000.0*ppxmax),
     *         (1000.0*xmax),(1000.0*xlim/xlimf),tfymax,fymax, (1000.0*
     *         ppymax), (1000.0*ymax),(1000.0*ylim/ylimf), ppercent
c
212         format (' ', f6.0, f8.3, f9.5, f9.5, f8.4, f9.2, f8.4, f9.5,
     *              f9.5, f8.4, f9.2, f8.4, f9.5, f9.5, f8.4)
c
c
            n = n + 1
            b = b * bf
         end do
11    continue
c
c close output files
c
      close(2)
      close(3)
      close(4)
      close(7)
      close(8)
      close(9)
      end
```

```
/*                                                              */
/*              Data Acquisition                                */
/*              and FFt program for dalcano sprY model 25 board */
/*              (DAFFY7.C - chatter avoidance version 1.1)      */
/*                        - and sofware data windowing          */
/*                                                              */
/*              updated : june 27, 1990                         */
/*              compiler: microsoft c                           */

#include        <math.h>
#include        <time.h>
#include        <stdio.h>
#include        <graph.h>
#include        <conio.h>

void get_inputs(void);
void set_constants(void);
int  peak_find(int vcs);
void control(void);
void vsd(int ns, int dns, int nsf);
void send_vsd(int indata, int ind);
void wait(unsigned int portno, char and_var, char xor_var);
void timer(int freq);

#define clock0  0x4
#define clock1  0x5
#define clock2  0x6
#define cntrlc  3
#define cntler  7

#define io320   0x300
#define seg320  0xd000
#define rseg    0x200
#define iseg    0x600
#define rseg1   0x201
#define iseg1   0x601
#define pklngth 0x38
#define finish  0x0058

#define pts     512
#define pts_1   511
#define pts2    256

#define cmmdreg 0x378
#define sttsreg 0x379
#define datareg 0x37a

#define no      0
#define yes     1
#define YES     2
#define zero    0
#define styfr   64

#define offset  170
#define pi      3.1416


unsigned dapgm[0x38] =
/*    0 */  {  0xFF80, 0x10, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0,
/*    8 */     0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0,
/*   10 */     0xCE01, 0xC804, 0xCA00, 0xCA49, 0x6000, 0XE700, 0xCA01, 0x6001,
/*   18 */     0xD100, 0x200, 0xD300, 0x200, 0xD200, 0x200, 0x5589, 0xFA80,
/*   20 */     0x23, 0xFF80, 0x1F, 0x83AB, 0xE4AA, 0xFB99, 0x1F, 0xD001,
/*   28 */     0x200, 0xD400, 0x200, 0x558C, 0xCBFF, 0x59A0, 0xD001, 0x300,
/*   30 */     0xD400, 0x300, 0xCBFF, 0x59A0, 0xCA58, 0x6000, 0xE700, 0x0};

unsigned pgm1a[0x38] =

/*    0 */  {  0xFF80, 0x9FC, 0xF900, 0x2, 0x200, 0x0, 0x192, 0x324,
/*    8 */     0x4B6, 0x647, 0x7D9, 0x96A, 0xAFB, 0xC8B, 0xE1B, 0xFAB,
/*   10 */     0x1139, 0x12C8, 0x1455, 0x15E2, 0x176D, 0x18F8, 0x1A82, 0x1C0B,
/*   18 */     0x1D93, 0x1F19, 0x209F, 0x2223, 0x23A6, 0x2528, 0x26A8, 0x2826,
```

```
/*   20 */     0x29A3, 0x2B1F, 0x2C98, 0x2E11, 0x2F87, 0x30FB, 0x326E, 0x33DE,
/*   28 */     0x354D, 0x36BA, 0x3824, 0x398C, 0x3AF2, 0x3C56, 0x3DB8, 0x3F17,
/*   30 */     0x4073, 0x41CE, 0x4325, 0x447A, 0x45CD, 0x471C, 0x4869, 0x49B4};

unsigned pgm1b[0x50] =

/*   38 */  {  0x4AFB, 0x4C3F, 0x4D81, 0x4EBF, 0x4FFB, 0x5133, 0x5269, 0x539B,
/*   40 */     0x54CA, 0x55F5, 0x571D, 0x5842, 0x5964, 0x5A82, 0x5B9D, 0x5CB4,
/*   48 */     0x5DC7, 0x5ED7, 0x5FE3, 0x60EC, 0x61F0, 0x62F1, 0x63EF, 0x64E8,
/*   50 */     0x65DD, 0x66CF, 0x67BD, 0x68A6, 0x698C, 0x6A6D, 0x6B4A, 0x6C24,
/*   58 */     0x6CF9, 0x6DCA, 0x6E96, 0x6F5E, 0x7023, 0x70E2, 0x719E, 0x7255,
/*   60 */     0x7307, 0x73B5, 0x745F, 0x7504, 0x75A5, 0x7641, 0x76D9, 0x776C,
/*   68 */     0x77FA, 0x7884, 0x7909, 0x798A, 0x7A05, 0x7A7D, 0x7AEF, 0x7B5D,
/*   70 */     0x7BC5, 0x7C29, 0x7C89, 0x7CE3, 0x7D39, 0x7D8A, 0x7DD6, 0x7E1D,
/*   78 */     0x7E5F, 0x7E9D, 0x7ED5, 0x7F09, 0x7F38, 0x7F62, 0x7F87, 0x7FA7,
/*   80 */     0x7FC2, 0x7FD8, 0x7FE9, 0x7FF6, 0x7FFD, 0x7FFF, 0x0, 0x0};


unsigned pgm2[0xd0] =

/*  9F8 */ {   0x0, 0x0, 0x0, 0x0, 0x558D, 0xC500, 0x0, 0xCE01,
/*  A00 */     0xC804, 0xCA49, 0x6001, 0xE701, 0xC804, 0xCA01, 0x6001, 0x3C01,
/*  A08 */     0xA004, 0xCE14, 0x5800, 0x2000, 0x1001, 0x6002, 0x2F00, 0x6803,
/*  A10 */     0x2E00, 0x6804, 0x3C01, 0xA005, 0xCE14, 0x6019, 0xA200, 0xCE14,
/*  A18 */     0x601A, 0xA600, 0xCE14, 0x101A, 0x601B, 0xCA00, 0x600A, 0x600B,
/*  A20 */     0x4102, 0x1101, 0x100A, 0xF380, 0xA50, 0x410B, 0x100A, 0xF280,
/*  A28 */     0xA3D, 0x201A, 0xB, 0x5814, 0x1B, 0x5816, 0x201A, 0xA,
/*  A30 */     0x5815, 0x1B, 0x5817, 0x201A, 0xB, 0x5915, 0x1B, 0x5917,
/*  A38 */     0x201A, 0xA, 0x5914, 0x1B, 0x5916, 0x2003, 0x6018, 0x200B,
/*  A40 */     0x1018, 0xF380, 0xA48, 0x600B, 0x2F18, 0x6818, 0xFF80, 0xA3F,
/*  A48 */     0x200B, 0x18, 0x600B, 0x410A, 0x1, 0x600A, 0xFF80, 0xA20,
/*  A50 */     0xCA00, 0x6005, 0x6006, 0xCA01, 0x6007, 0x6008, 0x2104, 0x6009,
/*  A58 */     0x2007, 0x600A, 0x7, 0x6007, 0xCA00, 0x600B, 0x200A, 0x100B,
/*  A60 */     0xF280, 0xAB6, 0x2019, 0x5, 0x5813, 0x2019, 0x4, 0x1005,
/*  A68 */     0x5812, 0x200B, 0x1008, 0xF380, 0xA75, 0x2005, 0x1006, 0x6005,
/*  A70 */     0xCA00, 0x1012, 0x6012, 0xFF80, 0xA78, 0x2005, 0x6, 0x6005,
/*  A78 */     0x200B, 0x600C, 0x2002, 0x100C, 0xF380, 0xAB1, 0x200A, 0xC,
/*  A80 */     0x600D, 0x201A, 0xD, 0x580E, 0x1B, 0x580F, 0x201A, 0xC,
/*  A88 */     0x5810, 0x1B, 0x5811, 0xCA00, 0x3C12, 0x380E, 0x3D13, 0x380F,
/*  A90 */     0x3D12, 0x6814, 0xCA00, 0x380F, 0x3D13, 0x380E, 0xCE16, 0x6815,
/*  A98 */     0x2E10, 0xF14, 0x6910, 0x4414, 0x690E, 0x2E11, 0xF15, 0x6911,
/*  AA0 */     0x4415, 0x690F, 0x201A, 0xD, 0x590E, 0x1B, 0x590F, 0x201A,
/*  AA8 */     0xC, 0x5910, 0x1B, 0x5911, 0x200C, 0x7, 0x600C, 0xFF80,
/*  AB0 */     0xA7A, 0x200B, 0x1, 0x600B, 0xFF80, 0xA5E, 0x200A, 0x6008,
/*  AB8 */     0x2F09, 0x6809, 0x6806, 0x2003, 0x100A, 0xF180, 0xA58, 0xCA58,
/*  AC0 */     0x6001, 0xE701, 0xFF80, 0xAC2, 0x6880, 0x3800, 0x6890, 0x201A};

int       sampling, loopi, teeth, graphic, iave, xi[513], zr[513], zi[513];
float     amp[513], vamp[513], w[513], n, nfac, dnmax, dnstep, nfmax, nfstep, thresfac,
          osfreq, oscamp, dfreq, nz, thres, peak, peak2, chatterf, chatterf2,
          gain;
FILE      *f1, *f2, *f3;
int       temp, window;

/*  Input file

        Example             Explanation

     ctest9da.prn        file name for frequency spectrum averages (before and after trigger)
     ctest9db.prn        file name for freqeuncy spectrum data for each iteration
     25                  number of iterations
     3200                sampling freqeucny of data acquisition
     4                   number of teeth
     0                   file option : 1- text, 2 graphics, 3 both (2 & 3 not recommended)
     1000.0              spindle speed
     2.22                speed factor between spindle speed and controller stator speed
     50.0                maximum speed variation amplitude
     50.0                speed variation amplitude increment
     2.5                 maximum speed variation frequency
     2.5                 speed variation frequency increment
     5.0                 threshold factor for chatter avoidance
     200.0               highest frequency for low freqeuncy average
```

```
     200.0              lowest freqeuncy for peak search
     1200.0             highest frequency for peak search
     1                  1 for time data windowing (hanning)
     1.0                gain for data acquisiiton
*/

main()
{
        int     j, k, i, unstable, bsfrq, xr[513], trans, itrans;
        float   freq;
        double  tused, tperstep;
        time_t  tstart, tstop;
        char    var, dummy;

        get_inputs();
        set_constants();

        bsfrq = (int)(255.0*n/61.0);
        vsd(bsfrq,0,0);
        osfreq = 0.0;
        oscamp = 0.0;
        trans = no;
        _settextposition(10, 1);
        printf("    \n ");
        _settextposition(7, 1);
        printf("loop = %d \n", k);
        printf("vsc = %d \n", trans);
        printf("A/D in progress \n");
        time(&tstart);
        for (i = 1; i <= loopi; i++)
        {
            peekwf(rseg1,seg320,&zr[0],pts_1);
            peekwf(iseg1,seg320,&zi[0],pts_1);
            pokewf(dapgm,zero,pklngth,seg320);
            go320(io320);
            pokewf(xi,iseg,pts,seg320);
            if (i > 1) unstable = peak_find(trans);
            while(regin(io320) != finish) {}
            hlt320(io320);
            pokewf(pgm1a,zero,pklngth,seg320);
            peekwf(rseg,seg320,&xr[0],pts);
            if (window == yes)
            {
                for (j=0; j <pts; j++) xr[j] = (int)(((float)xr[j] - 2048.0)*w[j] + 2048.0);
            }
            pokewf(xr,rseg,pts,seg320);
            go320(io320);
            if ((unstable == yes) && (i != 1))
            {
               if (trans == no)
               {
                   trans = yes;
                   itrans = i;
               }
               control();
            }
            while(regin(io320) != finish) {}
            hlt320(io320);
            fprintf(f3, "%d %d %.3f %.3f %.3f %.3f %.3f %.3f %.3f %.3f \n", k, i,
                        n, oscamp, osfreq, thres, peak, chatterf, peak2, chatterf2);
        }
        time(&tstop);
        _settextposition(9, 1);
        printf("A/D pending     \n");
        fprintf(f2, "\n %d \n %.3f \n %.3f \n %.3f \n %.3f \n %.3f \n %.3f \n\n",
                        k, n, oscamp, osfreq, thres, peak, chatterf);
        freq = 0.0;
        for (j = 0; j < pts2; j++)
        {
             amp[j] = amp[j]/((float)itrans - 1.0);
             vamp[j] = vamp[j]/((float)(loopi - trans));
             freq += dfreq;
```

```
            fprintf(f2, "%.3f %.3f %.3f \n", freq, amp[j], vamp[j]);
        }
        co80320();
        tused = difftime(tstop, tstart);
        tperstep = tused/(double)loopi;
        printf(" Total time = %f seconds \n", tused);
        printf(" Time per iteration = %f milliseconds \n", tperstep*1000.0);
        fcloseall();
}

void get_inputs(void)
{
        int k;
        char fname1[11], fname2[11];

        f1 = fopen("daffy7.dat", "r");
        printf(" Enter the file name for control data: \n");
        fgets(fname1, 13, f1);
        f2 = fopen(fname1, "w");
        fgetc(f1);
        printf(" Enter the file name for fft data: \n");
        fgets(fname2, 13, f1);
        f3 = fopen(fname2, "w");

        printf( " Enter the number of iterations per save (~50) : \n");
        fscanf(f1,"%d",&loopi);
        printf( " Enter the sampling frequency (~2500): \n");
        fscanf(f1,"%d",&sampling);
        printf( " Enter the number of teeth (~4): \n");
        fscanf(f1,"%d",&teeth);
        printf( " Enter the display option (<0> = txt, <1> = graphics, <2> g+t): \n");
        fscanf(f1,"%d",&graphic);
        printf( " Enter the nominal spindle speed (~600.0): \n");
        fscanf(f1,"%f",&n);
        printf( " Enter the stator speed/spindle speed factor (~2.2): \n");
        fscanf(f1,"%f",&nfac);
        printf( " Enter the maximum amplitude of variation (~60.0): \n");
        fscanf(f1,"%f",&dnmax);
        printf( " Enter the amplitude of variation step (~10.0): \n");
        fscanf(f1,"%f",&dnstep);
        printf( " Enter the maximum frequency of variation (~6.0): \n");
        fscanf(f1,"%f",&nfmax);
        printf( " Enter the frequency of variation step (~1.0): \n");
        fscanf(f1,"%f",&nfstep);
        printf( " Enter the treshold factor (~2.0): \n");
        fscanf(f1,"%f",&thresfac);
        printf( " Enter the number of points for low freq averaging (~10): \n");
        fscanf(f1,"%d",&iave);
        printf( " Enter (1) for windowing : \n");
        fscanf(f1,"%d",&window);
        printf( " Enter the gain for window (~5.0 for proximitor): \n");
        fscanf(f1,"%f",&gain);
}

void set_constants(void)
{
        int k, nstart;

        hlt320(io320);
        timer(sampling);
        pokewf(pgm1a,zero,pklngth,seg320);
        pokewf(pgm1b,pklngth,0x50,seg320);
        pokewf(pgm2,0x9f8,0xd0,seg320);

        dnmax = dnmax*nfac/60.0;
        dnstep = dnstep*nfac/60.0;
        n = n*nfac/60.0;

        outp(datareg, 0);
        vsd(0,0,0);
        osfreq = 0.0;
        oscamp = 0.0;
```

```
        nstart = (int)(255.0*n/61.0);
        vsd(nstart, 0, 0);

        for (k = 0; k < pts; k++) w[k] = (sin((pi*(float)k)/((float)pts-1.0)))*
                gain*(sin((pi*(float)k)/((float)pts-1.0)));
        for (k = 0; k < pts; k++) xi[k]=0;
        for (k = 0; k < pts2; k++) amp[k] = vamp[k] = 0.0;
        pokewf(xi,iseg,pts,seg320);

        dfreq = (float)sampling/(2.0*(float)pts2);
        nz = n*(float)teeth;

        _clearscreen(_GCLEARSCREEN);
        _settextposition(1, 1);
        printf("n = %.2f \n", n);
        if ((graphic == yes) || (graphic == YES))
        {
                hres320();
                _settextposition(1, 1);
                printf("n = %.2f \n", n);
        }

}

int peak_find(int vcs)
{
        int   i, k, y[513], p[513], chatter;
        long  mag2, mmag2, mmag22, mag2ave;

        mmag2 = 0;
        mmag22 = 0;
        mag2ave = 0;
        for (i = (pts2 - 1); i >= 0; i--)
        {
                mag2 = (long)zr[i]*(long)zr[i] + (long)zi[i]*(long)zi[i];
                y[i] = (int)sqrt((double)mag2);
                p[i] = offset - y[i];
                if (vcs == no)
                {
                        amp[i] += (float)y[i];
                }
                else
                {
                        vamp[i] += (float)y[i];
                }
                if (mag2 > mmag2)
                {
                        mmag2 = mag2;
                        chatterf = ((float)i+1.0)*dfreq;
                        if (i >= 50)
                        {
                                mmag22 = mag2;
                                chatterf2 = ((float)i+1.0)*dfreq;
                        }
                }
                if (i <= iave) mag2ave += mag2;
        }
        if (graphic == no)
        {
                _settextposition(2, 1);
                printf("a = %.2f \n", oscamp);
                printf("f = %.2f \n", osfreq);
                printf("t = %.2f \n", thres);
                printf("p = %.1f \n", peak);
                printf("f = %.1f \n", chatterf);
        }
        else
        {
                if (graphic == yes)
                {
                        sigpts(&p[0], pts2);
                }
```

```
                else
                {
                        _settextposition(2, 1);
                        printf("a = %.2f \n", oscamp);
                        printf("f = %.2f \n", osfreq);
                        printf("t = %.2f \n", thres);
                        printf("p = %.1f \n", peak);
                        printf("f = %.1f \n", chatterf);
                        sigpts(&p[0], pts2);
                }
        }
        mag2ave /= iave;
        thres = thresfac*(float)sqrt((double)mag2ave);
        peak = (float)sqrt((double)mmag2);
        peak2 = (float)sqrt((double)mmag22);
        if (peak > thres)
        {
                chatter = yes;
        return(chatter);
        }
        else
        {
                chatter = no;
                return(chatter);
        }
}

void control(void)
{
        int   bsfrq, osmp, osfrq;

        osfreq += nfstep;
        oscamp += dnstep;
        if (osfreq > nfmax) osfreq = nfmax;
        if (oscamp > dnmax) oscamp = dnmax;
        bsfrq = (int)(255.0*n/61.0);
        osmp  = (int)(255.0*oscamp/10.0);
        osfrq = (int)(255.0*osfreq/20.0);
        vsd(bsfrq, osmp, osfrq);
}

void vsd(int ns, int dns, int nsf)
{
        send_vsd(nsf, 7);
        send_vsd(dns, 6);
        send_vsd(ns, 5);
}

void send_vsd(int indata, int ind)
{
        outp(cmmdreg, indata);
        outp(datareg, ind);
        wait(sttsreg, styfr, zero);
        outp(datareg, 0);
}


void wait(unsigned int portno, char and_var, char xor_var)
{
        char result;

        do
        {
                result = inp(portno);
                result = ((result^xor_var) & and_var);
        }
        while (result == 0x00);
}


void timer(int freq)
{
```

```
        int tim, mode, cntrlw, k, hinum, lonum;
        unsigned int num;

        tim = 2;
        mode = 2;
        cntrlw = 0xb4;
        num = 10000000/freq;
        hinum = num >> 8;
        lonum = num & 0xff;
        outp((io320+cntler),cntrlw);
        outp((io320+clock0+tim),lonum);
        outp((io320+clock0+tim),hinum);
}
```