COMBINING TOOL WEAR AND DYNAMICS IN HIGH-SPEED MACHINING PERFORMANCE PREDICTION

By

JAYDEEP MOHAN KARANDIKAR

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

2010

© 2010 Jaydeep Mohan Karandikar

To my family

ACKNOWLEDGMENTS

I would like to thank all members of my committee, Dr. Tony Schmitz, Dr. John Schueller and Dr. Hitomi Greenslet for their support and involvement with this project and taking the time to review this thesis. A special thanks to my advisor, Dr. Tony Schmitz for giving me an opportunity to work in a great research environment. His advice and guidance helped a lot through my graduate studies. This work would not have been possible without support from Thomas Long and Srikanth Bontha of Kennametal, Inc. I would also like to thank all my lab mates at Machine Tool Research Center for their friendship and help, especially Raul Zapata. Last, but in no way the least, I would like to thank my family members for their constant encouragement and unwavering belief in me.

TABLE OF CONTENTS

page

APPENDIX

LIST OF TABLES

LIST OF FIGURES

Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

COMBINING TOOL WEAR AND DYNAMICS IN HIGH-SPEED MACHINING PERFORMANCE PREDICTION

By

Jaydeep Mohan Karandikar

May 2010

Chair: Tony L. Schmitz
Major: Mechanical Engineering

A milling “super diagram” is described that incorporates limitations to milling productivity and part quality imposed by stability, surface location error (part errors due to forced vibrations), and tool wear. Combinations of axial depth of cut and spindle speed that offer stable cutting conditions with an acceptable, user-defined surface location error level are identified by a gray-scale color coding scheme. The effect of tool wear is incorporated by determining the variations in force model coefficients used for process dynamics prediction with tool wear. The force model coefficients increase as a function of the flank wear width (used as a measure of tool wear). The increase in force model coefficients is determined as a function of the volume of material removed. Using these coefficients, a super diagram is constructed for any user-defined volume of material removed using the selected cutter. Additionally, user beliefs about data and model accuracy are applied to identify safety margins relative to the deterministic boundaries in the diagrams.

Experimental results are provided for an inserted (carbide) cutter used to machine 1018 steel. The wear behavior is incorporated as changes in the force model coefficients as a function of the volume of material removed at different operating

parameters. The flank wear is also measured using an on-machine microscope (to avoid tool removal from the spindle) and correlated to the force model coefficients. Super diagrams are developed that correspond to the new and worn tool performance and experimental results are provided to verify changes in the process stability with tool wear. Once-per-revolution sampling is used to determine a variance parameter that is subsequently used to identify stable and unstable cuts. Test results are shown that compare the distribution in once-per-revolution sampled force data for new and worn inserts at various axial depths of cut. Synchronous (stable) behavior is characterized by a tight distribution, while asynchronous (unstable) behavior yields larger distributions. A normalized variance parameter is used to separate the two cases.

CHAPTER 1
INTRODUCTION

Motivation and Research Objective

High speed machining (HSM) has made significant technological advances in recent years. Improved spindle designs enable speeds up to 20000 rpm and higher in milling. High material removal rates (MRR) can be obtained by machining at higher axial depths of cut and spindle speeds. However, a limitation to machining at higher axial depths of cut is chatter or unstable cutting. Tlusty defined the mechanism for chatter or self excited vibrations as regeneration of surface waviness during material removal [1]. Subsequent work involved developing techniques like time domain simulations, frequency domain analyses or temporal finite element methods to predict the stability lobes in HSM [2-5]. Stability lobe diagrams separate stable cutting conditions from unstable or chatter conditions and are represented as a function of axial depth of cut and spindle speed. The models used to predict the stability lobes require the tool point frequency response function (FRF), tool geometry, cutting parameters and cutting force coefficients.

Tool wear is also an important process limitation in machining. Taylor established empirical relationships between cutting parameters and tool wear, which is still used as the basis for predicting tool life [6]. Tool wear is primarily dependent on cutting speed, which has a strong influence on the temperatures in cutting. Higher maximum temperatures increase the severity of thermal stress cycling [7]. Tool condition monitoring (TCM) research has developed methods to estimate tool wear based on cutting force signals. Research has shown an increase in the cutting force and force coefficients due to tool wear [8-15]. However, the effect of tool wear on stability lobe

diagrams has not been previously explored. The tool wear effects can be incorporated through the force model coefficients used to calculate the stability lobes. This research studies the role of tool wear as a process limitation through its effect on milling stability lobes. The final objective is to present a milling super diagram which simultaneously incorporates stability, surface location error, tool wear and uncertainty. The super diagram will enable a user to select optimum cutting conditions to maximize MRR and reduce cost considering these process limitations.

Stability Lobes in Milling

In milling, relative motion between a rotating cutter and workpiece is responsible for material removal. As the cutter is engaged, it experiences a cutting force which causes the tool to vibrate. The tool vibrations are imprinted on the workpiece leaving behind a wavy surface. The wavy surface left behind by one tooth is removed by the following tooth. Thus, surface regeneration occurs from one tooth to the next. The instantaneous chip thickness depends on the state of vibration of the current tooth and the surface left behind by the previous tooth and governs the cutting force. If the two surfaces are in phase, the chip thickness varies only according to cut geometry. This gives periodic cutting forces and tool vibrations and provides stable cutting conditions. However, an out of phase profile results in a variable chip thickness which affects the cutting force, and subsequently, the tool vibrations. The resulting vibrations again affect the chip thickness. This feedback mechanism may result in self-excited vibrations or chatter in milling.

Stability lobes separate stable operating points from unstable or chatter points. All operating points below the stability boundary are stable and the ones above are unstable. Figure 1.1 shows typical stability lobes in milling.

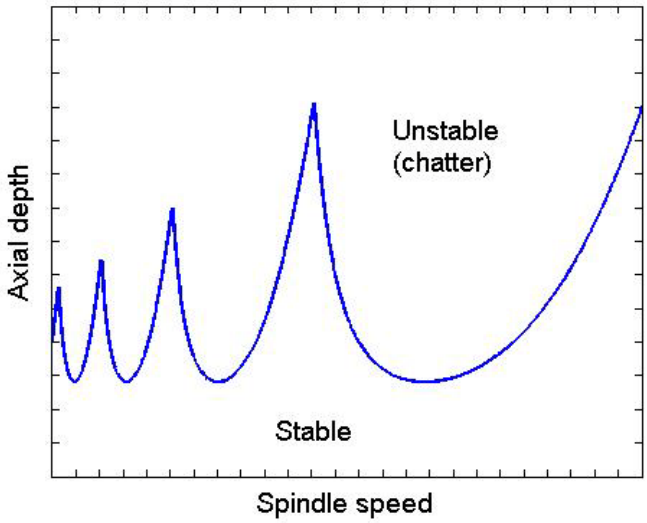


Figure 1-1. Typical stability lobes in milling identifying stable and unstable (chatter) zones

The stability lobes are calculated at a certain radial depth of cut and feed per tooth. The user can select optimum operating conditions for spindle speed and axial depth of cut based on this diagram.

Tool Wear

Tool wear in machining is the loss of tool material due to interaction with the workpiece during cutting. Tool wear can ultimately result in catastrophic failure of the cutting edge. Tool wear is undesirable as it affects the cutting forces and quality of the machined surface. Replacing a worn tool increases tool change time, which increases the cost of the product. Tool wear also results in increased cutting forces and temperatures. The various mechanisms that can cause tool wear include mechanical microbreakages, abrasion, adhesion, diffusion and oxidation.

Taylor established empirical relationships between tool life and cutting parameters [6]. The Taylor-type tool life equation is given by $T = Cv^{-p}f_t^{-q}$, where T is tool life in

minutes, v is surface speed of the cutter in mm/min (given by $v = \frac{\pi d \Omega}{60}$, d is tool diameter in mm and Ω is spindle speed in rpm) and f_t is the feed per tooth in mm/tooth. The constant C and exponents p and q can be calculated from the experiments where the time to reach a predetermined wear level is measured for various cutting conditions. The exponent p is usually higher than q, which indicates a greater dependence of tool life on the spindle speed than on feed per tooth.

Tool wear features can be flank wear, crater wear and notch wear. Figure 1.2 shows the various forms of tool wear.

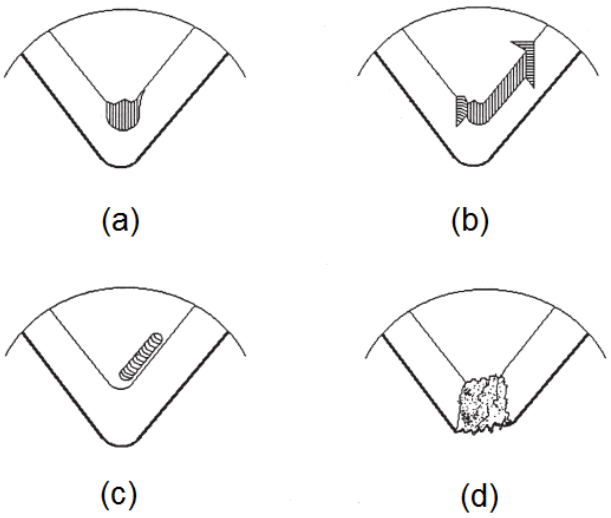


Figure 1-2. Various forms of tool wear – A) Nose wear B) Notch and Flank wear C) Crater wear D) Plastic/breakage [16].

Flank wear is the most common of the wear features and can be used to monitor tool wear. Flank wear is caused by abrasive wear of the main cutting edge against the workpiece and occurs on the tool flank face over the length equal to axial depth of cut in zero helix end milling. Flank wear is expressed in terms of flank wear width (FWW). Flank wear increases with cutting time as shown in Fig. 1.3 [7]. The tool life is based on

the time required for the maximum FWW to reach a preselected value. According to ISO, the permissible average value of FWW is 0.3 mm in the case of uniform wear or 0.6 mm maximum in case of irregular wear for cemented carbides

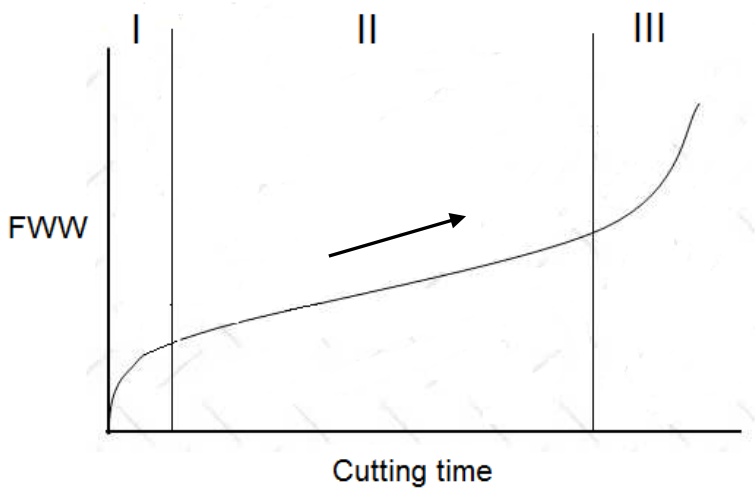


Figure 1-3. Increase in flank wear width with cutting time. – I) Initial rapid wear II) Uniform wear III. Final wear (catastrophic failure) [17].

The increase in flank wear width consists of three parts:

- Initial rapid wear where FWW increases rapidly.
- Uniform wear where FWW increases at a uniform rate.
- Final accelerated wear leading to a catastrophic failure of the tool

Surface Location Error (SLE)

In milling the tool experiences a cutting force which causes it vibrate when it engages the workpiece. Thus, the tool undergoes periodic forced vibrations, even under stable cutting conditions. These forced vibrations can cause surface location errors or geometric inaccuracies of the workpiece. The Surface Location Error (SLE) is given by the difference between the location of the intended final surface and the machined surface. The position of the cutting edge as it enters the cut in up milling or exits the cut in down milling as the tool vibrates determines the location of the machined surface. Depending on the state of the tool vibration as it leaves the final machined surface, it

can be undercut or overcut. Figure 1-2 shows an example of SLE in down milling. Because SLE is caused by forced vibration, its value depends on the spindle speed (excitation frequency).

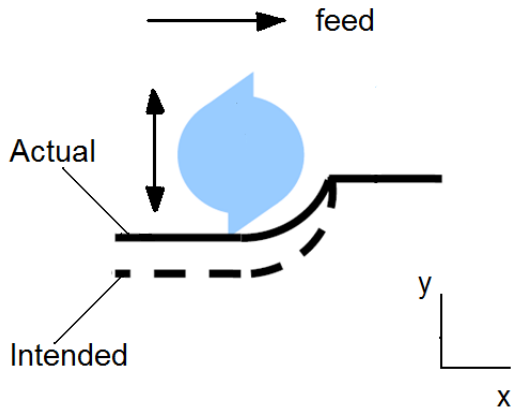


Figure 1-4. Example of an undercut in down milling. The difference between actual and intended surface is SLE. The feed of the tool is in x direction and cutting forces cause it vibrate in the y direction. The tool vibrations are exaggerated for depiction purposes.

SLE can be predicted using the same inputs required for stability prediction, including the tool point frequency response function, force model coefficients (dependent on the workpiece materials), geometry of tool and cutting parameters. A frequency domain approach calculates SLE in the same axial depth of cut - spindle speed domain as the frequency domain analysis [18]. The SLE graph can be shown as contours of constant SLE values.

CHAPTER 2
LITERATURE REVIEW

The literature review presents prior research in: 1) tool condition monitoring using cutting force signals; and 2) techniques to predicting stability during machining. Both topics are critical for the scope of this thesis.

Extensive research has been completed to better understand tool wear mechanisms and approaches to reduce tool wear. Taylor defined an empirical relation to calculate tool life by means of a tool life equation, which is still used as a basis for defining tool wear [6]. Subsequent research studied mechanisms of tool wear more closely for various materials and explored influencing factors. The effect of various tool materials and coatings on tool wear was studied [19-29]. Recently, efforts have been dedicated to tool conditioning monitoring (TCM). Multiple sensor systems can be used to monitor various signals during machining to determine in-process tool wear status [30-31, 16]. Most TCM systems use indirect measurements of cutting performance to predict tool wear. In many cases cutting force is measured using a table-mounted or rotating spindle dynamometer [8-15, 32-34]. The dynamometers measures three mutually perpendicular forces in the x (feed), y and z (axial) directions. Time series modeling of the x and y direction forces can then used to estimate the tool wear status. The increase in dynamic and static components of force signals has been found to increase with tool wear [9, 32]. Some studies measured the change in magnitude of cutting force harmonics in the frequency domain using the Fourier transform of the force signal [8, 15]. For example, a broken tooth can be detected by the corresponding increase in the magnitude of the cutting force harmonics using a pre-defined threshold

value. A cutting force coefficient based method to monitor tool wear was developed which showed the force coefficients increase with tool wear as well [33-34].

Chatter in high speed machining has been the subject of research studies for many years. Tlusty was among the first to develop an analytical method for predicting chatter in milling by means of stability lobes [1]. Tlusty assumed an average angle of the tooth in the cut and, therefore an average force direction. This eliminated the time dependence of cutting force direction in milling and simplified the analysis. Based on Tlusty's analysis, stability lobes for any machine-tool holder assembly can be calculated using an appropriate force model, geometric properties of the tool (number of teeth and helix angle) and parameters of the machining operations (radial depth of cut and feed per tooth). Altintas et al. defined an alternative technique where the time dependent coefficients for the milling force equations are expanded into a Fourier series and the series is truncated to include only the average term. This technique also provides a closed form solution [35]. However, both the theories have been found to be invalid for low radial immersions (less than 20%) where the cutting forces resemble short duration impulses [36]. Research has shown that the instability for low radial immersion is due to quasi-periodic chatter (referred to as flip bifurcation), which manifests itself as two tightly grouped clusters of points after once per revolution sampling. Therefore, techniques such as Poincare sectioning (once per revolution sampling) can be used to indentify instability [2-5, 37-43]. Other techniques like time domain analysis, semi-discretization method and temporal finite element analysis can also be used to predict stability [2-5]. More recently, a milling super diagram, which combines stability with surface location

error has been developed. The super diagram uses the frequency domain analysis for predicting stability and surface location error [18,44].

CHAPTER 3
SUPER DIAGRAM DESCRIPTION

Concept

A milling stability lobe diagram gives information about stable and unstable (chatter) points over axial depth of cut - spindle speed domain. A frequency domain approach can be used to calculate the deterministic stability boundaries [35]. The method offers a convenient closed form solution to calculate the stability lobes. A frequency domain approach can be used to calculate surface location error (SLE) in the same domain [18]. The simulation results can be presented as contours of constant SLE values. Calculation of stability lobes and SLE require the same inputs such as tool point frequency response function, force model coefficients (dependent on tool and workpiece material), tool geometry (number of teeth, tool diameter and helix angle) and cutting parameters (radial depth of cut and feed per tooth). The super diagram combines stability and SLE information in a single user-friendly format using a grey scale approach [46]. The acceptable SLE threshold of SLE is user defined.

To construct a super diagram, the user selects an axial depth of cut - spindle speed domain, radial depth of cut, feed per tooth and limiting SLE value. The axial depth of cut – spindle speed domain is discretized into a grid of points. The grid points are then penalized according to whether they are stable or unstable and if they fall within acceptable SLE limit or not. The penalties are given as follows

- No penalty is given to the points that are stable and within the user defined acceptable value of SLE; they are set as zero. The feasible zone is denoted by white on the super diagram.
- The points that are stable, but fall outside the acceptable SLE value are penalized by 1 (value -1). The SLE limited points are shown by a grey zone on the super diagram.

- The points that are unstable are penalized by 2 (value -2). The unstable points are represented by a black zone on the super diagram.

Effect of Tool Wear

The milling cutting force model used for stability and SLE calculations in this work is given by equations 3-1 and 3-2

$$F_t = k_t bh + k_{te} b \quad (3\text{-}1)$$

$$F_n = k_n bh + k_{ne} b \quad (3\text{-}2)$$

where, F_t is the tangential force component, k_t is the tangential cutting force coefficient, b is the axial depth of cut, h is the instantaneous chip thickness (which depends on the feed per tooth and cutter angle), k_{te} is the tangential edge coefficient, F_n is the normal force component, k_n is the normal cutting force coefficient, and k_{ne} is the normal edge coefficient [45]. The cutting force coefficients are used to calculate stability and SLE. It has been shown that the cutting force coefficients increase with tool wear [33-34]. The cutting force model in equations 3-1 and 3-2 can be experimentally evaluated as a function of tool wear. Thus, by correlating the force model coefficients to wear status of the tool, tool wear effects can be included in the super diagram.

Incorporating Uncertainty

The super diagram provides information about stable/unstable cutting conditions and acceptable/unacceptable SLE values in a binary format. Both the stability and SLE predictions are deterministic. However, there is inherent uncertainty in their actual locations due to the distributions in the inputs that are used to calculate them. These input uncertainties are incorporated in the super diagram as a user defined safety limit, which is applied to the feasible (white) zone boundary. The user selects how close he/she is willing to operate to the feasible boundary. An additional penalty is then

applied to the points that fall within the zone that violates the safety margin of the user. A light dark grey level is incorporated to indicate points that violate the safety margin. Dark gray now indicates the stable points where the SLE limit is exceeded, while light gray represents the previously feasible points which violate the safety margin.

Numerical Case Study

To demonstrate the incorporation of tool wear and uncertainty into the super diagram, a numerical case study is presented here. The parameters used for the case study are listed in Table 3-1

Table 3-1. Numerical case study parameters

Parameter	Value	Units
Stiffness	5×10^6	N/m
Damping ratio	0.05	
Natural frequency	300	Hz
Tool diameter	19.05	mm
Helix angle	0	degrees
Number of teeth	1	
Tangential cutting coefficient	2×10^9	N/m^2
Normal cutting coefficient	0.667×10^9	N/m^2
Feed per tooth	0.06	mm/tooth
Radial depth of cut	4.725	mm

Figure 3-1 shows the FRF of the dynamic system calculated from the data in Table 3-1. It is a single degree of freedom system with natural frequency at 300 Hz. The x and y direction dynamics were taken to be equal. Figure 3-2 shows the super diagram for the parameters listed in Table 3-1. The acceptable SLE limit was taken as 50 μm. The super diagram was constructed using the frequency domain approaches of calculating stability and SLE. The feasible values of axial depth of cut and spindle speed are selected to be in the range 0 mm to 20 mm and 2000 rpm to 10000 rpm, respectively.

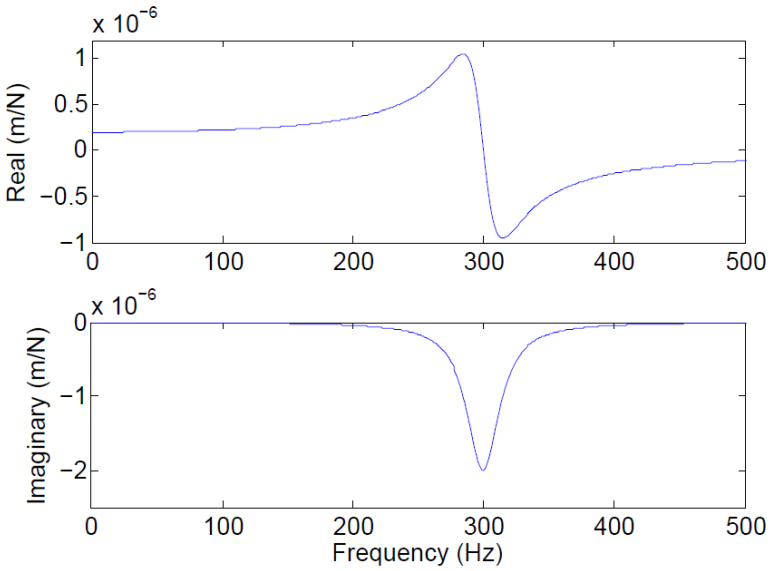


Figure 3-1. FRF for the system dynamics parameters used for numerical study

As shown, three zones can be identified as follows

- White feasible zone – stable and within acceptable SLE limit.
- Grey zone – stable points higher than acceptable SLE limit.
- Black unstable zone – unstable points

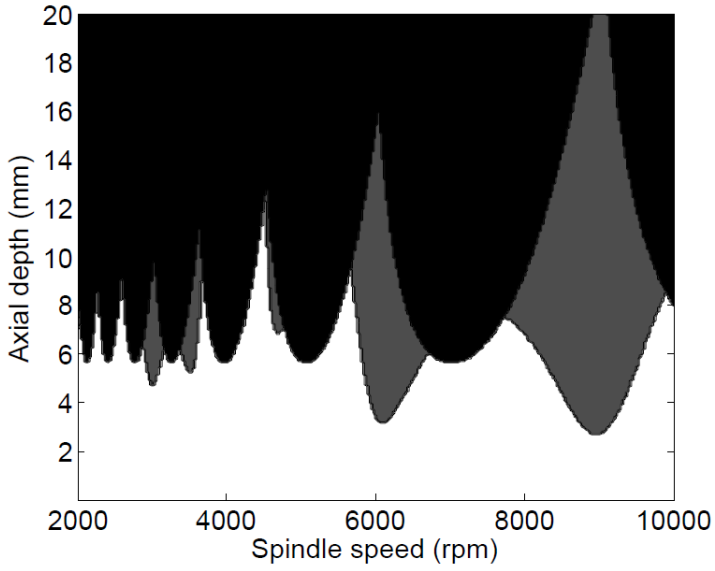


Figure 3-2. Super diagram with 50 μm stability limit.

Flank wear is a common wear feature. Tool life is often based on the cutting time required for the flank wear width (FWW) to reach a pre-determined level. For the

numerical study, a linear relationship of FWW with volume of material removed was assumed as shown in figure 3-3.

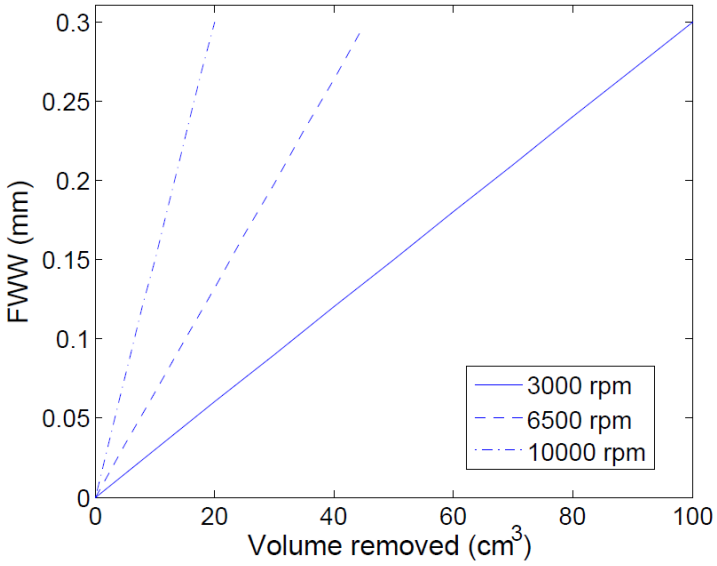


Figure 3-3. Variation in FWW assumed with volume removed at different speeds.

The FWW tends to increase with cutting time or volume removed and is spindle speed, Ω, dependent. The end of tool life was defined as the time required to reach a FWW of 0.3 mm.

As noted, the cutting forces and force coefficients tend to grow with FWW. The increase in cutting coefficients shown in equation 3-1, with volume removed, V, was assumed to be linear as shown in figures 3-4 and 3-5. The assumed linear relationship between the coefficients K_t and K_n with volume removed at different spindle speeds is provided in equations 3-2 and 3-3, where the intercepts, $c_{0,t}$ and $c_{0,n}$, are the coefficients for a new/unused tool (Table 3-1), and $c_{1,t}$ and $c_{1,n}$ are the speed-dependent rates of increase in the force coefficients with V.

$$K_t(\Omega, V) = c_{0,t} + c_{1,t} V \quad (3\text{-}3)$$

$$K_n(\Omega, V) = c_{0,n} + c_{1,n} V \quad (3\text{-}4)$$

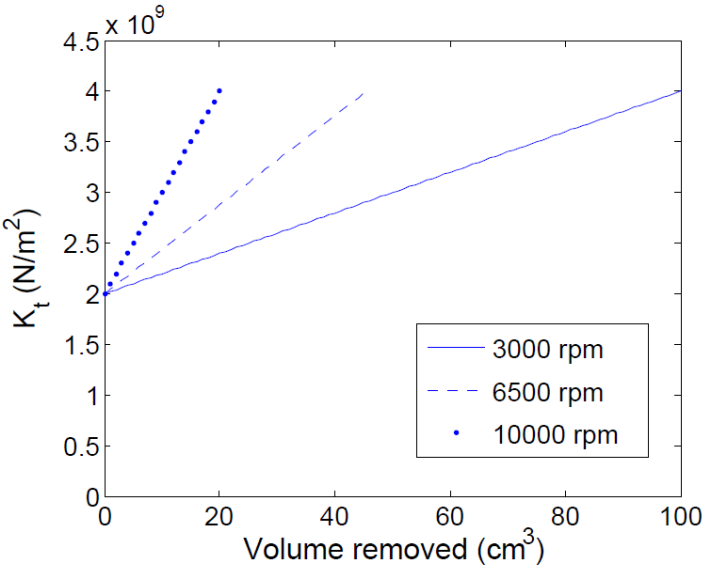


Figure 3-4. Variation in K_t assumed with volume removed at different speeds

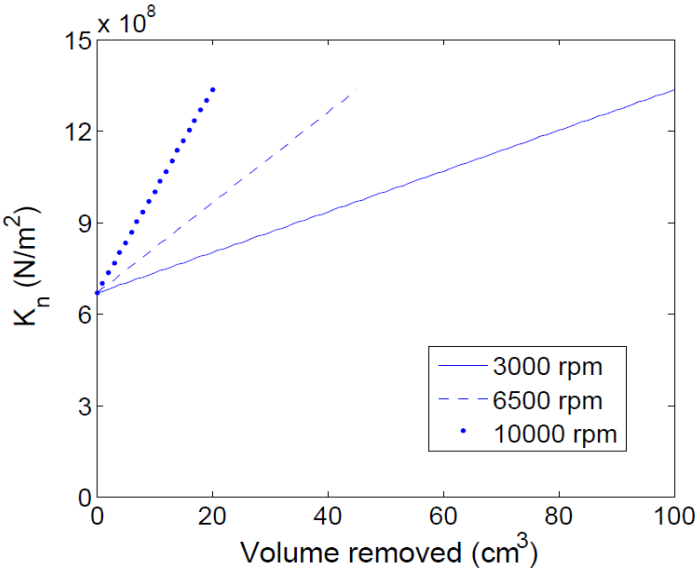


Figure 3-5. Variation in K_n assumed with volume removed at different speeds

The slopes ($c_{1,t}$ and $c_{1,n}$) were assumed to increase linearly with Ω between 2000 rpm and 10000 rpm such that the coefficients doubled at 10000 rpm for V = 20 cm^3 (where FWW = 0.3 mm; see Fig. 3-2) with no change at 2000 rpm for the same V. The assumption is based on the fact that FWW will be minimal at low speeds to remove 20

cm^3 and hence the force coefficients can be assumed to be constant. The edge coefficients are neglected in this numerical example without the loss of generality.

To illustrate, at V = 20 cm^3 for Ω = 3000 rpm, the slope $c_{1,t}$ can be calculated by linear interpolation as follows

$$c_{1,t} = \frac{c_{0,t}}{V} \times \frac{3000 - 2000}{10000 - 2000} = 1.25 \times 10^7 \text{ N/m}^2/\text{cm}^3 \quad (3\text{-}5)$$

Given this relationship between cutting coefficients, Ω and V, the effect of tool wear can now be incorporated into the super diagram. The volume to be removed must first be selected by the user. Then, new increased coefficients are calculated for each spindle speed as shown in equation 3-5. These coefficients are then used to determine the stability limit at each spindle speed. Also, SLE is calculated at each axial depth grid point for the given spindle speed using the increased value of the coefficients at that speed. The new super diagram for V = 20 cm^3 is shown if figure 3-6.

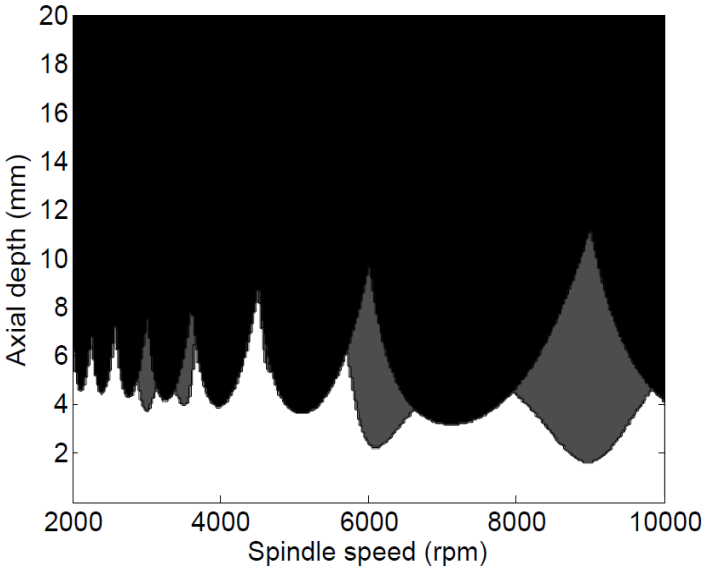


Figure 3-6. Super diagram including tool wear effects (V = 20 cm^3).

The new super diagram takes into account the effect of tool wear by applying the higher coefficient value. Since the cutting coefficients grow with Ω for a given volume to be removed, the stability limit decreases and the SLE infeasible zone grows while moving from left to right in the diagram.

Next, the super diagram can be modified to incorporate the user's uncertainty beliefs regarding the actual location of the deterministic boundaries. To carry out this task, the user defines safety limits for spindle speed, $\Delta\,\Omega$ and axial depth of cut, Δ b. These values give the distances from the boundaries that represent his/her 95% confidence level for actual feasible performance. For each feasible point in the { b, Ω } domain defined by the white zone, the penalty value of the surrounding eight points at distances $\Delta\,\Omega$ and Δ b from the test point are queried (see figure. 3-7). If any of these points are infeasible (with a penalty of -1 or -2), then the test point is penalized and also identified as infeasible as shown in figure 3-7.

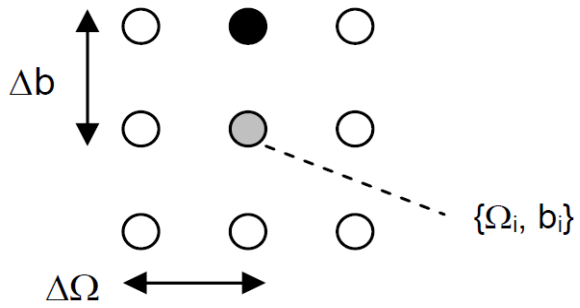


Figure 3-7. The safety limit is identified by testing the feasibility of the eight grid points surrounding {b_i, Ω_i}. In this case, the test point is penalized (-1) because the (black) point above it is unstable.

A new gray-scale is then implemented where the point values are: feasible (0, white), safety margin (-1, light gray), SLE limit (-2, dark gray), and unstable (-3, black).

Thus, the white feasible zone is further reduced on the application of safety margins by the user. Figure 3-8 shows a super diagram with uncertainties included as $\Delta\,\Omega$ = 100 rpm and Δ b = 0.5 mm.

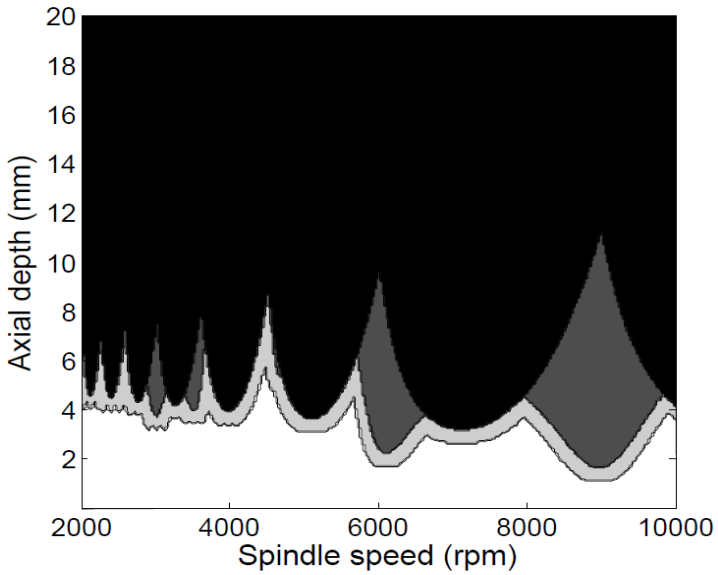


Figure 3-8. Super diagram including tool wear effects and the user-defined safety margin (V =20 cm^3, $\Delta\,\Omega$ = 100 rpm, and Δ b = 0.5 mm)

CHAPTER 4
TOOL WEAR EXPERIMENTS

Experimental Setup

The setup for the tool wear experiments is described in this section. A 19 mm (3/4”) diameter, single insert end mill (Kennametal KICR075SD260) was used for the tool wear experiments. An uncoated square carbide insert was used to achieve accelerated wear and reduce testing time. (Kennametal 107888126 C9 JC carbide insert; zero rake and helix angles, 15 deg relief angle, 9.53 mm square x 3.18 mm). The workpiece was 1018 steel (152.4 x 101.6 x 38.1 mm) with a Rockwell C hardness of 55. An atomic force microscope (AFM) was used to measure the topography of the carbide inserts. Figure 4-1 shows an example 50 µm x 50 µm measurement (256 line scans, no digital filtering) of the rake face. It is seen that there is a small chamfer with a 167 deg angle at the cutting edge. The roughness average, Ra, for the rake face was 310 nm.

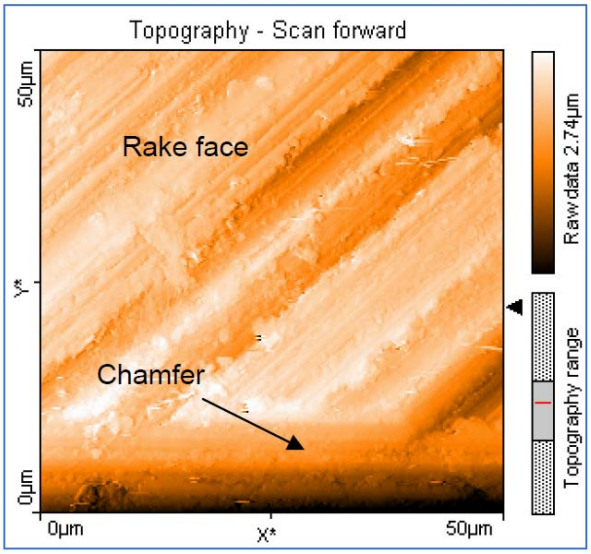


Figure 4-1. AFM measurement of carbide insert rake face

Figure 4-2 shows the experimental set-up. The workpiece was attached to a table-mounted cutting force dynamometer (Kistler 9257B).

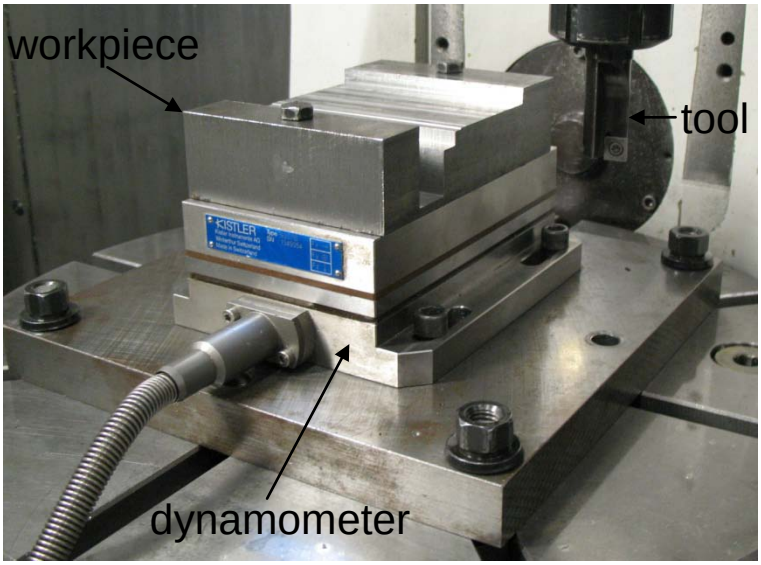


Figure 4-2. Setup for tool wear experiments.

Parameter Selection

The parameters for the first test were selected using the tool life data for machining 1018 steel with uncoated carbide inserts reported in [34] and [47].The recommended speeds and feeds for machining 1018 steel with an uncoated carbide insert for a tool life of 45 minutes were Ω = 2368 rpm and f_t = 0.177 mm/tooth for a three-tooth, 20° helix angle tool with 25.4 mm diameter, an axial depth of 5.08 mm and a radial depth of cut, a, of 25.4 mm (100% RI) [34]. As reported in [47], a tool life of 42 minutes was obtained by Ω = 2500 rpm and f_t = 0.06 mm/tooth using a single flute, 30° helix angle tool with 12.7 mm diameter, an axial depth of 5.08 mm and a 50% radial immersion. Considering these two data sets, the parameters for the initial test was selected as Ω = 2500 rpm, b = 3 mm, a = 4.7625 mm (25% RI) and f_t = 0.06 mm/tooth. Based on the selected parameters, the material removal rate (MRR) is given by equation 4-1.

$$MRR = abf_t \Omega N_t \quad = 2143.125 \text{ mm}^3/\text{min} \qquad (4\text{-}1)$$

where N_t is the number of teeth ($N_t = 1$). Figure 4-3 shows details of the test block. A slot to a depth of 35 mm was made to facilitate partial radial immersion cutting. The effective volume of the block to be removed was equal to 80.96 mm x 101.6 mm x 33 mm = 27450 mm^3. The time required to machine one block, based on the parameters selected, is equal to 126.6 minutes.

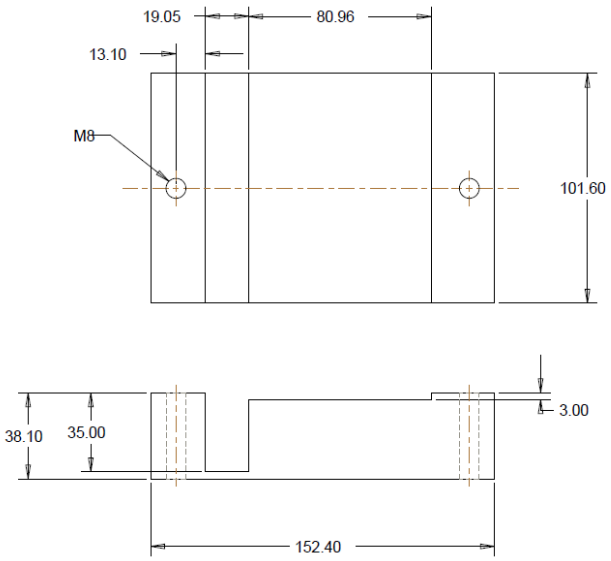


Figure 4-3. Details of test block. Note the effective length that can be removed is equal to 80.96 mm. The figure shows one 3 mm layer removed from the block. A total of 17 passes can be made per layer using a = 4.7625 mm (25% RI).

Testing Procedure

Figure 4-4 shows the tool path for machining the workpiece. The block was machined at the test parameters, while intermittently measuring forces at varying f_t values. As seen from figure 4-4, the feed was in the x direction. The tool retracted after making a cut, returned to the x starting location, and then stepped over in the y direction by an amount equal to the radial immersion in preparation for the next cut.

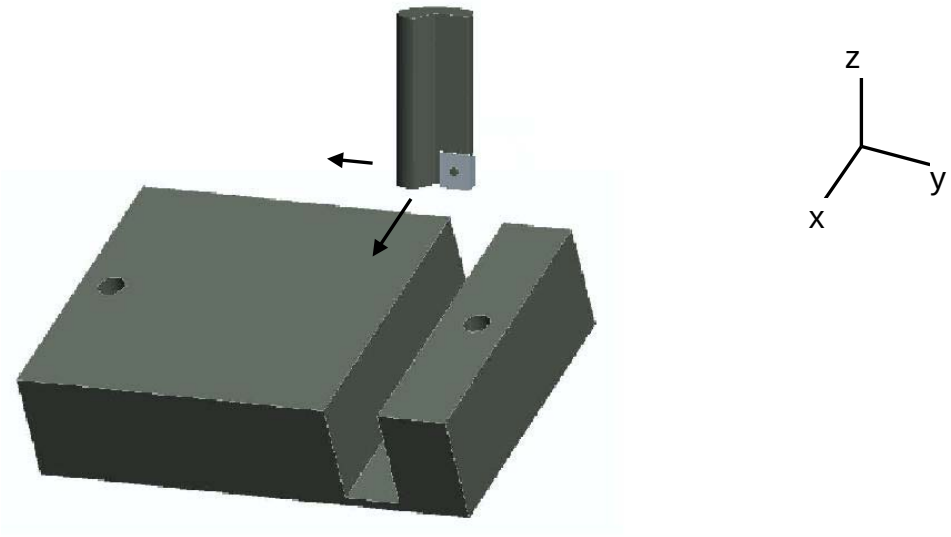


Figure 4-4. The tool path for machining workpiece. The tool was fed in the x- direction while the radial depth of cut was commanded using the y-direction. The tool retracted after cutting, returned to the starting x location, and was then incremented in the y- direction by the radial depth of cut. All the cuts were downmilling.

Stability for Cutting Tests

The stability lobes for the given spindle-holder-tool combination were calculated to ensure the wear tests are stable. The stability lobes were calculated by using the method described in [35]. The cutting force coefficients for the tool-workpiece material combination and the frequency response function of the system were measured and used as inputs for the stability lobe diagram development.

Calculation of Force Coefficients

The cutting force coefficients for the given tool-workpiece material combination were calculated using a linear regression on the mean x (feed) and y direction forces. The force was measured during cuts at 2500 rpm, 3.0 mm axial depth of cut and 25% RI over a range of feed per tooth values (0.04, 0.05, 0.06 and 0.07 mm/tooth). The slopes, a1x and a1y, and the intercepts, a0x and a0y, for the x and y direction data,

respectively, were used to calculate the cutting force coefficient given by equations 4-2 to 4-5

$$K_t = \frac{8\pi(a1y.(2\phi_e - 2\phi_s + \sin(2\phi_s) - \sin(2\phi_e)) + a1x.(\cos(2\phi_s) - \cos(2\phi_e)))}{((Ntb).((2\phi_e - 2\phi_s + \sin(2\phi_s) - \sin(2\phi_e))^2 + (\cos(2\phi_s) - \cos(2\phi_e))^2))} \quad (4\text{-}2)$$

$$K_n = \frac{8\pi(a1x.(2\phi_e - 2\phi_s + \sin(2\phi_s) - \sin(2\phi_e)) + a1y.(\cos(2\phi_s) - \cos(2\phi_e)))}{((Ntb).((2\phi_e - 2\phi_s + \sin(2\phi_s) - \sin(2\phi_e))^2 + (\cos(2\phi_s) - \cos(2\phi_e))^2))} \quad (4\text{-}3)$$

$$K_{te} = \frac{\pi(a0x.(\sin(\phi_e) - \sin(\phi_s)) - a0y.(\cos(\phi_e) - \cos(\phi_s)))}{(Ntb).(1 - \cos(\phi_e - \phi_s))} \quad (4\text{-}4)$$

$$K_{ne} = \frac{\pi(a0x.(\cos(\phi_e) - \cos(\phi_s)) + a0y.(\sin(\phi_e) - \sin(\phi_s)))}{(Ntb).(1 - \cos(\phi_e - \phi_s))} \quad (4\text{-}5)$$

where Φ_s is the start or entry angle and Φ_e is the exit angle. The force coefficients were calculated as K_t = 2.1899$\times 10^9$ N/m^2, K_n = 1.7327$\times 10^9$ N/m^2, K_{te} = 54474 N/m , K_{ne} = 64471 N/m for a new insert. Figure 4-5 shows the linear regression of mean x and y forces used to calculate the cutting force coefficients [46,50].

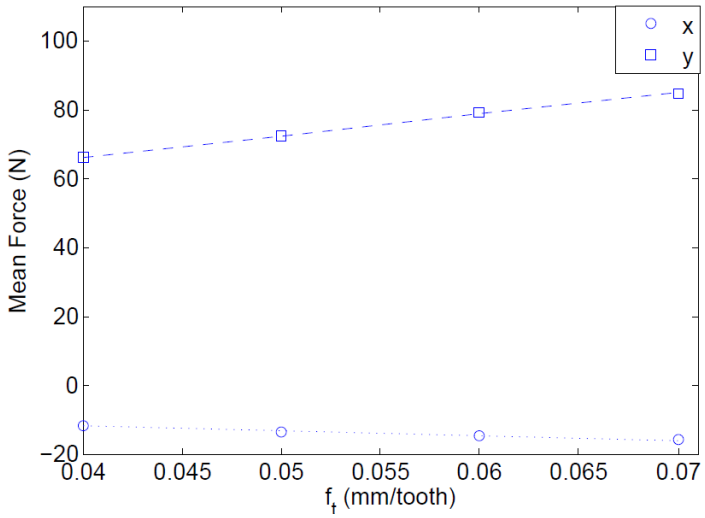


Figure 4-5. Linear regression of mean x and y direction forces.

Calculation of Stability Lobes

The tool point FRF was measured by impact testing and the stability lobes were calculated for the system to ensure stable cutting tests. Figure 4-6 shows the x and y direction FRF of the tool-spindle-holder. Figure 4-7 shows the corresponding stability lobes for the dynamic system with 25% RI.

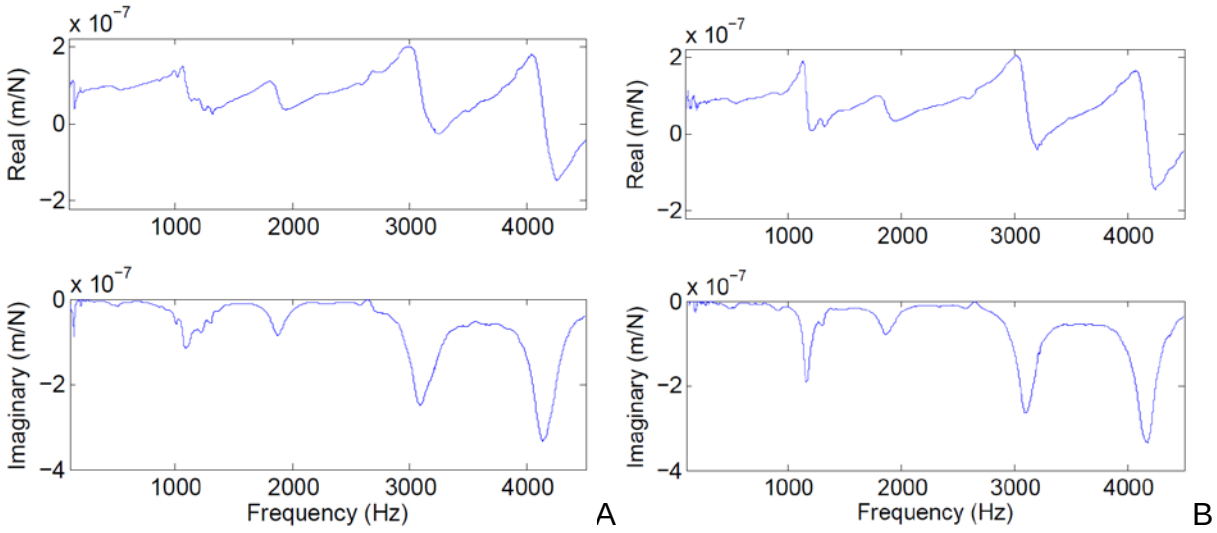


Figure 4-6. Tool point FRF of the system: A) x direction B) y direction.

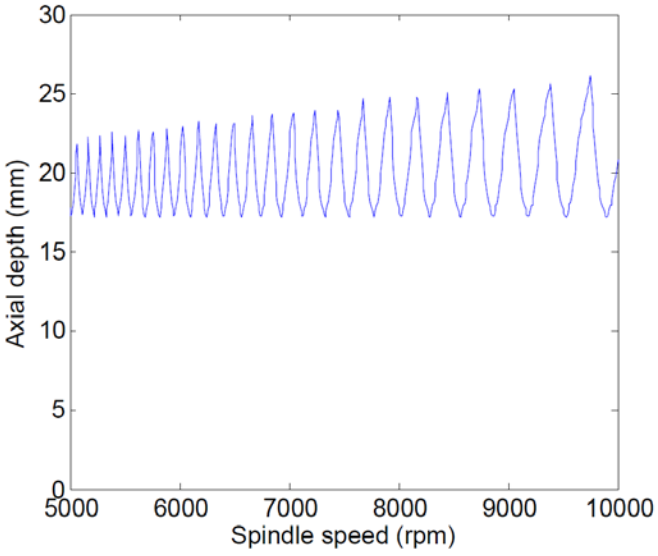


Figure 4-7. Stability lobe diagram for the system used for tool wear experiments.

As seen from figure 4-4, the critical value of axial depth of cut is 17 mm. The axial depth of cut selected for wear tests was decided to be 3 mm. Thus, the wear tests for the selected parameters are stable at all cutting conditions.

Wear Test Results

The first tool wear test was done at the parameters listed previously and was repeated three times. The cutting force coefficient regression was completed intermittently while wearing the tool. The feed per tooth values for the regression analysis were {0.03, 0.04, 0.05, 0.06 and 0.07 mm/tooth}. In addition to monitoring the cutting force, the insert wear status was also measured at the same intervals as the cutting force coefficients. To avoid removing the insert/tool from the spindle, a handheld microscope (60x magnification) was used to record the rake and flank surfaces. The calibrated digital images were used to identify the FWW. No crater wear was observed. Example FWW results for 2500 rpm tests are provided in figure 4-8 (1 σ error bars), where the interval between measurements was 12 cm^3. The test was terminated when the maximum FWW reached 0.7 mm.

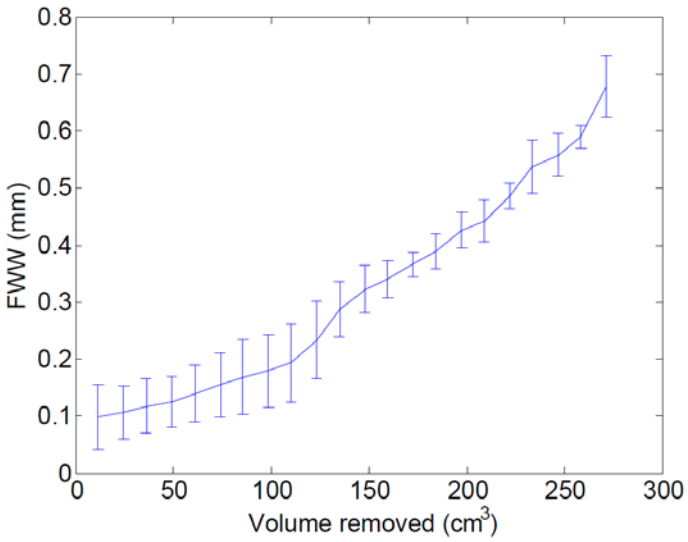


Figure 4-8. Variation in FWW with volume removed (Ω = 2500 rpm).

Microscope images of the relief face for selected volumes of material removed are displayed in figure 4-9. The maximum FWW increases with the volume removed, V, as expected.

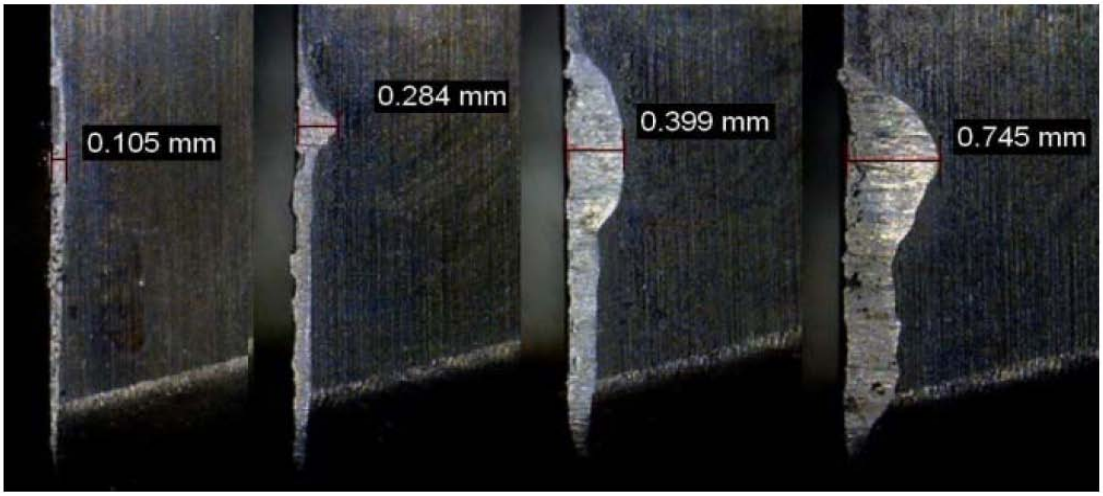


Figure 4-9. Images of FWW at 60x magnification (from left to right, V = {50, 125, 200, and 275} cm^3).

The force coefficients were also calculated at each interval. Figures 4-10 and 4-11 shows the results.

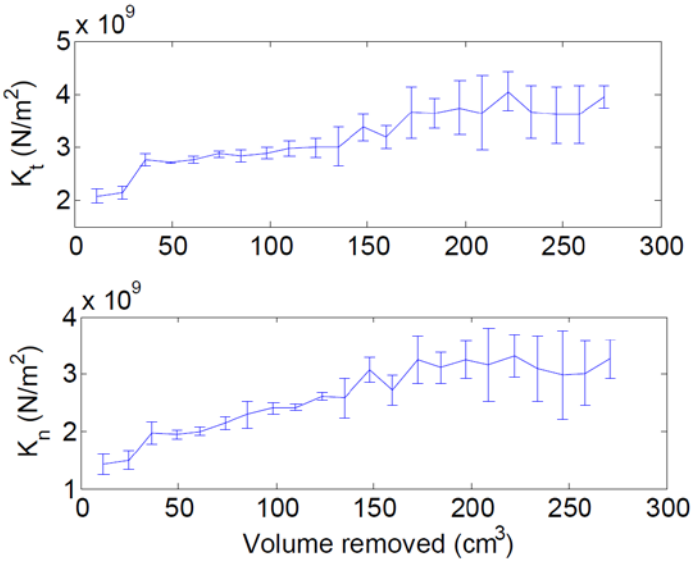


Figure 4-10. Variation in K_t and K_n with volume removed (Ω = 2500 rpm).

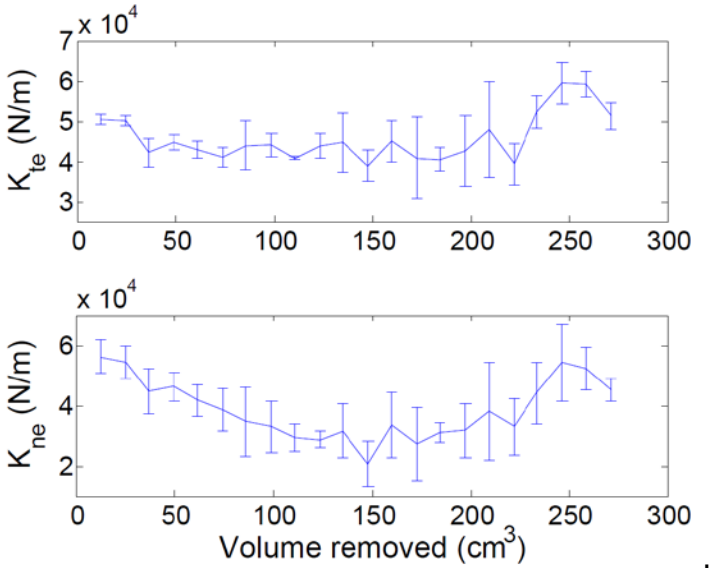


Figure 4-11. Variation in K_{te} and K_{ne} with volume removed (Ω = 2500 rpm)

For the selected tool/material combination, K_t and K_n show an approximately linear increase, while K_{te} and K_{ne} show no clear trend. A linear fit provides a good approximation of the increase in K_t and K_n (R^2 value is 0.877 and 0.853 for K_t and K_n, respectively). As a next step, the change in force coefficient behavior with spindle speed was evaluated by performing additional tests at {3750, 5000, 6250 and 7500} rpm. Figure 4-12 shows the results and the linear least square fits to the results. The procedure was the same as described for the 2500 rpm testing. All parameters were unchanged, except for spindle speed.

The K_{te} and K_{ne} values again did not exhibit any significant trend at the additional spindle speeds. As expected, the rates of K_t and K_n growth (i.e., the slopes) increased with spindle speed. Interestingly, when plotted versus the corresponding FWW (measured with the handheld digital microscope), the five different spindle speed results collapse onto a single line as seen from figure 4-13. Thus, the increase in K_t and K_n is dependent on FWW and independent of spindle speed. This suggests that if the FWW

were monitored, it could provide an in-process approach to updating the force model coefficients based on the tool wear status.

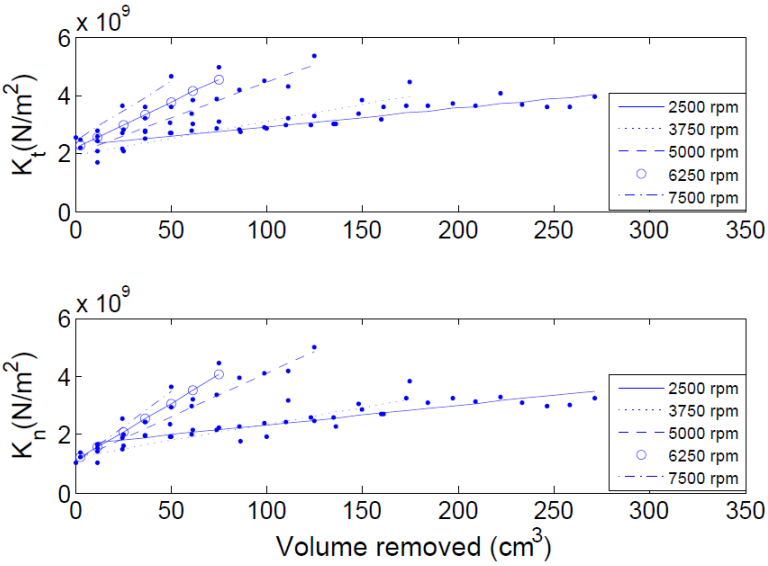


Figure 4-12. Variation in K_t and K_n with volume removed for various spindle speeds.

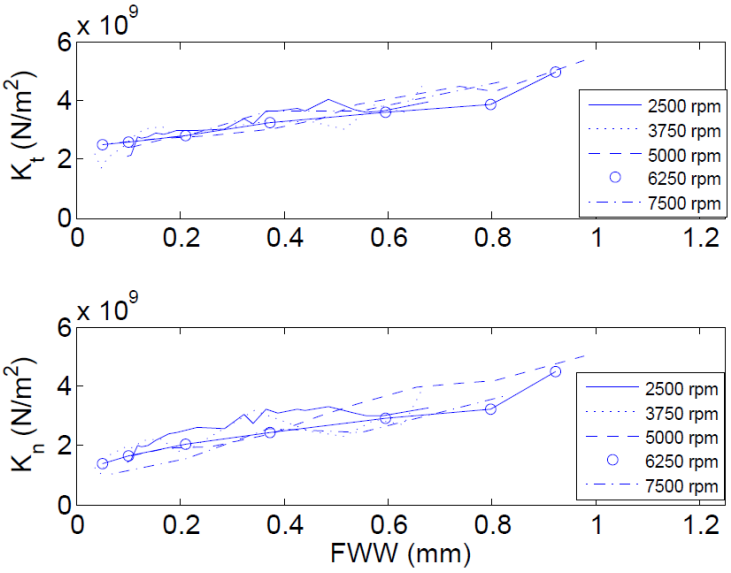


Figure 4-13. Variation in K_t and K_n with FWW at various spindle speeds.

To incorporate the variation in K_t and K_n with FWW at various spindle speeds, the slopes of the individual (K_t and K_n versus volume removed) lines in figure 4-12 are plotted against spindle speed. As seen from figure 4-14, the slopes increase linearly

with spindle speed as was assumed in the numerical case study provided in Chapter 3. The error bars on slopes were obtained by performing a Monte-Carlo simulation on the range of K_t and K_n values and calculating slopes for each random combination.

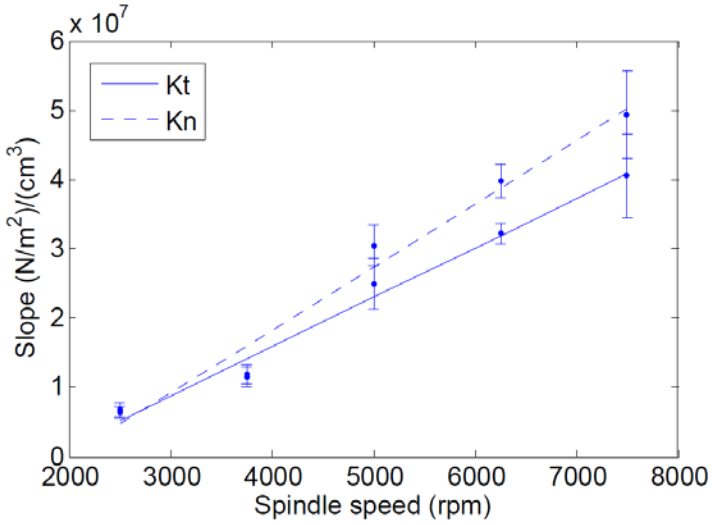


Figure 4-14. Variation in slope with spindle speed for the K_t and K_n versus volume removed lines (from figure 4-13).

Therefore, equations 3-3 and 3-4 which describe a linear relationship between K_t and K_n and the volume removed are applicable to this tool-material pair. The slope of the lines in figure 4-14 are 7.1×10^3 $(N/m^2/cm^3)$/rpm and 9.1×10^3 $(N/m^2/cm^3)$/rpm and the intercepts are -1.3×10^7 $N/m^2/cm^3$ and -1.8×10^7 $N/m^2/cm^3$ for the K_t and K_n data, respectively. The negative intercept values are attributed to the linear fit with inherent experimental uncertainty. The terms $c_{1,n}$ and $c_{1,n}$ defined in equations 3-3 and 3-4 can be calculated at different speeds by multiplying these slopes (from the lines in figure 4-14) by the corresponding spindle speed adding the intercept shown in equation 4.6 and 4.7. The intercepts in equations 3-3 and 3-4 are taken as the starting value (V = 0) of the force coefficients obtained using a new insert. Thus, equations 3-3 and 3-4 can be written for the tool-material combination as shown in equations 4-6 and 4-7.

$$K_t = 2.2\times10^9 + (7.1\times10^3.\Omega - 1.3\times10^7)V \quad (4\text{-}6)$$

$$K_n = 1.2\times10^9 + (9.1\times10^3.\Omega - 1.7\times10^7)V \quad (4\text{-}7)$$

As noted earlier, since the K_{te} and K_{ne} values did not show any significant trend, the mean values from figure 4-11 were applied i.e., $K_{te} = 4.6\times10^4$ N/m and $K_{ne} = 3.9\times10^4$ N/m.

Given this relationship, the super diagram that incorporates tool wear can then be developed at a user-selected volume by calculating K_t and K_n for each spindle speed and using this value to calculate the speed-dependent stability boundary and SLE as shown in the numerical example.

All the previous testing was performed at f_t = 0.06 mm/tooth. This enables a super diagram to be developed at that value. However, changing the f_t value will change the SLE values with all other parameters remaining the same. Therefore, a similar set of experiments were completed at feed per tooth values of {0.03, 0.045, 0.075, and 0.09} mm/tooth. The tests were completed at 5000 rpm and other parameters remained the same as before. Figure 4-15 shows the variation in cutting force coefficients at different feed per tooth values.

The wear rate is higher and the volume of material that can be removed is lower for the smaller feed per tooth values. The wear rate trend suggests that strain hardening may be in effect. The thinner chips with increased hardness can cause accelerated wear. The reduced amount of material that can be removed could also be attributed to the increase in cutting time and the number of passes through the material required to remove the same volume.

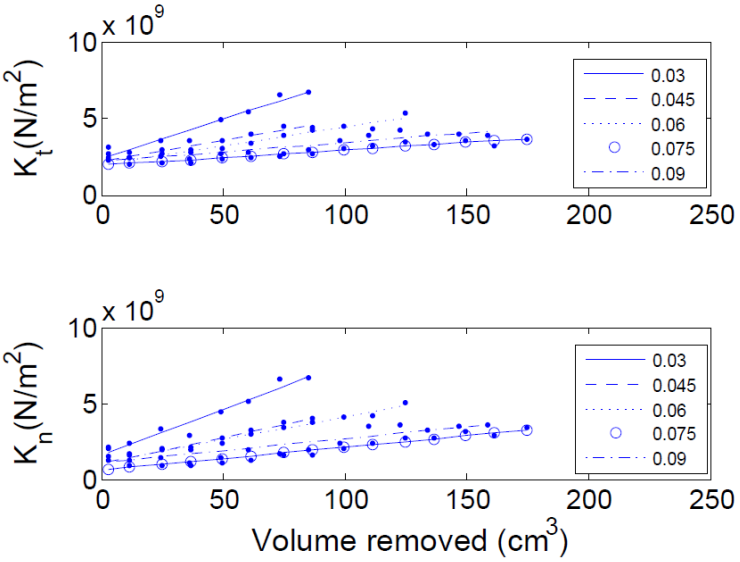


Figure 4-15. Variation in K_t and K_n with volume removed for various feed per tooth values.

Finally, the variation in wear rate behavior with axial depth of cut was evaluated. The axial depths were {3, 4.5 and 6} mm, the spindle speed was 5000 rpm, the feed per tooth was 0.06 mm/tooth and the radial depth remained at 4.7 mm. Figure 4-16 shows the results. Note that the K_t and K_n values are plotted against volume normalized by the axial depth of cut, $V_n = V/b$. This normalization was necessary because the independent variable, V, is a function of the dependent variable, b. As seen in the figure, the three tests sets collapse onto a single line for the usable tool life when plotted versus the normalized volume.

The agreement between b values demonstrates that testing at a single axial depth is sufficient. The divergence at the highest V_n value for b = 6 mm is due to excessive FWW for that test (> 0.7 mm). It has also been suggested that variation in FWW is not observed at different radial depths of cut and differing number of teeth (assuming no runout) [7]. Similar results can be expected for tests with varying radial depth of cut or the number of teeth if K_t and K_n are again plotted against normalized volume.

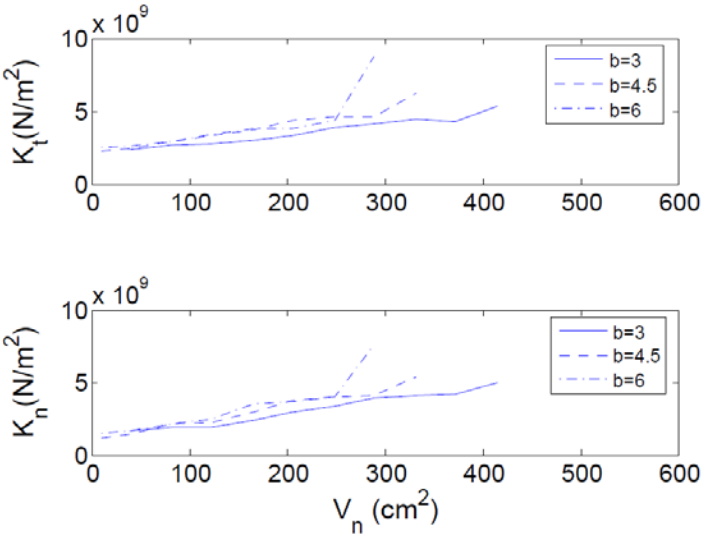


Figure 4-16. Variation in K_t and K_n with normalized volume removed for various axial depths of cut.

By normalizing the volume removed by the axial and radial depths of cut and number of teeth, the required number of tests can be dramatically reduced. For a given tool-workpiece combination, testing can be completed only at a selected axial depth of cut, radial depth of cut and number of teeth. The results can then be extended to other combinations by plotting the values of force coefficients against the normalized volume removed.

CHAPTER 5
STABILITY DIAGRAM VALIDATION

Experimental Setup

The tool wear experimental results showed a linear increase in cutting force coefficients K_t and K_n with volume removed due to progressive flank wear. This increase in force coefficients causes the stability limit to decrease. Results also showed a spindle speed dependence on the rate of increase in the force coefficients. For a preselected volume to be removed (based on the workpiece geometry, for example) , the rate of increase in K_t and K_n for the given material-tool combination is given by equation 4-2 and 4-3. The stability diagram can be generated using the volume-based force coefficients which are also spindle speed dependent.

The stability diagram validation experiments were completed using new and worn inserts to show that the stability limit decreases with tool wear. The validation tests were carried out using 1018 steel and the same inserts as for tool wear experiments. As seen in figures 4-6 and 4-7, the stiff tool-holder-spindle combination yielded a critical stability limit of 17 mm. For the stability experiments, a long collet-type holder was used in order to obtain a more flexible dynamic system and lower stability limit (see figure 5-1).

Figure 5-1. Long collet-type holder for stability tests.

The workpiece was again attached to the table-mounted dynamometer to measure cutting force during the tests. For this setup, the workpiece was assumed to be rigid and only the tool dynamics were considered for the stability diagram calculations. As a result, the changes in the mass of the part as the material was removed did not affect the dynamics. Figure 5-2 shows the tool point FRFs in the x and y directions, respectively.

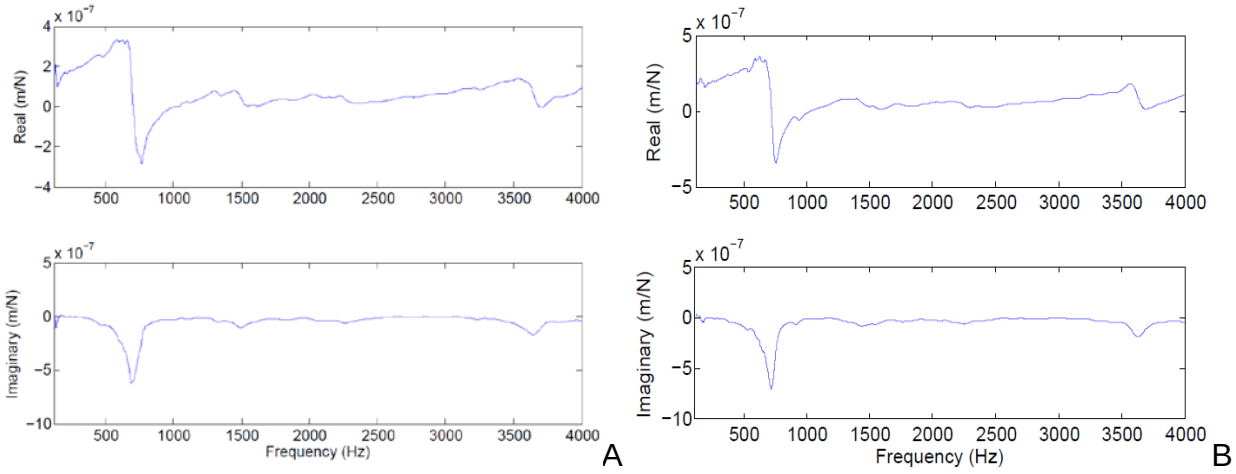


Figure 5-2. Tool point FRF of the system: A) x direction B) y direction.

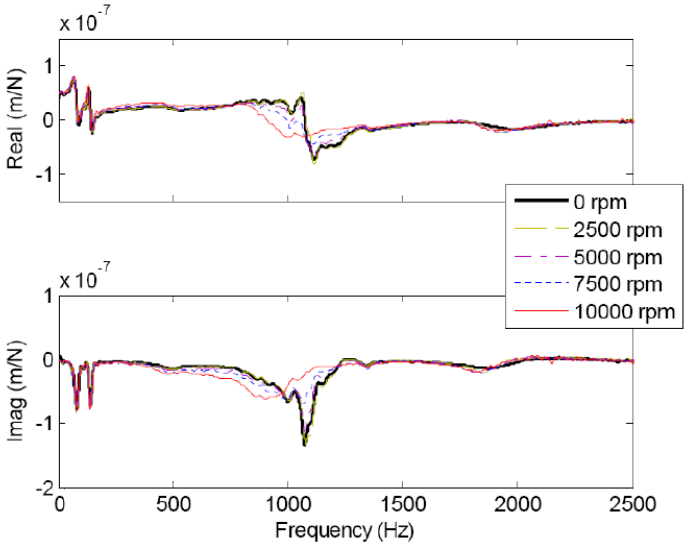


Figure 5-3. Change in spindle dynamics with spindle speed for the Mikron UCP Vario 600 used in this study as reported in [48]. Note that a different tool-holder combination was used so the natural frequencies differ.

However, the spindle dynamics change significantly with spindle speed for the particular machine used in this research (Mikron UCP Vario 600 with 20000 rpm, 14 kW Steptec spindle) [48]. As spindle speed increases, the natural frequency reduces and the dynamic stiffness increases as seen in figure 5-3. Therefore, for the calculation of stability lobes, a 50% increase in the dynamic stiffness of the system at 5000 rpm was assumed relative to the FRF recorded at zero speed.

Testing and Validation

A new and worn insert (0.5 mm FWW) were used for stability tests. The force coefficients for both the inserts were measured using the linear regression of the average x and y direction forces at varying f_t values. These force coefficients are listed in table 5-1.

Table 5-1. Force coefficients values for new and worn insert

	K_t (N/m^2)	K_n (N/m^2)	K_{te} (N/m)	K_{ne} (N/m)
New insert	1.90×10^9	0.78×10^9	45500	46650
Worn insert	4.98×10^9	4.51×10^9	45500	25500

The values listed in table 5-1 are in close agreement with the values expected at FWW = 0.5 mm from figure 4-13. Stability diagrams were separately generated for both inserts using their respective force coefficient values. Figure 5-4 shows stability diagrams for the new and worn inserts. The stability lobes were generated at a = 19.05 mm (100% RI).

Cutting tests were completed at 0.8 mm, 1.6 mm, 2.2 mm and 3 mm axial depth of cut using both inserts. The spindle speed was 5100 rpm. It can be safely assumed that there is no additional increase in the force coefficients during these tests due to small amount of material removed.

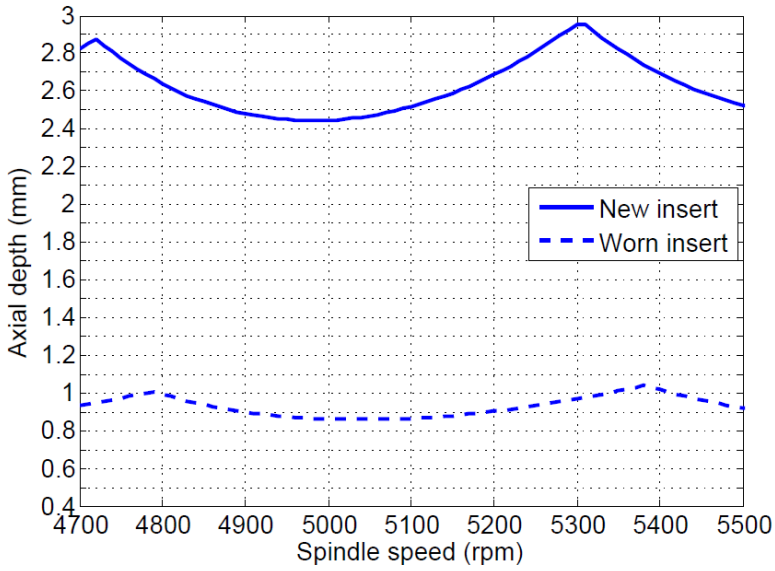


Figure 5-4. Stability lobes for new and worn inserts. Note the stability limit is reduced for the worn insert due to higher cutting force coefficients.

Figures 5-5 and 5-6 show the frequency spectrum of the force signal in the x and y directions for an axial depth of 1.6 mm with the new and worn inserts.

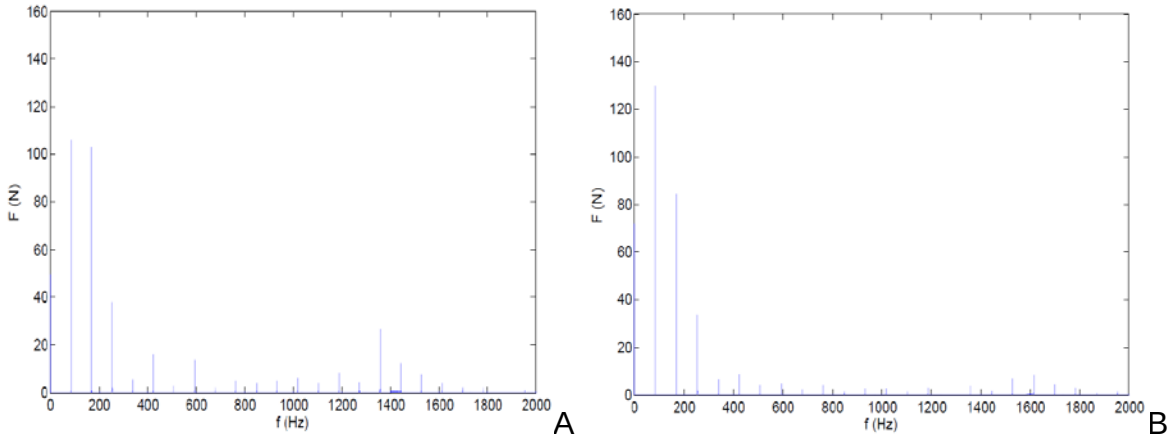


Figure 5-5. Force frequency spectrum for new insert at b = 1.6 mm: A) x direction B) y direction.

Figure 5-5 shows frequency content only at the tooth passing frequency (85 Hz) and its harmonics. The increased magnitude near 1400 Hz occurs due to the dynamometer's lowest natural frequency.

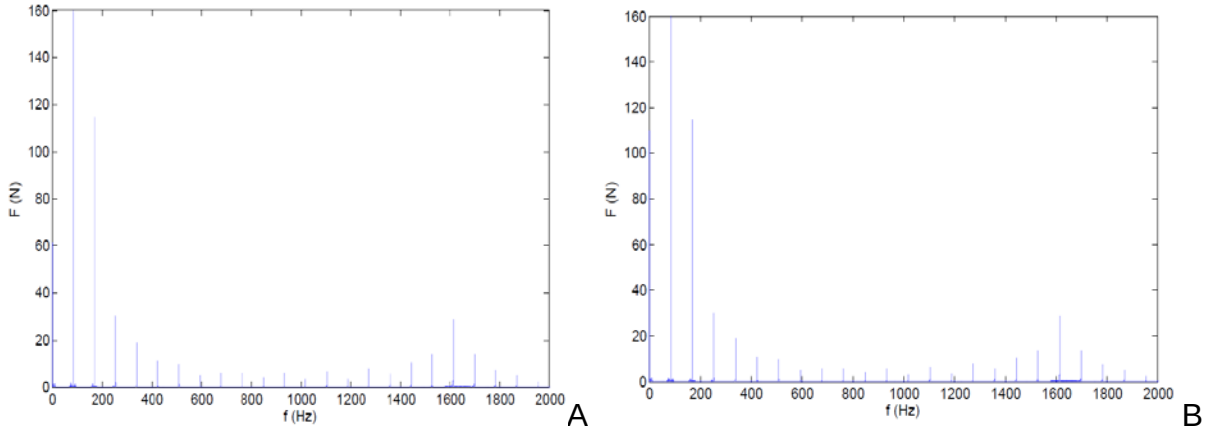


Figure 5-6. Force frequency spectrum for worn insert at b = 1.6 mm: A) x direction B) y direction.

Figure 5-6 shows increased magnitude at the tooth passing frequency and its harmonics relative to figure 5-5; this agrees with prior studies that demonstrate larger forces with tool wear [8,15]. Because the forced vibrations are large, using the force frequency content to identify chatter (by content at frequencies other than the once-per-revolution, tooth passing frequency, and corresponding harmonics) is challenging in this case. Therefore, a once-per-revolution force sampling strategy for the x (F_x) and y (F_y) directions was used to identify chatter. The once-per-revolution samples were obtained by sampling the force data at the commanded spindle rotating frequency [36]. The once-per-revolution samples for stable conditions (forced vibration only) are synchronous with spindle rotation and the behavior repeats each revolution. This produces a single small cluster of points in the plots. Unstable (chatter) behavior produces a more distributed set of points due to its asynchronous nature. In this case, it has been shown that the sampled tool displacement data collected during regenerative chatter is characterized by a ring of points which is characteristic of quasi-periodic (or asynchronous) motion [36-38]. Figure 5-7 shows the time domain simulation results for once-per-revolution sampling of tool displacements and forces (x and y directions) with

b = 1.6 mm and Ω = 5100 rpm for new insert [46, 49]. As seen from figure 5-7 the once-per-revolution samples are synchronous with spindle speed and produce a small cluster of points for both the displacement and the force plots.

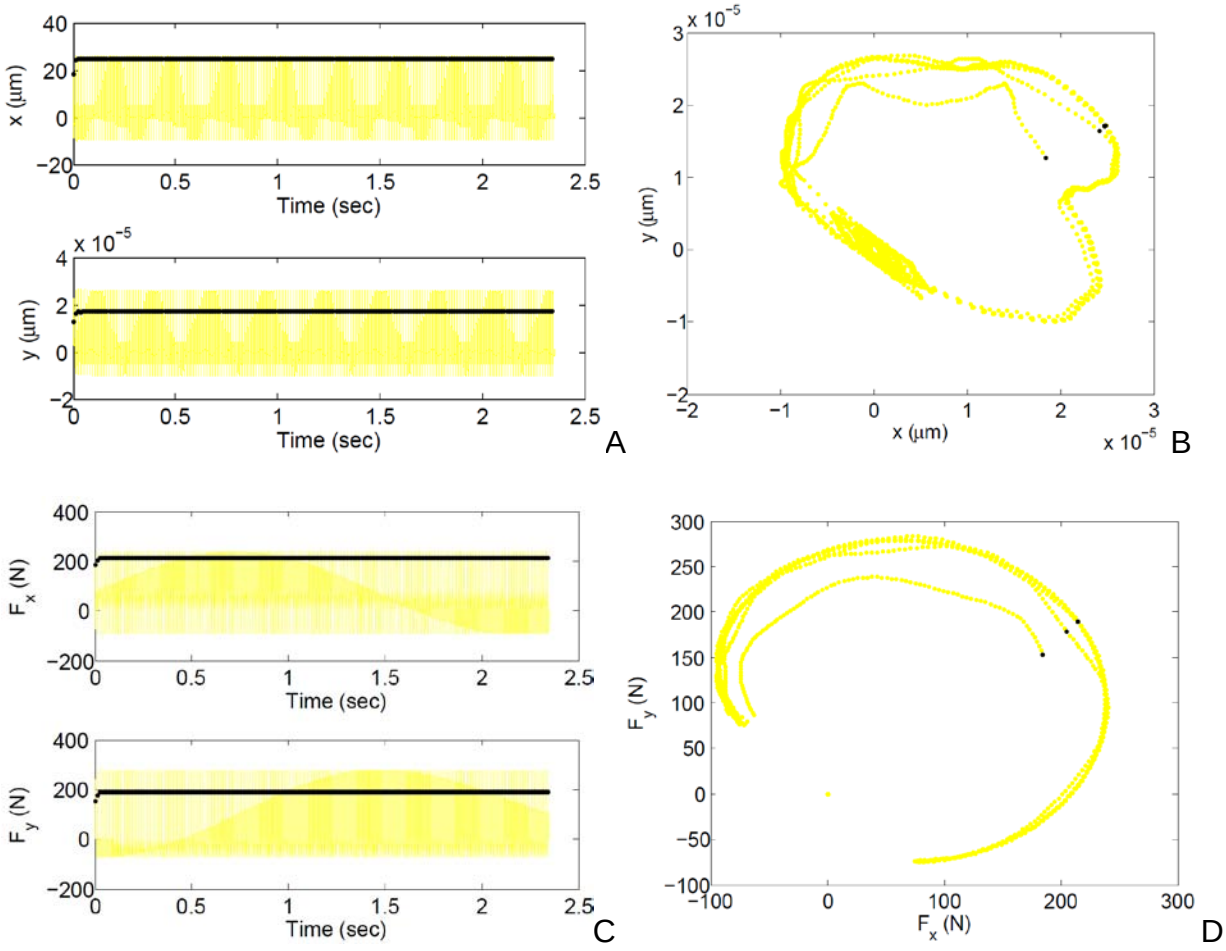


Figure 5-7. Once-per-revolution samples for b= 1.6 mm and Ω=5100rpm for new insert: A) time history of x and y displacements of the tool. The dark points represent the once-per-revolution samples. B) The x and y displacements are plotted against each other. The once-per-revolution samples are again shown as dark points. Similar results are observed for: C) time history of F_x and F_y and D) F_x vs F_y.

Figure 5-8 shows similar plots for an unstable cut at b = 1.6 mm and Ω = 5100 rpm when using the worn insert (new cutting force coefficients). Note that while the once-per-revolution plots for displacement show a distinctive ellipse, it appears as a line in the force plots.

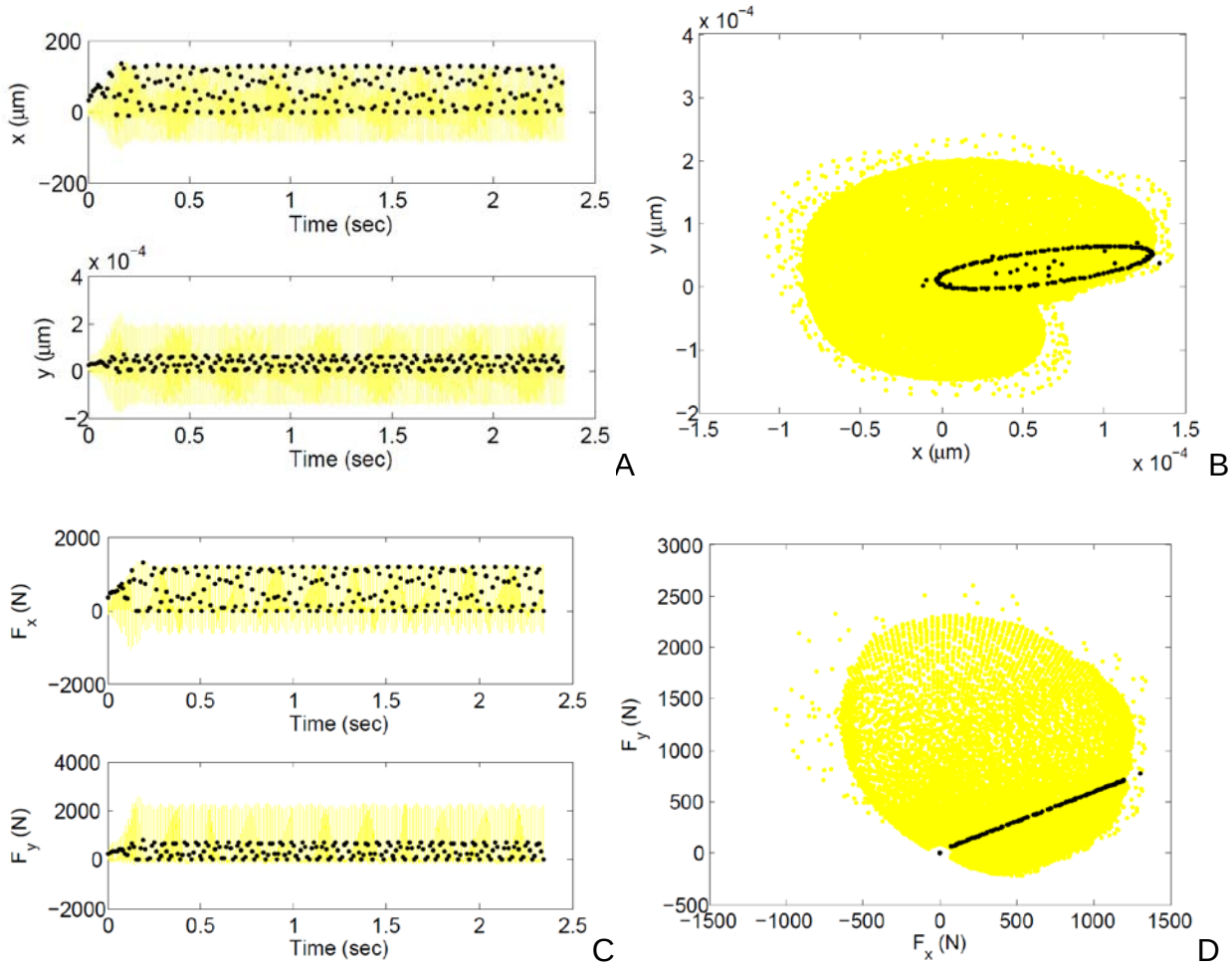


Figure 5-8. Once-per-revolution samples for b= 1.6 mm and Ω=5100 rpm using the worn insert: A) time history of x and y displacements of the tool. The once-per-revolution samples now vary considerably. B) The x and y displacements are plotted against each other. The once-per-revolution samples appear as an ellipse. C) Time history of F_x and F_y. D) F_x vs F_y. Note that the once-per-revolution samples collapse onto a straight line for the force data.

The force data was measured using the table-mounted dynamometer. Once-per-revolution sampling was limited to the steady state portion of the force to remove the influence of the cut entry and exit transients on the results. Figure 5-9 shows the once-per-revolution plots for new and worn insert tests.

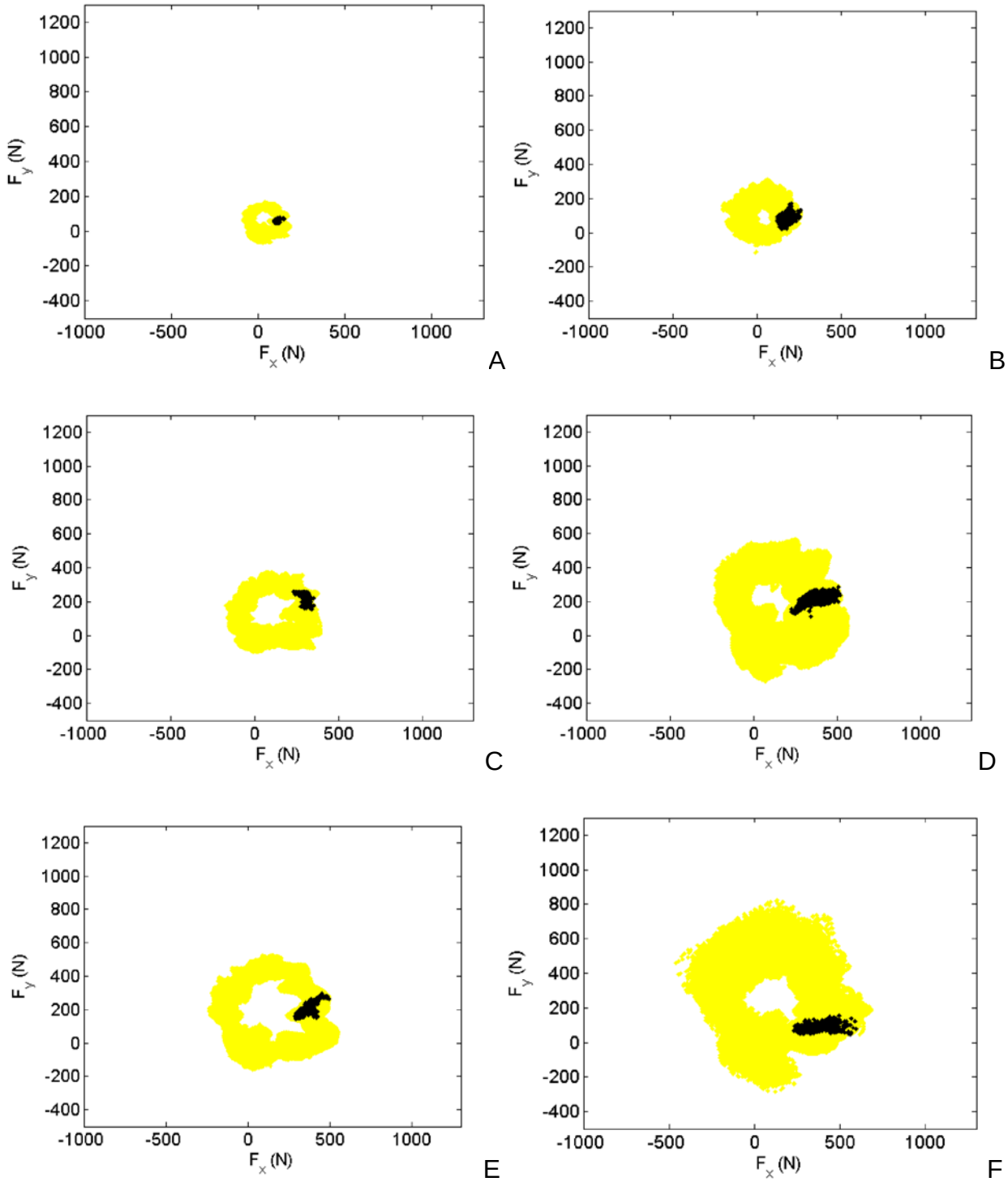


Figure 5-9. Once-per-revolution plots for new and worn inserts at 0.8 mm, 1.6 mm, 2.2 mm and 3.0 mm. A, C, E and G) New insert. B, D, F and H) Worn insert.

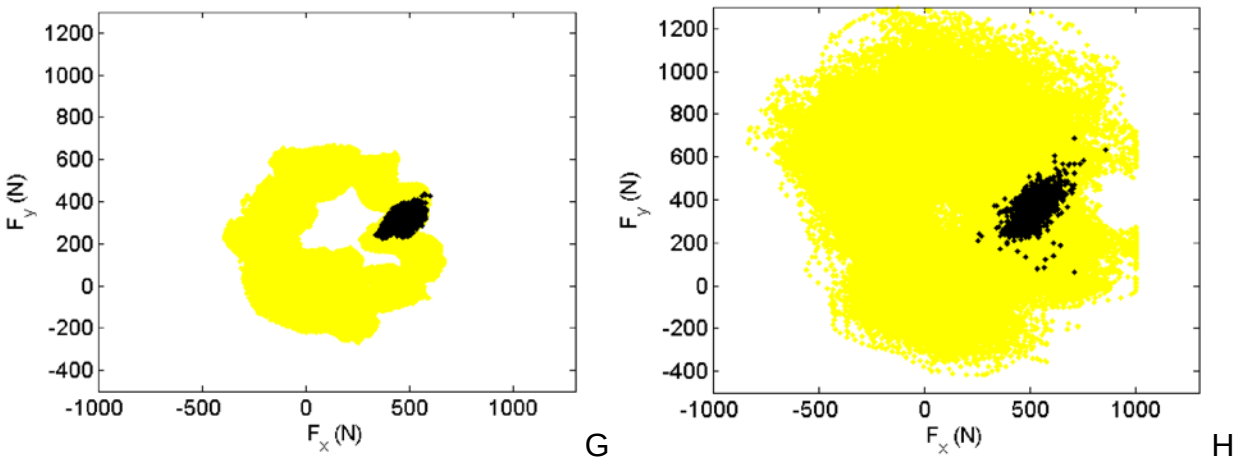


Figure 5-9 Continued.

As seen from figure 5-9, the once-per-revolution scatter shows a marked increase for tests at b = 2.2 mm and b = 3.0 mm for new insert and from b = 1.6 mm onwards for the worn insert; as described previously, this increased scatter indicates instability. These results confirm to the stability diagram for the new and worn inserts shown in figure 5-4.

Because the distribution of these points begins to widen when chatter occurs [36], the scatter in the sample distribution can be quantified using the variance (the square of the standard deviation) in the sampled points. The statistical variance in the once-per-revolution data can therefore be used identify chatter. The variance for unstable cuts is typically at least an order of magnitude larger than the variance for stable cuts [44]. The variance in the once-per-revolution sampled force data was normalized with the variance of the (unsampled) force data [48].

$$R = \frac{\sigma^2_{x,rev} + \sigma^2_{y,rev}}{\sigma^2_x + \sigma^2_y} \tag{5-1}$$

where R is the normalized variance, $\sigma^2_{x/y,rev}$ is the variance in the once-per-revolution sampled force data, and the x and y subscripts refer to the x and y directions.

Table 5-2 shows the R values for test cuts using the new and worn inserts. Figure 5-10 shows the increase in normalized variance with axial depth of cut for the new and worn inserts.

Table 5-2. Normalized variance for new and worn inserts.

	b (mm)	R
New Insert	0.8	0.0010
	1.6	0.0010
	2.2	0.0010
	3.0	0.0099
Worn Insert	0.8	0.0038
	1.6	0.0052
	2.2	0.0039
	3.0	0.0107

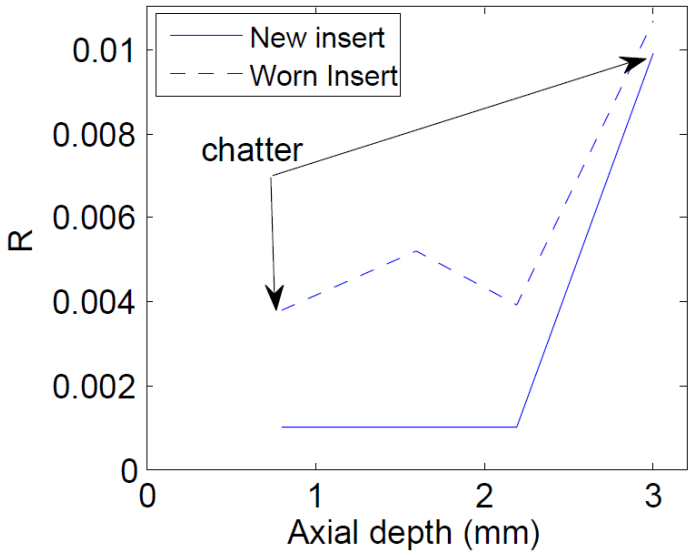


Figure 5-10. Normalized variance in the once-per-revolution samples with axial depth of cut for the new and worn inserts. The R values are listed in table 5-2.

The normalized variance values increases at b = 3.0 mm for new insert, which indicates chatter. The R values are larger for all instances when using the worn insert. The four to five times increase suggests that all the worn insert cuts were unstable, or marginally stable at best for the 0.8 mm, 1.6 mm and 2.2 mm tests.

Sample surface profiles for Ω = 5100 rpm with b = 1.6 mm for the new and worn inserts are provided in figures 5-11 and 5-12, respectively. The figures display the topography of the machined surface and were obtained using a scanning white light interferometer with a 10x magnification and 2.5 mm by 1 mm field of view.

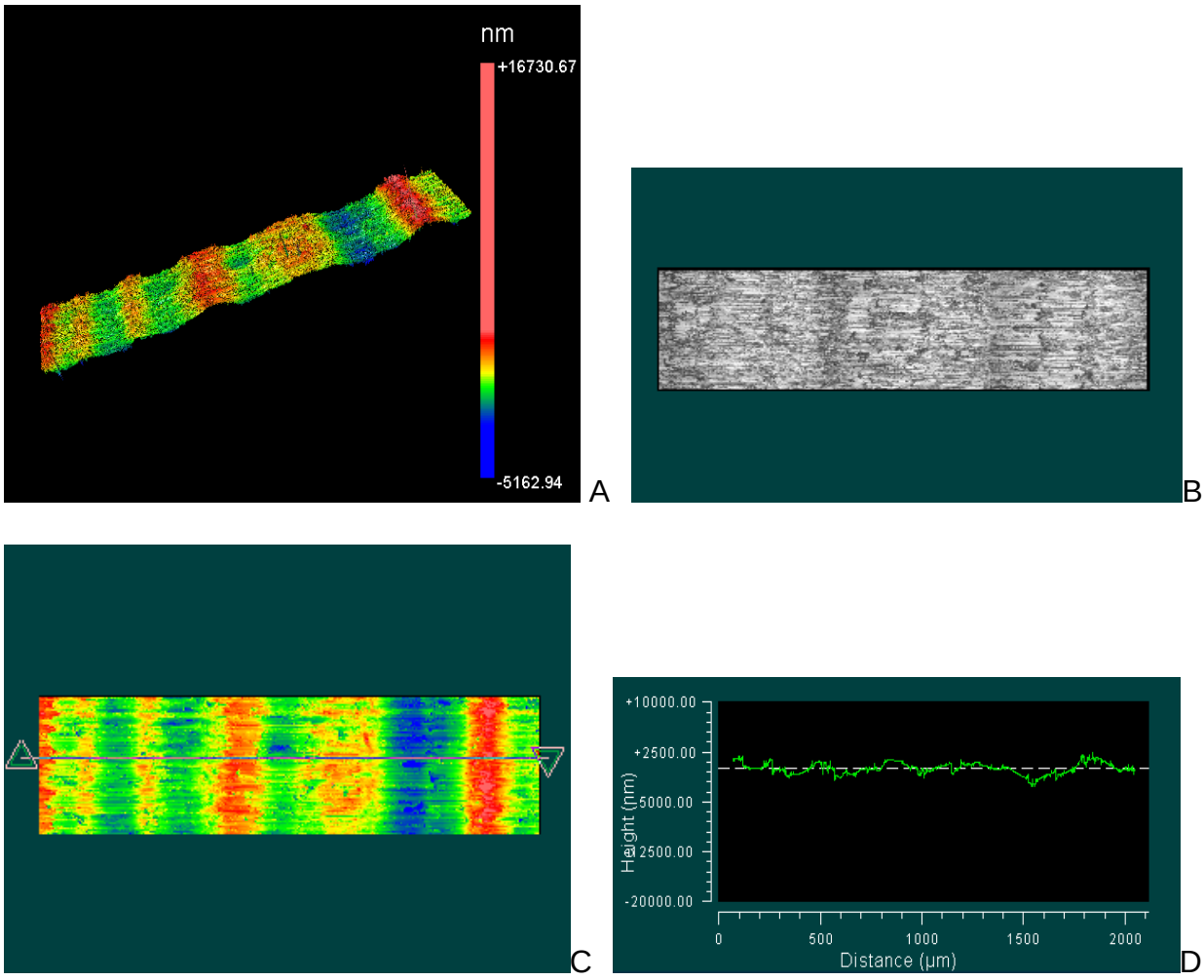


Figure 5-11. Topography of the surface left by the new insert; A) Topography of the surface shown in B). The image B) does not show any distinctive cutting marks indicative of chatter. C) is the color contoured image of stable cutting, with the line indicating the location at which the profile, D), was obtained in the feed direction.

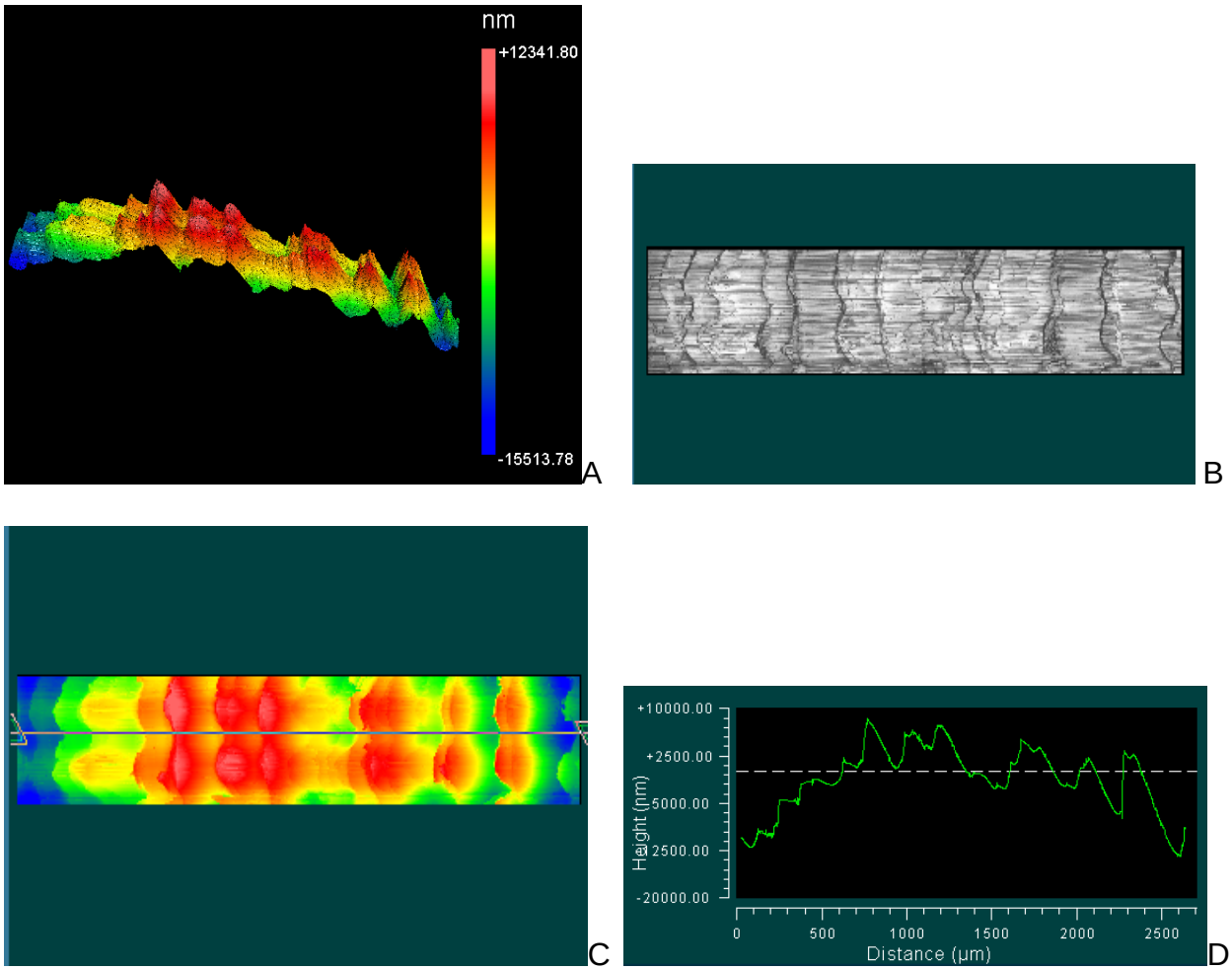


Figure 5-12. Topography of the surface left by the worn insert; A) Topography of the surface shown in B). B) shows wavy marks indicative of chatter. C) is the color contoured image, with the line indicating the location at which the profile, D), was obtained.

The average surface roughness for figure 5-11 D is 781.8 nm; this surface does not show any distinct chatter marks. The average surface roughness for the unstable result (worn tool) in figure 5-12 D is 4018.5 nm, which is five times the stable result obtained using the new insert. This shows that the stability limit decreases with increased tool wear.

Additional tests were completed to verify the decrease in stability limit due to tool wear. All the conditions were same as the previous tests (Ω = 5100 rpm, a = 19.05 mm

and f_t = 0.06 mm/tooth) and the force was measured using the dynamometer. The increase in K_t and K_n with volume removed (see figure 4-12) was used to determine the new coefficient values and the corresponding stability limit. Since the stability testing was carried out at different axial and radial depths of cut than tool wear experiments, the increase in K_t and K_n at 5000 rpm was plotted against volume normalized by both, i.e., $V_n = V/(a*b)$. Figure 5-13 shows the increase in K_t and K_n at 5000 rpm with normalized volume

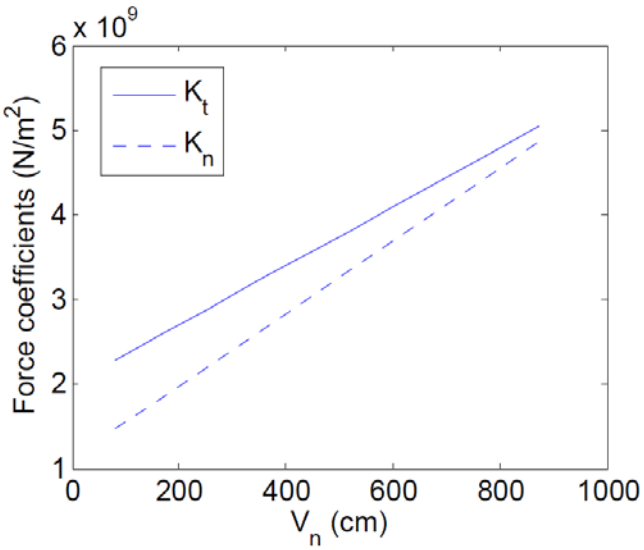


Figure 5-13. Variation in K_t and K_n with volume (normalized by the axial and radial depths of cut). Note that this is the same plot as figure 4-12 for 5000 rpm, except for the normalized volume ($V_n = V/(a*b)$).

From figure 5-13, the initial values (at V_n =0) for the cutting force coefficients are 2.0×10^9 N/m^2 and 1.1×10^9 N/m^2, while the slopes of these lines are 3.49×10^5 N/m^2/cm and 4.30×10^9 N/m^2/cm for K_t and K_n. For the additional stability evaluation, testing was continued at Ω = 5100 rpm and b = 2.5 mm until the R value reached 0.004, which was taken to indicate chatter based on figure 5-10. The normalized volume removed per test cut was 10.1 cm. A series of stability lobes at the corresponding V_n

were generated based on the increased coefficients (see figure 5-14); the new coefficients were selected using the intercept and slope values from figure 5-13.

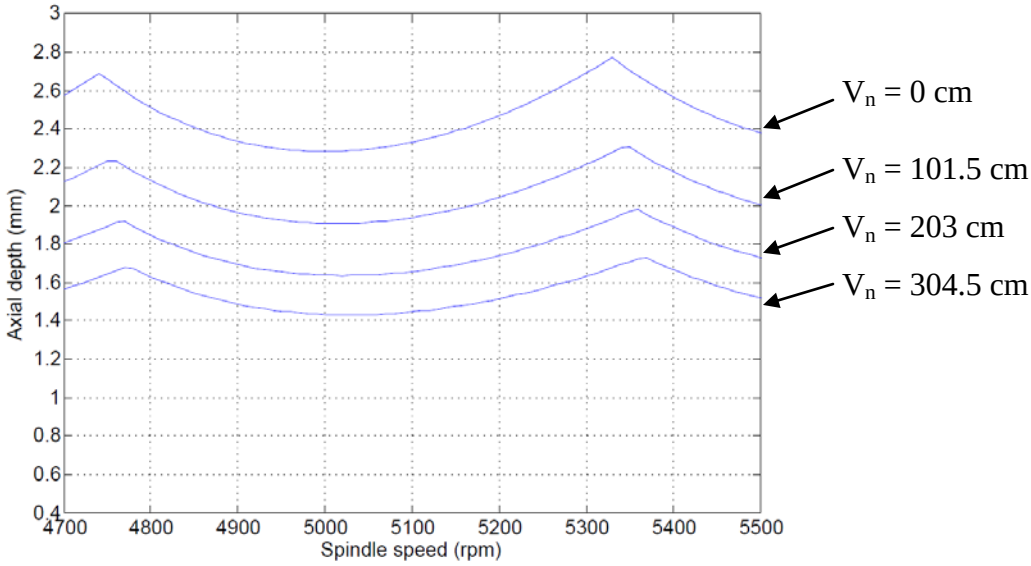


Figure 5-14. Stability lobes generated at V_n = {0,101.5,203 and 304.5} mm.

As seen from figure 5-14, the stability limit decreases with volume removed. Initial testing was performed at b = 2.5 mm. Figure 5-15 shows the variation in R with V_n. As seen in the figure, the R value reaches 0.0036 at a V_n of approximately 200 cm. The transition from stable to unstable behavior occurred later than predicted by figure 5-14 (should have been unstable by V_n ~ 100 cm). The disagreement is attributed to uncertainty in the slopes from figure 5-13 as well as the force coefficients (see figures 4-12 and 4-14) . When the threshold R value was reached (chatter), the axial depth of cut was reduced to 2.0 mm and testing was continued. It is seen that the R value rapidly increased and quickly exceeded 0.004. This agrees with the stability limit predicted by figure 5-14. The axial depth was decreased to 1.5 mm at V_n ~ 300 cm by reducing b = 1.5 mm. The R value dropped, but remained higher than the chatter threshold value. Figure 5-16 shows the once-per-revolution force data for selected tests.

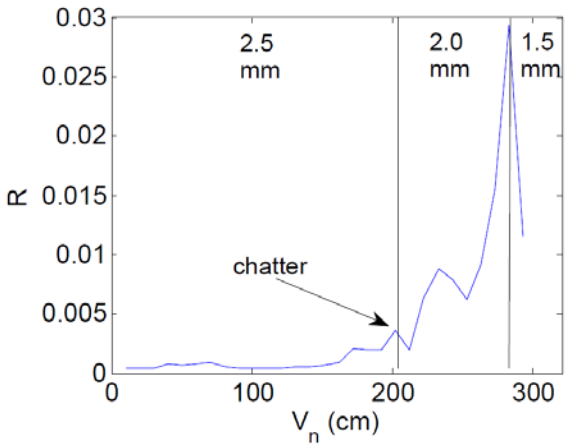


Figure 5-15. Variation in R with V_n

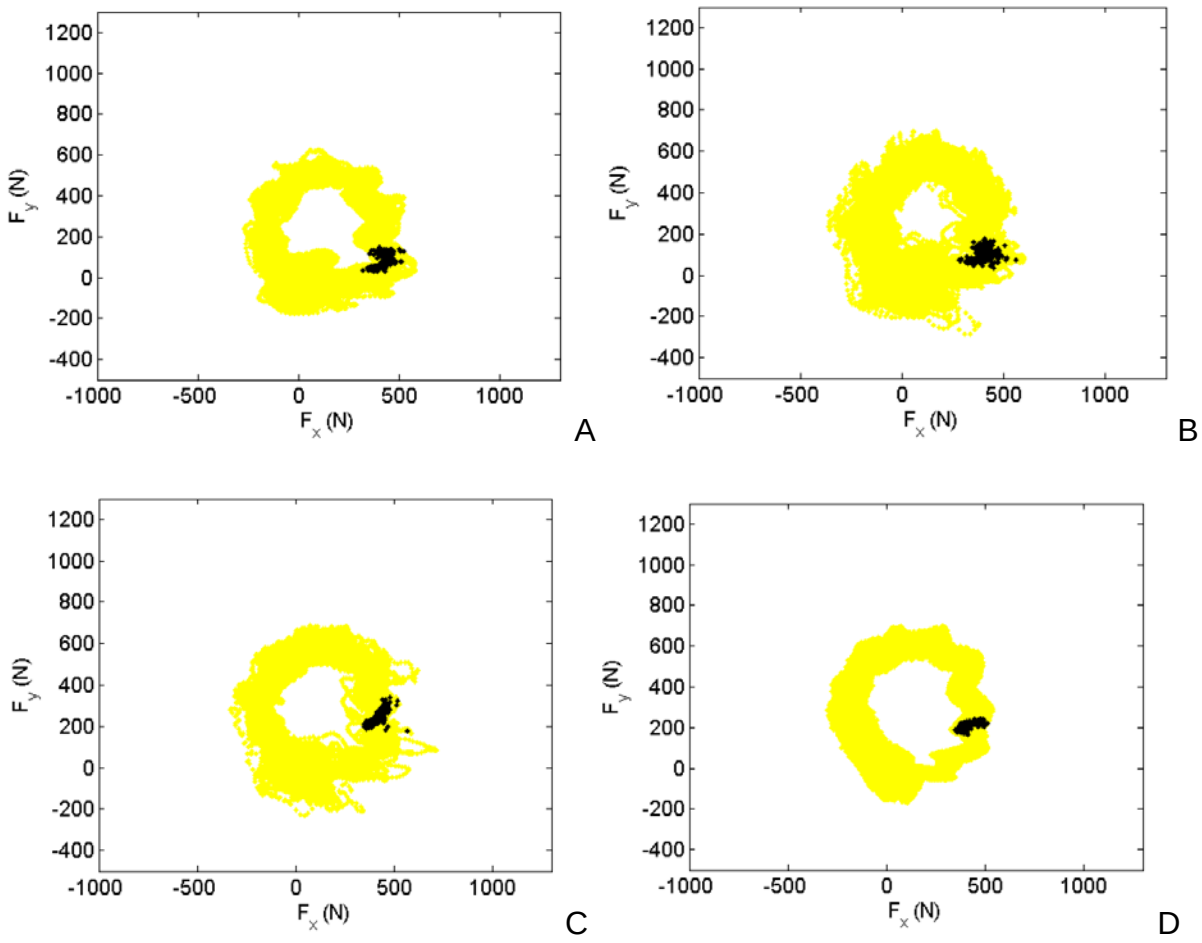


Figure 5-16. Once-per-revolution plots for tests at Ω = 5100 rpm. Plots A-G) corresponds to test numbers 1, 5, 9, 12, 16, 18 and 20 at b = 2.5 mm. Plots H-J) corresponds to test numbers 21, 25 and 28 at b = 2.0 mm. Plots K) and L) correspond to test numbers 29 and 30 at b = 1.5 mm. Each test included another V_n = 10.1 cm increment.

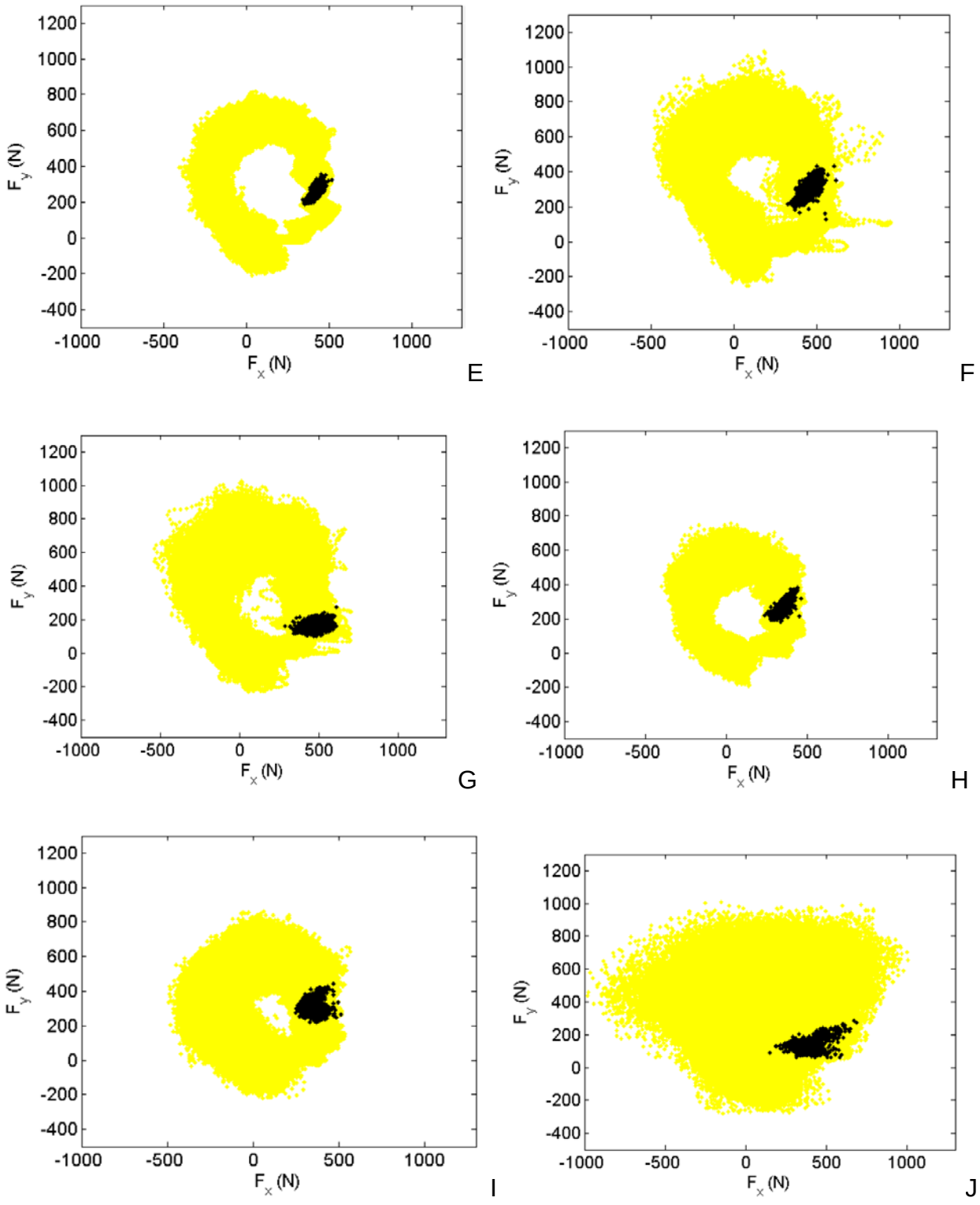
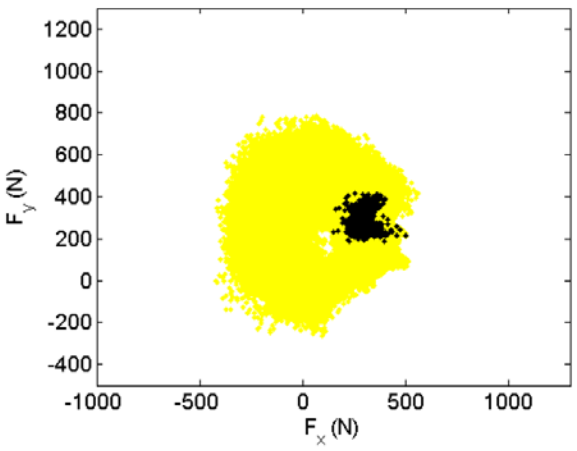
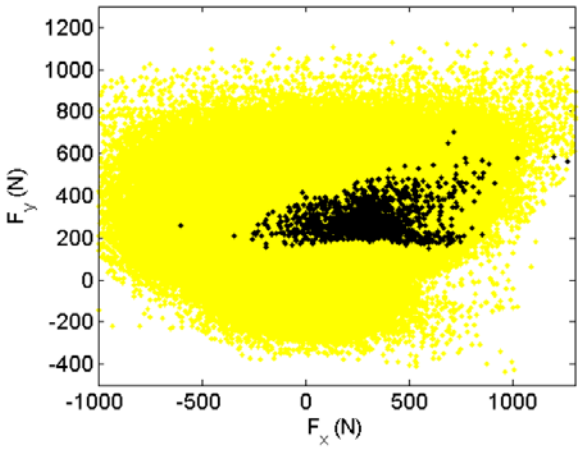


Figure 5-16. Continued.



K



L

Figure 5-16. Continued.

Note the increased scatter of the once-per-revolution data as testing continues from test numbers 1 to 20. Also, the scatter decreases as b is reduced from 2.5 mm to 2.0 mm at test number 20, which is reflected in the reduced R value. Similar results are obtained for further testing at 2.0 mm and 1.5 mm as shown. These test results further confirm the reduction in stability limit with tool wears (modeled here as the increase in force coefficients K_t and K_n).

CHAPTER 6
CONCLUSIONS AND FUTURE WORK

Completed Work

Tool wear is an important process limitation in milling. In this work, the effect of tool wear on cutting force was determined experimentally. By representing the force variation using a force model with appropriate cutting force coefficients, tool wear could be incorporated as a process limitation together with stability and SLE in the milling super diagram. Therefore, a new comprehensive milling super diagram can be developed that provides information at the process planning stage for stability, surface location error, tool wear, and uncertainty in a user-friendly graphical format. The gray-scale color scheme identifies: 1) stable combinations of axial depth of cut and spindle speed that offer both stable cutting conditions and an acceptable, user-defined surface location error level within a user-selected safety margin (white); 2) stable cutting conditions that meet the deterministic SLE limit but are not within the safety margin (light gray); 3) stable cutting conditions that do not meet the surface location error limit (dark gray); and 4) unstable cutting conditions (black). A numerical case study was presented to describe the diagram development for user-specified values of SLE, volume of material removed, and safety margins.

Tests were completed to establish the variation in cutting force coefficients with tool wear as a function of spindle speed and volume removed for a zero rake/zero helix angle, 15 deg relief angle, square, uncoated carbide insert used to machine 1018 steel. The single insert was mounted in a 19 mm diameter steel tool body. For the selected insert-material pair, it was observed that the cutting coefficients, which relate the tangential and normal force components to chip area, increased linearly with volume

removed and the corresponding slope increased linearly with spindle speed. However, the edge (plowing) coefficients, that relate the forces to chip width only, showed no appreciable trend with tool wear. Tests also showed the increases in force coefficients with volume are independent of axial, b, and radial, a, depths of cut. The volume removed can therefore be normalized by both b and a and the results can be extended to any other combination. Using this force model, a comprehensive super diagram can be developed for the user-selected volume to be removed by incorporating tool wear through an appropriate increase in the force coefficients.

Finally, stability predictions were validated by two sets of experiments. In the first experiments, stability lobes were calculated for new and worn inserts based on the corresponding force coefficient values and cutting tests were completed to validate the stability limits. In the second experiments, the increase in force coefficients with volume from tool wear experiments were used to determine the stability lobes at different volumes. In both cases, the normalized statistical variance in the once-per-revolution sampled force signal was used to identify chatter. The tests showed good agreement with the predicted stability limits.

Future Work

This study investigated the effect of tool wear on cutting force coefficients and the corresponding influence on process stability. Tests showed a linear increase in cutting force coefficients with tool wear, while edge coefficients showed no particular trend for the given tool-material combination. Also, the tests showed that the increase in force coefficients was closely related to flank wear width (FWW). Similar experiments can be carried out for different tool-material combinations to determine if the cutting force coefficients show similar trends in both their growth and relationship to FWW with

volume removed. Two options are possible: 1) the FWW can be used to update the force coefficients for a worn tool without requiring the knowledge of volume removed; or 2) the cutting force can be monitored and used to identify the wear state and corresponding force coefficients. Also, in the tool wear tests performed here, no crater wear was observed. Future experiments can be completed to study the effect of different forms of tool wear on force coefficients.

The increase in force coefficients causes not only the stability limit to decrease, but also a proportional increase in the SLE as shown in the numerical study. Tests can be completed to determine the effect of tool wear on SLE. In the stability tests, a once-per-revolution sampling strategy and data analysis was used instead of the analyzing the frequency spectrum of the cutting force signal. The frequency spectrum analysis was not an effective tool for detecting chatter for worn insert due to increased magnitude of forces and energy in the tooth passing frequency harmonics. Future work can be completed to explore effective analysis techniques for identifying chatter with worn tools.

The super diagram can be improved by the addition of surface roughness as a process limitation. Surface roughness depends strongly on the feed per tooth. Therefore, the selection of feed per tooth for the super diagram can be based on the surface roughness. Additionally, uncertainty can be better incorporated by using the probability distribution functions of the input parameters and propagating them through the milling process model, rather than the safety margin approach applied here.

APPENDIX A
CALCULATION OF FORCE COEFFICIENTS

```
% Jaydeep Karandikar
% Coefficient testing

close all
clear all
clc

% Remeber that of the 1 and 2 versions of this test the signs have to be
% reversed from one another.
%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Feed Per tooth DATA
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%

FT = [0.03 0.04 0.05 0.06 0.07]*1e-3; % chip load m/tooth
X_sensitivity=1;
Y_sensitivity=1;

mean_FX = zeros(5,5);
mean_FY = mean_FX;

Nt = 1;
a = 4.7625e-3;
b = 3e-3;
d = 19.05e-3; % teeth diameter, m
gamma = 0; % helix angle, deg
phis = 120*pi/180;
phie = 180*pi/180;

%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %%%%M0
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tic
FILENAME = 'C:\Study\Dynamics of Prod
machinary\Tests\Test19\0measure_6250_0.03_3_25%_down_pass1.pcs';
low_limit=5;
high_limit=11;

[Signal,Time] = pcscopenew(FILENAME);
Signal = -Signal;
Fx=Signal(:,1)*X_sensitivity;
Fy=Signal(:,2)*Y_sensitivity;

index=find(Time>=low_limit & Time<=high_limit);

% id_noise = find(Time<=0.5);
% remove_noise_avg_X = mean(Fx(id_noise));
% remove_noise_avg_Y = mean(Fy(id_noise));
```

```
% F_X = Fx(index)-remove_noise_avg_X ; % Force in X for feed rate i
% F_Y = Fy(index)-remove_noise_avg_Y ; % Force in Y for feed rate i
% %
% for cnt1 = 1:250
%     for cnt2 = 1:1000
%         F_X(cnt2+(cnt1-1)*2000) = 0;
%     end
% end


F_X =Fx(index);
F_Y = Fy(index);

FX_mean1 = mean(F_X);
FY_mean1 = mean(F_Y);

FILENAME = 'C:\Study\Dynamics of Prod
machinary\Tests\Test19\0measure_6250_0.04_3_25%_down_pass1.pcs';
low_limit=2;
high_limit=7;

[Signal,Time] = pcscopenew(FILENAME);
Signal = -Signal;
Fx=Signal(:,1)*X_sensitivity;
Fy=Signal(:,2)*Y_sensitivity;

index=(Time>=low_limit & Time<=high_limit);
F_X = Fx(index);
F_Y = Fy(index);

FX_mean2 = mean(F_X);
FY_mean2 = mean(F_Y);

FILENAME = 'C:\Study\Dynamics of Prod
machinary\Tests\Test19\0measure_6250_0.05_3_25%_down_pass1.pcs';
low_limit=2;
high_limit=7;

[Signal,Time] = pcscopenew(FILENAME);
Signal = -Signal;
Fx=Signal(:,1)*X_sensitivity;
Fy=Signal(:,2)*Y_sensitivity;

index=(Time>=low_limit & Time<=high_limit);
F_X = Fx(index);
F_Y = Fy(index);

FX_mean3 = mean(F_X);
FY_mean3 = mean(F_Y);

FILENAME = 'C:\Study\Dynamics of Prod machinery\
machinary\Tests\Test19\0measure_6250_0.06_3_25%_down_pass2.pcs';
low_limit =3;
high_limit=8;
```

```
[Signal,Time] = pcscopenew(FILENAME);
Signal = -Signal;
Fx=Signal(:,1)*X_sensitivity;
Fy=Signal(:,2)*Y_sensitivity;

index1=(Time>=low_limit & Time<=high_limit);
F_X = Fx(index1);
F_Y = Fy(index1);

FX_mean4 = mean(F_X);
FY_mean4 = mean(F_Y);

FILENAME = 'C:\Study\Dynamics of Prod
machinary\Tests\Test19\0measure_6250_0.07_3_25%_down_pass2.pcs';
low_limit =2;
high_limit=7;

[Signal,Time] = pcscopenew(FILENAME);
Signal = -Signal;
Fx=Signal(:,1)*X_sensitivity;
Fy=Signal(:,2)*Y_sensitivity;

index1=(Time>=low_limit & Time<=high_limit);
F_X = Fx(index1);
F_Y = Fy(index1);

FX_mean5 = mean(F_X);
FY_mean5 = mean(F_Y);

mean_FX(1,:) = [FX_mean1 FX_mean2 FX_mean3 FX_mean4 FX_mean5];
mean_FY(1,:) = [FY_mean1 FY_mean2 FY_mean3 FY_mean4 FY_mean5];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Linear regression using the mean-force data points

p1(1,:) = polyfit(FT,mean_FX(1,:),1);
p2(1,:) = polyfit(FT,mean_FY(1,:),1);

a0x(1) = p1(1,2);
a1x(1) = p1(1,1);
a0y(1) = p2(1,2);
a1y(1) = p2(1,1);
%
rx2(1) = (sum((mean_FX(1,:) - mean(mean_FX(1,:))).^2) - sum((mean_FX(1,:) -
a0x(1) - a1x(1)*FT).^2))/(sum((mean_FX(1,:) - mean(mean_FX(1,:))).^2));
ry2(1) = (sum((mean_FY(1,:) - mean(mean_FY(1,:))).^2) - sum((mean_FY(1,:) -
a0y(1) - a1y(1)*FT).^2))/(sum((mean_FY(1,:) - mean(mean_FY(1,:))).^2));

%calculate coefficients
kte_fit(1) = (pi*(a0x(1)*(sin(phie)-sin(phis))-a0y(1)*(cos(phie)-
cos(phis))))/(Nt*b*(1-cos(phie-phis)));                    % Eq. 4.7.24
(modify this line based on your solution)
```

```
kne_fit(1) = (pi*(a0x(1)*(cos(phie)-cos(phis))+a0y(1)*(sin(phie)-
sin(phis))))/(-Nt*b*(1-cos(phie-phis)));         % Eq. 4.7.24 (modify this
line based on your solution)

kt_fit(1) = 8*pi*(a1y(1)*(2*phie-2*phis+sin(2*phis)-sin(2*phie))+
a1x(1)*(cos(2*phis)-cos(2*phie))) / ((Nt*b)*((2*phie-2*phis+sin(2*phis)-
sin(2*phie))^2 + (cos(2*phis)-cos(2*phie))^2));
kn_fit(1) = 8*pi*(a1x(1)*(2*phie-2*phis+sin(2*phis)-sin(2*phie))-
a1y(1)*(cos(2*phis)-cos(2*phie))) / ((Nt*b)*((2*phie-2*phis+sin(2*phis)-
sin(2*phie))^2 + (cos(2*phis)-cos(2*phie))^2));

figure(1)
plot(FT, mean_FX(1,:), 'bo', FT, mean_FY(1,:), 'bs')
hold on
set(gca,'FontSize', 14)
xlabel('f_t (m/tooth)')
ylabel('Mean Force (N)')
legend('x', 'y')
%best fit lines
plot(FT, a0x(1)+a1x(1)*FT, 'b:', FT, a0y(1)+a1y(1)*FT, 'b--')
hold on
% %
Toc
```

APPENDIX B
SUPER DIAGRAM CODE

```
% Raúl E. Zapata
% Call program for super diagram data collection
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This program calls the stability_alt_trim program to create stability
% lobes and test a series of operating points to see if they are above or
% below the stability limit (unstable or stable).
% Afterwards the program inputs these same points into a surface location
% error calculation to obtain the contours of levels of error.
% These are then combined to form a diagram delimiting the areas that are
% and are not of use according to the user selected bounds of surface
% location error.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all
clear all
clc

% Inputs.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Variables %
%%%%%%%%%%%%%

% Max Real Values 0.75mm cuts
Ktc_vec     = 2000e6;                                                  % tangential
cutting force coefficient, N/m^2
Krc_vec     = 666e6;                                                 % radial
cutting force coefficient, N/m^2
Kte_vec     = 0;                          % tangential edge constant, N/m
Kre_vec     = 0;                          % radial edge constant, N/m


% Tool description
N       = 1;                         % teeth, integer
beta    = 0;                        % helix angle, deg
d       = 19.05*1e-3;                     % teeth diameter, m

% Stability lobes
Num_lobes   = 40;       % number of stability lobes to be calculated.
ss          = 2000:10:20000;  % spindle speed, rpm

% Machining specifications for test cases

ft      = .06e-3;          % feed per tooth (m/tooth)
RDOC    = .25*d;             % Radial depth of cut.

rho = RDOC/d;               % Radial Immersion as a fractional quantity not a
percent

% Milling direction used to determine entry and exit angle.
%1 = upmilling and 2 = downmilling
```

```
dir     = 2;

ADM     = 0;

SLE_MAX = 20;

%%%%%%%%%%%%%%%%%%%%%%%%
% FRF
%%%%%%%%%%%%%%%%%%%%%%%%
% MODAL FIT (I created)
%
n_teeth     = 1;                    % # of teeth on the tool
SS_best1    = 18000;               % "best speed" arbitrary value for the peak
around the first lobe.
F_best1     = SS_best1/60*n_teeth;  %

% Tool description
ky = [5e6];                      % N/m
zetay = [0.05];                  % damping coefficient
wny = [F_best1]*2*pi;            % rad/s
my = ky./(wny.^2);               % kg
cy = 2*zetay.*(my.*ky).^0.5;     % N-s/m

% Define y-direction FRF (could read in measurement from TXF)
fmeas = 0:1:5000;              % Hz
w = fmeas*2*pi;                  % rad/s
FRFy = (wny(1)^2/ky(1))./(wny(1)^2 - w.^2 + i*2*zetay(1)*wny(1).*w);

% this for loop adds up any other modes of vibration to the FRF if they
% exist in the formulation above.
for cnt = 2:length(ky)
    FRFy = FRFy + (wny(cnt)^2/ky(cnt))./(wny(cnt)^2 - w.^2 +
i*2*zetay(cnt)*wny(cnt).*w);
end

FRFx = FRFy; % for this test I am assumming the X and Y dynamics are the
same.

figure
subplot(211)
plot(fmeas,real(FRFx),'b')
subplot(212)
plot(fmeas,imag(FRFx),'b')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% .TXF FILE CALL

% Assumptions:
% 1. Files already have the hammer and sensor calibration factor in them, but
they are presented anyway just in case.
% 2. G consists of 6 columns the first of which is the FRF that we want, so
it will be extracted.
% 3. f consists if the frequencies used in the G vector.
```

```
% % Y direction
% Fmin = 220;
% Fmax = 5000;
%
%
% [FILENAME,PATH] = uigetfile('*.txf');
% FILENAME = [PATH FILENAME];
% [f G] = Txfnew(FILENAME);
%
% index   = find(f>Fmin & f<Fmax);
% G       = G(index,1);
% f       = f(index);
%
% fmeas   = f;
% FRFy    = G;
%
% % X direction
% [FILENAME,PATH] = uigetfile('*.txf');
% FILENAME = [PATH FILENAME];
% [f G] = Txfnew(FILENAME);
%
% index   = find(f>Fmin & f<Fmax);
% G       = G(index,1);
% f       = f(index);
%
% fmeas   = f;
% FRFx    = G;
%
% figure
% subplot(211)
% plot(fmeas,real(FRFx))
% ylabel('Real (m/N)')
% subplot(212)
% plot(fmeas,imag(FRFx))
% xlabel('Frequency (Hz)')
% ylabel('Imaginary (m/N)')


%%%%%%%%%%%%%%%%%%%%%%%%
% Test parameters
%%%%%%%%%%%%%%%%%%%%%%%%

% Define test grid.
spinspeed = linspace(2000,20000,300);
ADOC = linspace(.0001,10,300);

% Create test vectors
count = 0;
for cnt2 = 1:length(ADOC)
    for cnt1 = 1:length(spinspeed)

        count = count+1;
        SS_test(count)  = spinspeed(cnt1);
        ADOC_test(count)= ADOC(cnt2);
```

```
    end
end

clear count
% SS_test     = [5000 5000 5000 7500 7500 7500 9000 15000 15000 18000 18000
18000 18000];   % test point spindle speeds
% ADOC_test   = [.5 1 2 .5 1 2 1 .5 2 .5 1 2 3]/2;
% test point axial depths


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculations and procedures/functions
%%%%%%%%%%%%%%%%%%%%%%%%%

% "MEAN" FMC data calculations

Ktc     = Ktc_vec;
Krc     = Krc_vec;
Kte     = Kte_vec;
Kre     = Kre_vec;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DETERMINING ENTRY AND EXIT ANGLE %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if rho <= 0.5

    % Entry and Exit angles (radians)
    if dir == 1
        phi_in  = zeros(1,length(rho));                     % entry angle
(upmilling)
        phi_out = acos(1-rho*2);        % exit angle (upmilling)
    elseif dir == 2
        phi_in  = pi-acos(1-rho*2);     % entry angle (downmilling)
        phi_out = pi*ones(1,length(rho));                   % exit angle
(downmilling)
    else
        fprintf('The direction you have chosen is incorrect please run again
and select "up = 1" or "down = 2" for the dir variable. \n');
        return
    end
elseif rho > 0.5 & rho<1
    if dir == 1
        phi_in  = zeros(1,length(rho));
        %phi_out = phi_in + pi/2 + acos(1-(rho-0.5)*2);
        phi_out = phi_in + pi/2 + asin(2*rho-1);

    elseif dir == 2
        phi_out = pi*ones(1,length(rho));
        phi_in  = phi_out - pi/2 - asin(2*rho-1);
    else
```

```
        fprintf('The direction you have chosen is incorrect please run again
and select "up = 1" or "down = 2" for the dir variable. \n');
        return
    end
elseif rho == 1
    phi_in = zeros(1,length(rho));
    phi_out = pi*ones(1,length(rho));
elseif rho > 1 || rho <=0;
    fprintf('Please review the radial immersion of the cut, the current value
is illogical. \n');
    return
end


phistart    = phi_in*180/pi;                % starting angle, deg
phiexit     = phi_out*180/pi;               % exit end, deg


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
DTR = pi/180;                               % conversion Degrees to radians

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Stability
%%%%%%%%%%%%%%%%%

% This should output the stability conditions and operating conditions
% selected for testing in a table or matrix form, along with the vectors
% required to create a stability boundary plot (spindle speed and axial
depth).
tic
[Combination_mat,SS_final,Blim_final] = stability_alt_trim(Ktc, Krc, N,
phistart*DTR, phiexit*DTR, ss, Num_lobes,ADOC_test,SS_test, fmeas, FRFx,
FRFy);

figure
plot(SS_final,Blim_final,'b');%,SS_test,ADOC_test,'r*');
xlabel('Spindle speed (rpm)')
ylabel('Axial depth (mm)')
title(num2str(RDOC))
axis([-Inf Inf 0 20])


stab = Combination_mat(:,3);

toc
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SLE
%%%%%%%%%%%%%%%%%

quant1 = length(stab);
SLE_data_mat = zeros(quant1,4);
counter = 0;
tic
for cnt = 1:length(ADOC)
```

```
    [e] = sle_SSvec_capable(fmeas, FRFy, Ktc, Krc, Kte, Kre, N, beta, d,
phistart, phiexit, spinspeed, ADOC(cnt)*1e-3, ft, ADM);
    sle = e*1e6;

    for cnt1 = 1:length(sle)
        counter = counter+1;
        if abs(real(sle(cnt1)))<=SLE_MAX      %acceptable
            SLE_mult = 0;
        elseif abs(real(sle(cnt1)))>SLE_MAX   %unacceptable
            SLE_mult = 1;
        end

        SLE_data_mat(counter,:) = [spinspeed(cnt1) ADOC(cnt) real(sle(cnt1))
SLE_mult];
    end
end
SLE = SLE_data_mat(:,4);


index = find(SLE_data_mat(:,2)>= 2.006 & SLE_data_mat(:,2)<= 2.007);

figure
plot(SLE_data_mat(index,1),SLE_data_mat(index,3));
xlabel('Spindle speed (rpm)')
ylabel('SLE (mm)')


% ADOC_sle = 2.0068;
%     for slecnt = 1:length(spinspeed)
%     [e] = sle_SSvec_capable(fmeas, FRFy, Ktc, Krc, Kte, Kre, N, beta, d,
phistart, phiexit, spinspeed(slecnt), ADOC_sle, ft, ADM);
%     sle_aodc(slecnt) = e*1e6;
%     end
%
% figure
% plot(spinspeed,sle_adoc);
% xlabel('Spindle speed (rpm)')
% ylabel('SLE (mm)')
%
toc


%%%%%%%%%%%%%%%%%%%%%%%%%
% Data for Diagram
%%%%%%%%%%%%%%%%%%%%%%%%%

Total = stab+SLE;%life;

Total_mat = zeros(length(ADOC),length(spinspeed));

for cnt = 1:length(ADOC)
    Total_mat(cnt,:) = Total(1+(cnt-
1)*length(spinspeed):cnt*length(spinspeed)).';
end
```

```
%
% figure
% %contourf(spinspeed,ADOC,Total_mat)
% [c, h] = contour(spinspeed,ADOC,Total_mat,[0:3]);
% xlabel('Spindle speed (rpm)')
% ylabel('Axial Depth (mm)')
% clabel(c,h)
%
figure
contourf(spinspeed,ADOC,Total_mat,[0:3]);
xlabel('Spindle speed (rpm)')
ylabel('Axial Depth (mm)')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
%MRR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
for cnt2 = 1:length(SS_test)
    MRR1(cnt2) = RDOC*ft*N*SS_test(cnt2)*ADOC_test(cnt2);
end

%MRR_max = max(MRR)
for cnt3 = 1:length(SS_test)
    if Total(cnt3) == 1 | Total(cnt3) == 2 | Total(cnt3) == 3 | Total(cnt3)
== 4
        MRR(cnt3) = 0;
    elseif Total(cnt3) == 0
        MRR(cnt3) = MRR1(cnt3);
    end
end

MRR_max = max(MRR);
a=0;
b=0;


for cnt4 = 1:length(SS_test)
    Max_MRR = RDOC*ft*N*SS_test(cnt4)*ADOC_test(cnt4);
    if Max_MRR == MRR_max
        Max_Speed = SS_test(cnt4);
        Max_ADOC = ADOC_test(cnt4);
        b= cnt4;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
%Safety limits to Parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%

SS_tol = 100;        %Safety limit for Spindle speed
ADOC_tol = 0.5;      %Safety limit for Axial Depth of Cut
```

```
ds = spinspeed(2)-spinspeed(1);
da = ADOC(2)-ADOC(1);

SS_factor = ceil(SS_tol/ds);     % grid steps
ADOC_factor = ceil(ADOC_tol/da); % grid steps

MRR_safety = zeros(1,length(SS_test));

for cnt5 = 1:length(ADOC)
    counter = cnt5-1;
    for cnt6 = 1:length(spinspeed)

        if cnt5<=ADOC_factor
            if cnt6<=SS_factor
                if
Total_mat(cnt5+ADOC_factor,cnt6+SS_factor)==0&Total_mat(cnt5+ADOC_factor,cnt6
)==0&Total_mat(cnt5,cnt6+SS_factor)==0
                    MRR_safety(cnt6+counter*length(spinspeed)) =
MRR(cnt6+counter*length(spinspeed));
                end
            elseif cnt6>=SS_factor
                if cnt6>=length(spinspeed)-SS_factor
                    if Total_mat(cnt5+ADOC_factor,cnt6-
SS_factor)==0&Total_mat(cnt5+ADOC_factor,cnt6)==0&Total_mat(cnt5,cnt6-
SS_factor)==0
                        MRR_safety(cnt6+counter*length(spinspeed)) =
MRR(cnt6+counter*length(spinspeed));
                    end
                elseif cnt6<length(spinspeed)-SS_factor
                    if Total_mat(cnt5+ADOC_factor,cnt6-
SS_factor)==0&Total_mat(cnt5+ADOC_factor,cnt6+SS_factor)==0&Total_mat(cnt5+AD
OC_factor,cnt6)==0&Total_mat(cnt5,cnt6+SS_factor)==0&Total_mat(cnt5,cnt6-
SS_factor)==0
                        MRR_safety(cnt6+counter*length(spinspeed)) =
MRR(cnt6+counter*length(spinspeed));
                    end
                end
            end
        elseif cnt5>=ADOC_factor
            if cnt5>=length(ADOC)-ADOC_factor
                if cnt6<=SS_factor
                    if Total_mat(cnt5-
ADOC_factor,cnt6+SS_factor)==0&Total_mat(cnt5,cnt6+SS_factor)==0&Total_mat(cn
t5-ADOC_factor,cnt6)==0
                        MRR_safety(cnt6+counter*length(spinspeed)) =
MRR(cnt6+counter*length(spinspeed));
                    end
                elseif cnt6>=SS_factor
                    if cnt6>=length(spinspeed)-SS_factor
                        if Total_mat(cnt5-ADOC_factor,cnt6-
SS_factor)==0&Total_mat(cnt5-ADOC_factor,cnt6)==0&Total_mat(cnt5,cnt6-
SS_factor)==0 %check for surroundind 9 data points
                            MRR_safety(cnt6+counter*length(spinspeed)) =
MRR(cnt6+counter*length(spinspeed));
                        end
                    elseif cnt6<length(spinspeed)-SS_factor
```

```
                        if Total_mat(cnt5-
ADOC_factor,cnt6+SS_factor)==0&Total_mat(cnt5-ADOC_factor,cnt6-
SS_factor)==0&Total_mat(cnt5,cnt6+SS_factor)==0&Total_mat(cnt5-
ADOC_factor,cnt6)==0&Total_mat(cnt5,cnt6-SS_factor)==0 %check for surroundind
9 data points
                            MRR_safety(cnt6+counter*length(spinspeed)) =
MRR(cnt6+counter*length(spinspeed));
                        end
                    end
                end
            elseif cnt5<length(ADOC)-ADOC_factor
                if cnt6<=SS_factor
                    if
Total_mat(cnt5+ADOC_factor,cnt6+SS_factor)==0&Total_mat(cnt5-
ADOC_factor,cnt6+SS_factor)==0&Total_mat(cnt5+ADOC_factor,cnt6)==0&Total_mat(
cnt5,cnt6+SS_factor)==0&Total_mat(cnt5-ADOC_factor,cnt6)==0
                        MRR_safety(cnt6+counter*length(spinspeed)) =
MRR(cnt6+counter*length(spinspeed));
                    end
                elseif cnt6>=SS_factor
                    if cnt6>=length(spinspeed)-SS_factor
                        if Total_mat(cnt5+ADOC_factor,cnt6-
SS_factor)==0&Total_mat(cnt5-ADOC_factor,cnt6-
SS_factor)==0&Total_mat(cnt5+ADOC_factor,cnt6)==0&Total_mat(cnt5-
ADOC_factor,cnt6)==0&Total_mat(cnt5,cnt6-SS_factor)==0 %check for surroundind
9 data points
                            MRR_safety(cnt6+counter*length(spinspeed)) =
MRR(cnt6+counter*length(spinspeed));
                        end
                    elseif cnt6<length(spinspeed)-SS_factor
                        if Total_mat(cnt5+ADOC_factor,cnt6-
SS_factor)==0&Total_mat(cnt5+ADOC_factor,cnt6+SS_factor)==0&Total_mat(cnt5-
ADOC_factor,cnt6+SS_factor)==0&Total_mat(cnt5-ADOC_factor,cnt6-
SS_factor)==0&Total_mat(cnt5+ADOC_factor,cnt6)==0&Total_mat(cnt5,cnt6+SS_fact
or)==0&Total_mat(cnt5-ADOC_factor,cnt6)==0&Total_mat(cnt5,cnt6-SS_factor)==0
%check for surroundind 9 data points
                            MRR_safety(cnt6+counter*length(spinspeed)) =
MRR(cnt6+counter*length(spinspeed));
                        end
                    end
                end
            end
        end
    end
end


Max_MRR_safety = max(MRR_safety);

for cnt9 = 1:length(SS_test)
    if MRR_safety(cnt9) == Max_MRR_safety;
        Max_Speed_safety = SS_test(cnt9);
        Max_ADOC_safety = ADOC_test(cnt9);
        a= cnt9;
    end
end
```

```
[row,col] = size(Total_mat);
MRR_mat = zeros(row,col);

for cnt = 1:row
    MRR_mat(cnt,:) = MRR_safety(1+(cnt-1)*col:cnt*col);
end
%
% figure
% contourf(spinspeed,ADOC,MRR_mat)

id_MRR = find(MRR_mat>0);

Saferegion = MRR_mat;
Saferegion(id_MRR) = 1;
%
% figure
% contourf(spinspeed,ADOC,Saferegion)
% xlabel('Spindle speed (rpm)')
% ylabel('Axial depth (mm)')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Repeat with zero limits for comparison
SS_tol = 0;        %Safety limit for Spindle speed
ADOC_tol = 0;      %Safety limit for Axial Depth of Cut

ds = spinspeed(2)-spinspeed(1);
da = ADOC(2)-ADOC(1);

SS_factor = ceil(SS_tol/ds);     % grid steps
ADOC_factor = ceil(ADOC_tol/da); % grid steps

MRR_safety = zeros(1,length(SS_test));

for cnt5 = 1:length(ADOC)
    counter = cnt5-1;
    for cnt6 = 1:length(spinspeed)

        if cnt5<=ADOC_factor
            if cnt6<=SS_factor
                if
Total_mat(cnt5+ADOC_factor,cnt6+SS_factor)==0&Total_mat(cnt5+ADOC_factor,cnt6
)==0&Total_mat(cnt5,cnt6+SS_factor)==0
                    MRR_safety(cnt6+counter*length(spinspeed)) =
MRR(cnt6+counter*length(spinspeed));
                end
            elseif cnt6>=SS_factor
                if cnt6>=length(spinspeed)-SS_factor
                    if Total_mat(cnt5+ADOC_factor,cnt6-
SS_factor)==0&Total_mat(cnt5+ADOC_factor,cnt6)==0&Total_mat(cnt5,cnt6-
SS_factor)==0
                        MRR_safety(cnt6+counter*length(spinspeed)) =
MRR(cnt6+counter*length(spinspeed));
                    end
```

```
                elseif cnt6<length(spinspeed)-SS_factor
                    if Total_mat(cnt5+ADOC_factor,cnt6-
SS_factor)==0&Total_mat(cnt5+ADOC_factor,cnt6+SS_factor)==0&Total_mat(cnt5+AD
OC_factor,cnt6)==0&Total_mat(cnt5,cnt6+SS_factor)==0&Total_mat(cnt5,cnt6-
SS_factor)==0
                        MRR_safety(cnt6+counter*length(spinspeed)) =
MRR(cnt6+counter*length(spinspeed));
                    end
                end
            end
        elseif cnt5>=ADOC_factor
            if cnt5>=length(ADOC)-ADOC_factor
                if cnt6<=SS_factor
                    if Total_mat(cnt5-
ADOC_factor,cnt6+SS_factor)==0&Total_mat(cnt5,cnt6+SS_factor)==0&Total_mat(cn
t5-ADOC_factor,cnt6)==0
                        MRR_safety(cnt6+counter*length(spinspeed)) =
MRR(cnt6+counter*length(spinspeed));
                    end
                elseif cnt6>=SS_factor
                    if cnt6>=length(spinspeed)-SS_factor
                        if Total_mat(cnt5-ADOC_factor,cnt6-
SS_factor)==0&Total_mat(cnt5-ADOC_factor,cnt6)==0&Total_mat(cnt5,cnt6-
SS_factor)==0 %check for surroundind 9 data points
                            MRR_safety(cnt6+counter*length(spinspeed)) =
MRR(cnt6+counter*length(spinspeed));
                        end
                    elseif cnt6<length(spinspeed)-SS_factor
                        if Total_mat(cnt5-
ADOC_factor,cnt6+SS_factor)==0&Total_mat(cnt5-ADOC_factor,cnt6-
SS_factor)==0&Total_mat(cnt5,cnt6+SS_factor)==0&Total_mat(cnt5-
ADOC_factor,cnt6)==0&Total_mat(cnt5,cnt6-SS_factor)==0 %check for surroundind
9 data points
                            MRR_safety(cnt6+counter*length(spinspeed)) =
MRR(cnt6+counter*length(spinspeed));
                        end
                    end
                end
            elseif cnt5<length(ADOC)-ADOC_factor
                if cnt6<=SS_factor
                    if
Total_mat(cnt5+ADOC_factor,cnt6+SS_factor)==0&Total_mat(cnt5-
ADOC_factor,cnt6+SS_factor)==0&Total_mat(cnt5+ADOC_factor,cnt6)==0&Total_mat(
cnt5,cnt6+SS_factor)==0&Total_mat(cnt5-ADOC_factor,cnt6)==0
                        MRR_safety(cnt6+counter*length(spinspeed)) =
MRR(cnt6+counter*length(spinspeed));
                    end
                elseif cnt6>=SS_factor
                    if cnt6>=length(spinspeed)-SS_factor
                        if Total_mat(cnt5+ADOC_factor,cnt6-
SS_factor)==0&Total_mat(cnt5-ADOC_factor,cnt6-
SS_factor)==0&Total_mat(cnt5+ADOC_factor,cnt6)==0&Total_mat(cnt5-
ADOC_factor,cnt6)==0&Total_mat(cnt5,cnt6-SS_factor)==0 %check for surroundind
9 data points
                            MRR_safety(cnt6+counter*length(spinspeed)) =
MRR(cnt6+counter*length(spinspeed));
                        end
```

```
                    elseif cnt6<length(spinspeed)-SS_factor
                        if Total_mat(cnt5+ADOC_factor,cnt6-
SS_factor)==0&Total_mat(cnt5+ADOC_factor,cnt6+SS_factor)==0&Total_mat(cnt5-
ADOC_factor,cnt6+SS_factor)==0&Total_mat(cnt5-ADOC_factor,cnt6-
SS_factor)==0&Total_mat(cnt5+ADOC_factor,cnt6)==0&Total_mat(cnt5,cnt6+SS_fact
or)==0&Total_mat(cnt5-ADOC_factor,cnt6)==0&Total_mat(cnt5,cnt6-SS_factor)==0
%check for surroundind 9 data points
                            MRR_safety(cnt6+counter*length(spinspeed)) =
MRR(cnt6+counter*length(spinspeed));
                        end
                    end
                end
            end
        end
    end
end


Max_MRR_safety = max(MRR_safety);

for cnt9 = 1:length(SS_test)
    if MRR_safety(cnt9) == Max_MRR_safety;
        Max_Speed_safety = SS_test(cnt9);
        Max_ADOC_safety = ADOC_test(cnt9);
        a= cnt9;
    end
end

[row,col] = size(Total_mat);
MRR_mat = zeros(row,col);

for cnt = 1:row
    MRR_mat(cnt,:) = MRR_safety(1+(cnt-1)*col:cnt*col);
end
%
% figure
% contourf(spinspeed,ADOC,MRR_mat)

id_MRR1 = find(MRR_mat>0);

Saferegion1 = MRR_mat;
Saferegion1(id_MRR1) = 1;
%
% figure
% contourf(spinspeed,ADOC,Saferegion1)
% xlabel('Spindle speed (rpm)')
% ylabel('Axial depth (mm)')

Comparison = Saferegion1-Saferegion;

figure
contourf(spinspeed,ADOC,Comparison)
xlabel('Spindle speed (rpm)')
ylabel('Axial depth (mm)')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% figure
% %contourf(spinspeed,ADOC,Total_mat)
% [c, h] = contour(spinspeed,ADOC,Total_mat,[0:3]);
% xlabel('Spindle speed (rpm)')
% ylabel('Axial Depth (mm)')

save numerical_study_19_2000_20.mat SLE_data_mat Combination_mat spinspeed
ADOC MRR Comparison
```

Stability Code

```
% Raúl E. Zapata
% Altintas Stability Lobes Trimmed with search algorithm for stable and
% unstable individual points.
% October 2006
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [Combination_mat,SS_final,Blim_final] = stability_alt_trim(kt, kn,
N, phis, phie, spin_speed, Num_lobes, ADOC_vec, SS_vec, f, Gxx, Gyy)

% This function provides the full set off tested points along with the
% spindle speed and Blim vectors for the stability lobe diagram they were
% compared to. The last column on Combination_mat is the stability
% multiplier where 1 means that the parameter set is stable and zero means
% the parameter set is unstable.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculations
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

w_temp = f*2*pi;
w = zeros(1,length(w_temp));

for cnt = 1:length(w_temp)
    w(1,cnt) = w_temp(cnt);
end

%   Enter in the Cutting Coefficient Information
Ktc = kt;
Krc = kn;

% calculate cutting coefficients for monte carlo run
Kta=Ktc;    Kra=Krc/Ktc;

%  -----------------------------------------Altintas Code

alphaxx=0.5*((cos(2*phie)-2*Kra*phie+Kra*sin(2*phie))-(cos(2*phis)-
2*Kra*phis+Kra*sin(2*phis)));
alphaxy=0.5*((-sin(2*phie)-2*phie+Kra*cos(2*phie))-(-sin(2*phis)-
2*phis+Kra*cos(2*phis)));
alphayx=0.5*((-sin(2*phie)+2*phie+Kra*cos(2*phie))-(-
sin(2*phis)+2*phis+Kra*cos(2*phis)));
```

```
alphayy=0.5*((-cos(2*phie)-2*Kra*phie-Kra*sin(2*phie))-(-cos(2*phis)-
2*Kra*phis-Kra*sin(2*phis)));

% Calculate and Sort Eigenvalues

eigen1 = zeros(1,length(w));
eigen2 = zeros(1,length(w));


for cnt=1:length(w)
       A=[alphaxx*Gxx(cnt) alphaxy*Gyy(cnt);alphayx*Gxx(cnt)
alphayy*Gyy(cnt)];
       E=eig(A);
       temp=E(1);
       eigen1(cnt)=temp;
       temp=E(2);
       eigen2(cnt)=temp;
       if (cnt>1)
           dot_prod1=real(eigen2(cnt))*real(eigen2(cnt-
1))+imag(eigen2(cnt))*imag(eigen2(cnt-1));
           dot_prod2=real(eigen2(cnt))*real(eigen1(cnt-
1))+imag(eigen2(cnt))*imag(eigen1(cnt-1));
           if(dot_prod2>dot_prod1)
               temp=eigen2(cnt);
               eigen2(cnt)=eigen1(cnt);
               eigen1(cnt)=temp;
           end
       end
end
eigen1=eigen1';
eigen2=eigen2';

% Calculate alim values for each eigenvalue
alim1=(2*pi/N/Kta)./((real(eigen1)).^2+(imag(eigen1)).^2).*(real(eigen1).*(1+
(imag(eigen1)./real(eigen1)).^2));
alim2=(2*pi/N/Kta)./((real(eigen2)).^2+(imag(eigen2)).^2).*(real(eigen2).*(1+
(imag(eigen2)./real(eigen2)).^2));

% Choose positive Values of alim1
[index1]=find(alim1>0);
alim1=alim1(index1);
alim1=alim1*1000;
w1=w(index1).';
psi1=atan2(imag(eigen1),real(eigen1));
psi1=psi1(index1);
epsilon1=(pi-2*psi1);                    % Note that this is a column vector

%Choose positive Values of alim2
[index2]=find(alim2>0);
alim2=alim2(index2);
alim2=alim2*1000;
w2=w(index2).';
psi2=atan2(imag(eigen2),real(eigen2));
psi2=psi2(index2);
epsilon2=pi-2*psi2;                      % Note that this is a column vector
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Matrix Formulation

N1_matrix = zeros(length(epsilon1),Num_lobes);
N2_matrix = zeros(length(epsilon2),Num_lobes);

for cnt1 = 1:Num_lobes
    N1_matrix(:,cnt1) = (60/N)*w1./(epsilon1+2*(cnt1-1)*pi);
    N2_matrix(:,cnt1) = (60/N)*w2./(epsilon2+2*(cnt1-1)*pi);
end

SS_final = zeros(1,length(spin_speed));
Blim_final = SS_final;

for cnt2=1:length(spin_speed)

    SS_final(cnt2)=spin_speed(cnt2);

    %  Initiallize Search Matrices
    blim1_temp = zeros(length(epsilon1),Num_lobes);
    blim1_temp(1,:)=1e20;
    blim2_temp = zeros(length(epsilon2),Num_lobes);
    blim2_temp(1,:)=1e20;

    % Populate search matrices by linearly interpolating the desired SS
points.
    for cnt3 = 1:Num_lobes
        for cnt4=2:length(epsilon1)

            if(spin_speed(cnt2)>N1_matrix(cnt4,cnt3) &
spin_speed(cnt2)<N1_matrix(cnt4-1,cnt3) |
spin_speed(cnt2)<N1_matrix(cnt4,cnt3) & spin_speed(cnt2)>N1_matrix(cnt4-
1,cnt3))
                blim1_temp(cnt4,cnt3)=alim1(cnt4-1)+(alim1(cnt4)-alim1(cnt4-
1))*(spin_speed(cnt2)-N1_matrix(cnt4-1,cnt3))/(N1_matrix(cnt4,cnt3)-
N1_matrix(cnt4-1,cnt3));
            else
                blim1_temp(cnt4,cnt3)=1e20;
            end
        end

        for cnt4=2:length(epsilon2)

            if(spin_speed(cnt2)>N2_matrix(cnt4,cnt3) &
spin_speed(cnt2)<N2_matrix(cnt4-1,cnt3) |
spin_speed(cnt2)<N2_matrix(cnt4,cnt3) & spin_speed(cnt2)>N2_matrix(cnt4-
1,cnt3))
                blim2_temp(cnt4,cnt3)=alim2(cnt4-1)+(alim2(cnt4)-alim2(cnt4-
1))*(spin_speed(cnt2)-N2_matrix(cnt4-1,cnt3))/(N2_matrix(cnt4,cnt3)-
N2_matrix(cnt4-1,cnt3));
            else
                blim2_temp(cnt4,cnt3)=1e20;
```

```
            end
        end

    end
    Blim1_min = min(min(blim1_temp));           % Find minimum boundary for
alim1 lobes at this SS
    Blim2_min = min(min(blim2_temp));           % Find minimum boundary for
alim2 lobes at this SS

    Blim_final(cnt2) = min([Blim1_min Blim2_min]);   % Find global minimum
boundary for this SS
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Test parameters stability check

Counter = 0;
Combination_mat = zeros(length(ADOC_vec),3);

%for cnt50 = 1:length(ADOC_vec)
    for cnt51 = 1:length(SS_vec)
        cnt50 = cnt51;
        Counter = Counter+1;

        index_low = max(find(SS_final < SS_vec(cnt51)));
        index_high= min(find(SS_final >= SS_vec(cnt51)));

        SS_left = SS_final(index_low);
        SS_right = SS_final(index_high);
        B_lim_left = Blim_final(index_low);
        B_lim_right = Blim_final(index_high);

        Blim_value = B_lim_right+(B_lim_left-B_lim_right)*(SS_vec(cnt51)-
SS_right)/(SS_left-SS_right);

        if Blim_value > ADOC_vec(cnt50)         %stable
            STAB_mult = 0;
        elseif Blim_value <= ADOC_vec(cnt50);   %unstable
            STAB_mult = 2;
        end

        Combination_mat(Counter,:) = [ADOC_vec(cnt50) SS_vec(cnt51)
STAB_mult];
    end

%end
```

SLE Code

```
function [e] = sle_SSvec_capable(fmeas, FRFmeas, kt, kn, kte, kne, m, beta,
d, phistart, phiexit, omega, b, ft, ADM)
% T. Schmitz (10/20/05)
```

```
% This is a program to find the SLE in helical peripheral end milling using a
frequency domain approach.
% Regeneration is not considered.

% Consider only up and downmilling cases
if phistart == 0
    flag = 0 + 2*ADM*tan(beta)/d;   % upmilling
else
    flag = 180 + 2*ADM*tan(beta)/d; % downmilling
end

DTR = pi/180;                   % degrees to radians conversion

% Simulation specifications
n = 11;
steps = 2^n;                    % steps for one cutter revolution, int
dt = 60./(steps*omega);          % integration time step, s
dphi = 360/steps;               % angular steps size between time steps, deg
if beta == 0                    % straight teeth
    db = b;                     % discretized axial depth, m
else                            % nonzero helix angle
    db = d*(dphi*DTR)/2/tan(beta*DTR);
end
steps_axial = round(b/db);      % number of steps along tool axis
tooth_angle = 0:360/m:(360-360/m);  % angles of m cutter teeth starting from
zero, deg

% Initialize vectors
teeth = round(tooth_angle/dphi) + 1;
phi = linspace(0, (steps-1)*dphi, steps);
Force_y = zeros(1, steps);

%************************ MAIN PROGRAM
(Force)*****************************

for cnt1 = 1:steps              % time steps, s
    for cnt2 = 1:m
        teeth(cnt2) = teeth(cnt2) + 1;      % index teeth pointer one
position (rotate cutter by dphi)
        if teeth(cnt2) > steps
            teeth(cnt2) = 1;
        end
    end

    Fy = 0;
    for cnt3 = 1:m                          % sum forces over all teeth, N
        for cnt4 = 1:steps_axial            % sum forces along axial depth of
helical endmill, N
            phi_counter = teeth(cnt3) - (cnt4-1);
            if phi_counter < 1              % helix has wrapped through phi =
0 deg
                phi_counter = phi_counter + steps;
            end
            phia = phi(phi_counter);        % angle for given axial disk, deg
```

```
            if (phia >= phistart) && (phia <= phiexit)   % verify that tooth
angle is in specified range for current disk, deg
                h = ft*sin(phia*DTR);                % chip thickness, m
                ftan = kt*db*h + kte*db;
                frad = kn*db*h + kne*db;
            else    % tooth angle is outside range bounded by radial
immersion
                ftan = 0;
                frad = 0;
            end

            Fy = Fy - frad*cos(phia*DTR) + ftan*sin(phia*DTR);    % N
        end     % cnt4 loop
    end         % cnt3 loop

    Force_y(cnt1) = Fy;
end             % cnt1 loop

%************************* END OF MAIN PROGRAM
*****************************

e = zeros(1,length(dt));

for cnt_dt = 1:length(dt)

    % Compute FFT of Fourier-based y-direction force
    [FY, freq] = spec(Force_y', 1/dt(cnt_dt));          % compute FFT (in
spec.m, there should be no multiplication by T)
    FY = FY/(2^n);                                  % correct magnitude to N

    % Define y-direction FRF on proper frequency vector, freq
    index = find(freq >= min(fmeas) & freq <= max(fmeas));
    freqtemp = freq(index);
    FRFytemp2 = interp1(fmeas, FRFmeas, freqtemp, 'spline');
    FRFytemp1 = ones(1, (index(1)-1))*FRFytemp2(1);
    FRFytemp3 = ones(1, (length(freq)-
index(length(index))))*FRFytemp2(length(FRFytemp2));
    FRFy = [FRFytemp1 FRFytemp2' FRFytemp3]';

    Yf = FY.*FRFy;                          % F * X/F = X

    Yf_dc = Yf(1);                          % DC component
    Yf(1) = 0;                              % DC extraction

    y = real(ifft(Yf*(2^n)));               % convert to time-domain
    y = y+Yf_dc;                            % DC insertion to create full
signal

    t = 0:2*dt(cnt_dt):(length(y)-1)*2*dt(cnt_dt);              % new time
vector after inverse FFT

    % Use automatic method to sample y
    tfirst = (flag*DTR)/(omega(cnt_dt)/60*2*pi);     % time for first SLE
point, s
```

```
    index = find(t > tfirst);
    first_point = index(1) - 1;                 % first point for SLE
    index = first_point:round(60/(omega(cnt_dt)*m)/(2*dt(cnt_dt))):length(t);
    y_sampled = y(index);                       % sampled position vector --
SLE values
    e(cnt_dt) = y_sampled(length(y_sampled));           % record SLE, m

end
```

Fast Fourier Transform Code Used in SLE Calculations

```
% Computes the fft X of signal x and the corresponding frequency vector f
given
% the sampling frequency fs.
%
% [X,f]=spec(x,fs)
%
% [X,f]=spec(x,fs,'whole') returns values around the whole unit circle

function [X,f]=spec(x,fs,whole)
T=1/fs;
N=length(x);
%X=T*fft(x);
X=fft(x);
f=[0:fs/N:(1-1/(2*N))*fs]';
if nargin == 2
 X=X(1:N/2+1,:);
 f=f(1:N/2+1,:);
end
```

APPENDIX C
TIME DOMAIN SIMULATION

Code for Time Domain Simulation

```
% University of Florida
% T. Schmitz

clc
close all
clear all

%Define cutting force coefficients
% Ks = 2000e6;                      % N/m^2
% beta = 60;                       % deg
kt = 4.9883e9;%Ks*sin(beta*pi/180);
kn = 4.5145e9;%Ks*cos(beta*pi/180);
kte = 45500;%55600;
kne = 25500;

% Define modal parameters for x direction
load MODAL_FIT_short2.mat k_x wn_x zeta_x k_y wn_y zeta_y
ky = 1.5*k_y;
zetay = zeta_y;
wny = wn_y;

kx = 1.5*k_x;
zetax = zeta_x;
wnx = wn_x;

my = ky./(wny.^2);              % kg
cy = 2*zetay.*(my.*ky).^0.5;    % N-s/m
x_modes = length(ky);           % number of modes in x direction

% Define modal parameters for y direction
mx = kx./(wnx.^2);              % kg
cx = 2*zetax.*(mx.*kx).^0.5;    % N-s/m
y_modes = length(kx);           % number of modes in y direction


% Define cutting parameters
Nt = 1;
d = 19.05e-3;                      % teeth diameter, m
gamma = 0;                      % helix angle, deg
phis = 0;                        % deg
phie = 180;                     % deg
omega = 5100;                   % rpm
b = 1.6e-3;                    % m
ft = 0.06e-3;                     % m
steps_rev = 360;

% Calculate actual steps_rev to ensure integer value
% for ratio (steps_rev/Nt)
temp = round(steps_rev/Nt);
```

```
steps_rev = temp*Nt;
dt = 60/(steps_rev*omega);      % s
dphi = 360/steps_rev;           % deg
if gamma == 0
    db = b;
else
    % discretized axial depth, m
    db = d*(dphi*pi/180)/2/tan(gamma*pi/180);
end
% number of steps along tool axis
steps_axial = round(b/db);
rev = 200;
steps = rev*steps_rev;

% Initialize vectors
for cnt = 1:Nt
    teeth(cnt) = (cnt-1)*steps_rev/Nt + 1;
end
for cnt = 1:steps_rev
    phi(cnt) = (cnt - 1)*dphi;
end

surf = zeros(steps_axial, steps_rev);

% Euler integration initial conditions
x = 0;
y = 0;
dp = zeros(1, x_modes);
p = zeros(1, x_modes);          % x-direction modal displacements, m
dq = zeros(1, y_modes);
q = zeros(1, y_modes);          % y-direction modal displacements, m

% Function to keep track of simulation progress
handle = waitbar(0, 'Please wait... simulation in progress.');

for cnt1 = 1:steps
    waitbar(cnt1/steps, handle)

    for cnt2 = 1:Nt
        teeth(cnt2) = teeth(cnt2) + 1;
        if teeth(cnt2) > steps_rev
            teeth(cnt2) = 1;
        end
    end

    Fx = 0;
    Fy = 0;

    for cnt3 = 1:Nt
        for cnt4 = 1:steps_axial
            phi_counter = teeth(cnt3) - (cnt4-1);
            if phi_counter < 1          % helix has wrapped through phi = 0
deg
                phi_counter = phi_counter + steps_rev;
            end
```

```
            phia = phi(phi_counter);    % angle for given axial disk, deg

            if (phia >= phis) & (phia <= phie)
                n = x*sin(phia*pi/180) - y*cos(phia*pi/180);            % m
                h = ft*sin(phia*pi/180) + surf(cnt4, phi_counter) - n;  % m
                if h < 0
                    Ft = 0;
                    Fn = 0;
                    surf(cnt4, phi_counter) = surf(cnt4, phi_counter) +
ft*sin(phia*pi/180);
                else
                    Ft = kt*db*h + kte*db;
                    Fn = kn*db*h + kne*db;
                    surf(cnt4, phi_counter) = n;
                end
            else
                Ft = 0;
                Fn = 0;
            end

            Fx = Fx + Ft*cos(phia*pi/180) + Fn*sin(phia*pi/180);
            Fy = Fy + Ft*sin(phia*pi/180) - Fn*cos(phia*pi/180);
        end
    end

    Forcex(cnt1) = Fx;
    Forcey(cnt1) = Fy;

    % Numerical integration for position
    x = 0;
    y = 0;

    % x direction
    for cnt5 = 1:x_modes
        ddp = (Forcex(cnt1) - cx(cnt5)*dp(cnt5) - kx(cnt5)*p(cnt5))/mx(cnt5);
        dp(cnt5) = dp(cnt5) + ddp*dt;
        p(cnt5) = p(cnt5) + dp(cnt5)*dt;
        x = x + p(cnt5);        % m
            end
        ax(cnt1) = ddp;
    % y direction
    for cnt5 = 1:y_modes
        ddq = (Forcey(cnt1) - cy(cnt5)*dq(cnt5) - ky(cnt5)*q(cnt5))/my(cnt5);
        dq(cnt5) = dq(cnt5) + ddq*dt;
        q(cnt5) = q(cnt5) + dq(cnt5)*dt;
        y = y + q(cnt5);        % m
          end
    xpos(cnt1) = x;
    ypos(cnt1) = y;
      ay(cnt1) = ddq;
end

close(handle);              % close progress bar
time = ((1:steps)-1)*dt;    % s
```

```
figure(1)
subplot(211)
plot(time, Forcex, 'b')
hold on
xlim([0 max(time)])
set(gca,'FontSize', 14)
ylabel('F_x (N)')
subplot(212)
plot(time, xpos*1e6)
xlim([0 max(time)])
set(gca,'FontSize', 14)
xlabel('t (s)')
ylabel('x (\mum)')

figure(2)
subplot(211)
plot(time, Forcey, 'b')
hold on
xlim([0 max(time)])
set(gca,'FontSize', 14)
ylabel('F_y (N)')
subplot(212)
plot(time, ypos*1e6)
xlim([0 max(time)])
set(gca,'FontSize', 14)
xlabel('t (s)')
ylabel('y (\mum)')

F = (Forcex.^2 + Forcey.^2).^0.5;   % N
figure(3)
plot(time, F)
%axis([0.09 0.1 0 1500])
set(gca,'FontSize', 14)
xlabel('t (s)')
ylabel('F (N)')

low_limit =2.3e-3;
high_limit=low_limit+200*60/omega;

index1=(time>=low_limit & time<=high_limit);
Forcex1 = Forcex(index1);
Forcey1 = Forcey(index1);
time = time(index1);
x1 = xpos(index1);
y1 = ypos(index1);

fx2 = Forcex1(1:steps_rev:length(Forcex1));
time2 = time(1:steps_rev:length(time));
fy2 = Forcey1(1:steps_rev:length(Forcey1));

x2 = x1(1:steps_rev:length(x1));
y2 = y1(1:steps_rev:length(y1));

figure(9)
subplot(211)
```

```
plot(time, Forcex1,'y')
hold on
plot(time2,fx2,'k.')
set(gca,'FontSize', 14)
xlabel('Time (sec)')
ylabel('F_x (N)')
subplot(212)
plot(time, Forcey1,'y')
hold on
plot(time2,fy2,'k.')
set(gca,'FontSize', 14)
xlabel('Time (sec)')
ylabel('F_y (N)')


figure(10)
plot(Forcex1,Forcey1,'y.')
hold on
plot(fx2,fy2,'k.')
set(gca,'FontSize', 14)
xlabel('F_x (N)')
ylabel('F_y (N)')

figure(11)
subplot(211)
plot(time, x1*1e6,'y')
hold on
plot(time2,x2*1e6,'k.')
set(gca,'FontSize', 14)
xlabel('Time (sec)')
ylabel('x (\mum)')
subplot(212)
plot(time, y1,'y')
hold on
plot(time2,y2,'k.')
set(gca,'FontSize', 14)
xlabel('Time (sec)')
ylabel('y (\mum)')

figure(12)
plot(x1,y1,'y.')
hold on
plot(x2,y2,'k.')
set(gca,'FontSize', 14)
xlabel('x (\mum)')
ylabel('y (\mum)')
% %
```

Code for Modal Fitting

```
% Raúl Zapata
% Peak Picking modal fit
% Based off Scott Duncan's code
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all; clear all; clc;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FRF's %
%%%%%%%%%

% % % % Mid-length tool
% % % load G_middlex.mat G_mass f;
% % % FRF_xdir = G_mass;
% % % freq1 = f;
% % % clear G_mass f
% % %
% % % load G_middley.mat G_mass f;
% % % FRF_ydir = G_mass;
% % % clear G_mass f

% Short tool
load FRFx.mat FRFx fmeas;
FRF_xdir = FRFx;
freq1 = fmeas;
clear FRFx fmeas

load FRFy.mat FRFy fmeas;
FRF_ydir = FRFy;
clear FRFy fmeas

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% X DIRECTION

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURES FOR PEAK PICKING %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Plots of the unprocessed holder/spindle data

figure(1)
subplot(211)
plot(freq1,real(FRF_xdir))
title('Unprocessed Frequency Response Function, Holder/Spindle, Real: X')
xlabel('Frequency (hz)')
ylabel('Real (m/N)')
subplot(212)
plot(freq1,imag(FRF_xdir))
title('Unprocessed Frequency Response Function, Holder/Spindle, Imaginary:
X')
xlabel('Frequency (hz)')
ylabel('Imaginary (m/N)')

%  Choose number of modes

number_modes = input('How many modes do you wish to fit?');

%  Select the points for peak picking
imag_min_x = ginput(number_modes);
real_max_x = ginput(number_modes);
real_min_x = ginput(number_modes);
```

```
%  Calculate the values for the initial guess parameters for nonlinear least
squares fit

for n = 1:number_modes
    fn_x(n) = imag_min_x(n,1);
    wn_x(n) = fn_x(n)*2*pi;
    zeta_x(n) = abs((real_max_x(n,1)-real_min_x(n,1))/(2*imag_min_x(n,1)));
    k_x(n) = -1/(2*imag_min_x(n,2)*zeta_x(n));
    m_x(n) = k_x(n)/wn_x(n)^2;
    c_x(n) = zeta_x(n)*2*sqrt(k_x(n)*m_x(n));
end

f = 1:10000;
w = f*2*pi;

G_measured_x = 0;
for n = 1:number_modes
G_measured_x = ones(size(w))./(k_x(n)*ones(size(w))-
m_x(n)*w.^2+i*c_x(n)*w)+G_measured_x;
end

% Write m, c, and k back to main gui program

figure(2)
subplot(211)
plot(freq1,real(FRF_xdir),f,real(G_measured_x))
title('Actual Versus Fit FRF for Holder/Spindle, Real: X')
xlabel('Frequency (hz)')
ylabel('Real (m/N)')
legend('Actual','Fit')
subplot(212)
plot(freq1,imag(FRF_xdir),f,imag(G_measured_x))
title('Actual Versus Fit FRF for Holder/Spindle, Imaginary: X')
xlabel('Frequency (hz)')
ylabel('Imaginary (m/N)')
legend('Actual','Fit')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Y DIRECTION

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIGURES FOR PEAK PICKING %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  Plots of the unprocessed holder/spindle data

figure(3)
subplot(211)
plot(freq1,real(FRF_ydir))
title('Unprocessed Frequency Response Function, Holder/Spindle, Real: Y')
xlabel('Frequency (hz)')
ylabel('Real (m/N)')
subplot(212)
plot(freq1,imag(FRF_ydir))
title('Unprocessed Frequency Response Function, Holder/Spindle, Imaginary:
Y')
```

```
xlabel('Frequency (hz)')
ylabel('Imaginary (m/N)')

%  Choose number of modes

number_modes = input('How many modes do you wish to fit?');

%  Select the points for peak picking
imag_min_y = ginput(number_modes);
real_max_y = ginput(number_modes);
real_min_y = ginput(number_modes);

%  Calculate the values for the initial guess parameters for nonlinear least
squares fit

for n = 1:number_modes
    fn_y(n) = imag_min_y(n,1);
    wn_y(n) = fn_y(n)*2*pi;
    zeta_y(n) = abs((real_max_y(n,1)-real_min_y(n,1))/(2*imag_min_y(n,1)));
    k_y(n) = -1/(2*imag_min_y(n,2)*zeta_y(n));
    m_y(n) = k_y(n)/wn_y(n)^2;
    c_y(n) = zeta_y(n)*2*sqrt(k_y(n)*m_y(n));
end

f = 1:10000;
w = f*2*pi;

G_measured_y = 0;
for n = 1:number_modes
G_measured_y = ones(size(w))./(k_y(n)*ones(size(w))-
m_y(n)*w.^2+i*c_y(n)*w)+G_measured_y;
end

% Write m, c, and k back to main gui program

figure(4)
subplot(211)
plot(freq1,real(FRF_ydir),f,real(G_measured_y))
title('Actual Versus Fit FRF for Holder/Spindle, Real: Y')
xlabel('Frequency (hz)')
ylabel('Real (m/N)')
legend('Actual','Fit')
subplot(212)
plot(freq1,imag(FRF_ydir),f,imag(G_measured_y))
title('Actual Versus Fit FRF for Holder/Spindle, Imaginary: Y')
xlabel('Frequency (hz)')
ylabel('Imaginary (m/N)')
legend('Actual','Fit')

% SAVING DATA
save MODAL_FIT_short2.mat k_x wn_x zeta_x k_y wn_y zeta_y
```

APPENDIX D
ONCE PER REVOLUTION AND VARIANCE CODE

```
 % Jaydeep Karandikar
% Coefficient testing

close all
clear all
clc

% Remeber that of the 1 and 2 versions of this test the signs have to be
% reversed from one another.
%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Feed Per tooth DATA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

FT = [0.01 0.02 0.03 0.04 0.05]*1e-3; % chip load m/tooth
X_sensitivity=1;
Y_sensitivity=1;

Nt = 1;
a = 19.05e-3;
b = 3e-3;
d = 19.05e-3; % teeth diameter, m
gamma = 0; % helix angle, deg
phis = 0*pi/180;
phie = 180*pi/180;
omega = 5100; % rpm
tic

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%M0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
handle = waitbar(0, 'Please wait... simulation in progress.');
for cnt=1:181
    waitbar(cnt/181, handle)
FILENAME = 'C:\Study\Dynamics of Prod
machinary\Tests\Stability_11_15_Final\worn_tool_5100rpm_0.4adoc_100ri_0pt06ft
.pcs';

[Signal,Time] = pcscopenew(FILENAME);
Signal = -Signal;
Fx=Signal(:,1)*X_sensitivity;
Fy=Signal(:,2)*Y_sensitivity;

time_rev = 1/(omega/60);
sample_pts = 1080;
incr = time_rev/sample_pts;

low_limit =2.0161+(cnt-1)*(5.8823e-3/180);%1.9607e-4;%-
floor(1.0085/time_rev)*time_rev;
high_limit=low_limit+15;
```

```
index1=(Time>=low_limit & Time<=high_limit);
FX = Fx(index1);
FY = Fy(index1);
Time = Time(index1);

time_rev = 1/(omega/60);
sample_pts = 1080;
incr = time_rev/sample_pts;

dt = low_limit:incr:high_limit; %
F_X = interp1(Time,FX,dt);
F_Y = interp1(Time,FY,dt);

fx = F_X(sample_pts:sample_pts:length(F_X));
time = dt(sample_pts:sample_pts:length(dt));
fy = F_Y(sample_pts:sample_pts:length(F_Y));

Vx_rev(cnt) = var(fx);
Vy_rev(cnt) = var(fy);
% Vx(cnt) = var(FX);
% Vy(cnt) = var(FY);
% R(cnt) = (Vx_rev(cnt)^2+Vy_rev(cnt)^2)/(Vx(cnt)+Vy(cnt));


end
toc
Vxrev = mean(Vx_rev)
Vyrev = mean(Vy_rev)
Vx = var(FX)
Vy = var(FY)
R = (Vxrev^2+Vyrev^2)/(Vx^2+Vy^2)

close(handle);              % close progress bar

figure
subplot(211)
plot(Time, FX,'b')
hold on
plot(time,fx,'k.')
xlabel('Time (sec)','FontSize', 14)
ylabel('F_x (N)','FontSize', 14)
xlim([low_limit high_limit])
ylim([-1000 1300])
set(gca,'FontSize', 14)
subplot(212)
plot(Time, FY, 'b')
hold on
plot(time,fy,'k.')
xlabel('Time (sec)','FontSize', 14)
ylabel('F_y (N)','FontSize', 14)
ylim([-500 1300])
xlim([low_limit high_limit])
set(gca,'FontSize', 14)
```

LIST OF REFERENCES

[1] J. Tlusty, W. Zaton, F. Ismail, Stability Lobes in Milling, Annals of the CIRP, 32/1 (1983) 309–313.

[2] J Gradisek, E Govekar, I Grabec, M Kalveram, On stability prediction for low radial immersion milling, Journal of Machine Science and Technology, 9 (220) 117–130.

[3] B. Mann, P. Bayly, M. Davies ,J. Halley, Limit cycles, bifurcations, and accuracy of the milling process, Journal of Sound and Vibration, 277 (1-2) (2004) 31–48.

[4] P. Bayly, J. Halley, B. Mann, M. Davies, Stability of Interrupted Cutting by Temporal Finite Element Analysis, Journal of Manufacturing Science Engineering 125 (2) (2003) 220–225.

[5] M. Campomanes, Y. Altintas, An Improved Time Domain Simulation for Dynamic Milling at Small Radial Immersions, Journal of Manufacturing Science Engineering 125 (3) (2003) 416–422.

[6] F. Taylor. “On the Art of Cutting Metals.” Transactions of the ASME, 28 (1906) 31–248.

[7] J. Tlusty Manufacturing processes and equipment. Upper Saddle River, NJ: Prentice-Hall; 2000.

[8] Y. Altintas, I. Yellowley, In process detection of tool failure in milling using cutting force models, ASME Journal of Engineering for Industry 111 (1989) 149–157.

[9] J. Tarn, M. Tomizuka, On-line monitoring of tool and cutting conditions in milling, ASME Journal of Engineering for Industry 111 (1989) 206–212.

[10] Y. Altintas, In-process detection of tool breakage’s using time series monitoring of cutting forces, International Journal of Machine Tools and Manufacture 28 (2) (1988) 157–172.

[11] Z. Deyuan, H. Yuntai, C. Dingchang, On-line detection of tool breakage’s using telemetering of cutting forces in milling, International Journal of Machine Tools and Manufacture 35 (1) (1995) 19–27.

[12] P. Huang, J. Chen, C. Chou, A statistical approach in detecting tool breakage in end milling operations, Journal of Industrial Technology 15 (3) (1999) 2–7.

[13] L. Lee, K. Lee, C. Gan, On the correlation between the dynamic cutting force and tool wear, International Journal of Machine Tools and Manufacture 29 (3) (1989) 295–303.

[14] A. Sarhan, R. Sayed, A. Nassr, R. El-Zahry, Interrelationships between cutting force variation and tool wear in end-milling, Journal of Materials Processing Technology 109 (2001) 229–235.

[15] D. Dimla Snr, P. Lister, On-line metal cutting tool condition monitoring—I: Force and vibration analyses, International Journal of Machine Tools and Manufacture 40 (5) (2000) 739–768.

[16] E. Dimla, Dimla Snr, Sensor signals for tool-wear monitoring in metal cutting operations—a review of methods, International Journal of Machine Tools and Manufacture 40 (2000) 1073–1098.

[17] Tool-life testing with single-point turning tools, ISO 3685:1993.

[18] T. Schmitz, B. Mann, Closed-form solutions for surface location error in milling, International Journal of Machine Tools and Manufacture 46 (2006) 1369–1377.

[19] M. Shaw Metal cutting principles. Oxford Science, Oxford, UK (1984).

[20] E. Trent, P. Wright Metal cutting. Butterworth–Heinemann, Boston, MA (2000).

[21] **Y.** Kevin Chou, Chris J. Evans, Tool wear mechanism in continuous cutting of hardened tool steels, Wear 212 (1) (1997) 59–65.

[22] G. Poulachon, A. Moisan, I. S. Jawahir, Tool-wear mechanisms in hard turning with polycrystalline cubic boron nitride tools, Wear 250 (1-12) (2001) 576–586.

[23] J. Barry, G. Byrne, Cutting tool wear in the machining of hardened steels: Part I: alumina/TiC cutting tool wear, Wear 247 (2) (2001) 139–15.

[24] Jie Gu, Gary Barber, Simon Tung, Ren-Jyh Gu, Tool life and wear mechanism of uncoated and coated milling inserts, Wear 225-229 (1) (1999) 273–284.

[25] H. Gekonde, S. Subramanian, Tribology of tool–chip interface and tool wear mechanisms, Surface and Coatings Technology 149 (2-3) (2002) 151–160.

[26] S Dolinšek, B Šuštaršič, J Kopač, Wear mechanisms of cutting tools in high-speed cutting processes, Wear 250 (1-12) (2001) 349–356.

[27] P. Koshy, R. Dewes, D. Aspinwall, High speed end milling of hardened AISI D2 tool steel ($\sim$58 HRC), Journal of Materials Processing Technology 127 (2) (2002) 266–273.

[28] C. Becze, P. Clayton, L. Chen, T. El-Wardany, M. Elbestawi, High-speed five-axis milling of hardened tool steel, International Journal of Machine Tools and Manufacture 40 (6) (2000) 869–885

[29] R.C Dewes, D.K Aspinwall, A review of ultra high speed milling of hardened steel. Journal of Materials Processing Technology, 69 (1-3) (1997) 1–17

[30] P. Prickett, C. Johns, An overview of approaches to end milling tool monitoring, International Journal of Machine Tools and Manufacture 39 (1999) 105–122

[31] A. Rehorn, J. Jiang,P. Orban, State-of-the-art methods and results in tool condition monitoring: a review, International Journal of Advanced Manufacturing Technology (2005) (26) 693–710.

[32] K. Lee, L. Lee, S. Teo, On-line tool wear monitoring using a PC, Journal of Material Processing Technology 29 (1992) 3–13.

[33] S. Choudhury, S. Rath, In-process tool wear estimation in milling using cutting force model, Journal of Materials Processing Technology 99 (2000) 113–119.

[34] Y. Cui, B. Fussel, R. Jerrard, J. Esterling, Tool wear monitoring for milling by tracking cutting force model coefficients, Transactions of NAMRI/SME 2009 (37) 613–20.

[35] Y. Altintas, E. Budak, Analytical Prediction of Stability Lobes in Milling, Annals of the CIRP, 44 (1) (1995) 357–362.

[36] M. Davies, B. Dutterer, J. Pratt, A. Schaut, J. Bryan, On the Dynamics of High-Speed Milling with Long, Slender Endmills, CIRP Annals - Manufacturing Technology 47 (1) (1998) 55–60.

[37] M. Davies, J. Pratt, B. Dutterer, Stability Prediction for Low Radial Immersion Milling, Journal of Manufacturing Science Engineering 124 (2) (2002) 217–226.

[38] M. Davies, J. Pratt, B. Dutterer, Stability of Low Radial Immersion Milling, CIRP Annals - Manufacturing Technology 49 (1) (2000) 37–40.

[39] J Gradisek, E Govekar, I Grabec, M Kalveram, On stability prediction for low radial immersion milling, Journal of Machine Science and Technology, 9 (2202) 117–130.

[40] S. Merdol, Y. Altintas, Multi Frequency Solution of Chatter Stability for Low Immersion Milling, Journal of Manufacturing Science Engineering 126 (3) 2004 459–466.

[41] T. Insperger, B. Mann, G. Stépán and P. Bayly, Stability of up-milling and down-milling, part 1: alternative analytical methods, International Journal of Machine Tools and Manufacture 43 (1) (2003) 25–34

[42] T. Insperger, B. Mann, G. Stépán, P. Bayly, Stability of up-milling and down-milling, part 1: experimental verification, International Journal of Machine Tools and Manufacture 43 (1) (2003) 35–40.

[43] T. Schmitz, M. Davies, K. Medicus, J. Snyder, Improving High-Speed Machining Material Removal Rates by Rapid Dynamic Analysis, CIRP Annals - Manufacturing Technology 50 (1) (2001) 263–268.

[44] T. Schmitz, K. Medicus, B. Dutterer, Exploring once per-revolution audio signal variance as a chatter indicator, Journal of Machining Science and Technology 6 (2) (2002) 215–233.

[45] R. Zapata, Applying decision analysis to milling with system dynamics constraints: A new frontier in machining science. Ph D. Dissertation, University of Florida 2009

[46] T. Schmitz, S. Smith, Machining Dynamics Frequency response to improved productivity, Springer

[47] E. Oberg Machinery's handbook : a reference book for the mechanical engineer, designer, manufacturing engineer, draftsman, toolmaker, and machinist, New York : Industrial Press, c2008.

[48] C. Cheng, Improved prediction of spindle-holder-tool frequency response functions. Ph.D. Dissertation, University of Florida; 2007.

[49] S. Smith, J. Tlusty, An overview of Modeling and Simulation of the Milling Process, Journal of Engineering for Industry, 113 (1991) 169–175

[50] Y. Altintas. Manufacturing automation. Cambridge, UK: Cambridge University Press; 2000.

BIOGRAPHICAL SKETCH

The author was born in the city of Mumbai (formerly known as Bombay), India in 1985. After spending his early years in Mumbai, he, along with his family, moved to the small town of Pune, India popularly known as the Oxford of the East. After finishing his high school education in 2003, he pursued his under-graduate degree in mechanical engineering at the University of Pune, India. After four grueling years of studies, project assignments and voluntary community service activities, the author obtained his bachelor's degree in mechanical engineering with distinction. He secured a job with Forbes Marshall India Pvt Ltd. a Company engaged in the manufacturing of control valves. After working with Forbes as a design engineer for a year, the author quit his job in order to pursue masters in mechanical engineering from University of Florida, Gainesville. The author plans to pursue PhD under the guidance of his advisor Dr. Tony L. Schmitz at the Machine Tool Research Center.