Automated Identification of Cutting Force Coefficients and Tool Dynamics on CNC Machines

by

Keith Dunwoody

B.A.Sc., The University of British Columbia, 2005

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

The Faculty of Graduate Studies

(Mechanical Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

March 2010

Abstract

The complexity and variation of parts are continuously increasing due to technologically oriented consumers. The objective of present manufacturing industry is to increase the quality while decreasing the machining costs. This thesis presents a smart machining strategy which allows the automated prediction of chatter-free cutting conditions using sensors integrated to Computer Numerical Controlled (CNC) machine tools.

The prediction of vibration free spindle speeds and depth of cuts require to have material's cutting force coefficient and frequency response function (FRF) of the machine at its tool tip. The cutting force coefficients are estimated from the cutting force measurements collected through dynamometers in laboratory environment. The thesis presents an alternative identification of tangential cutting force coefficient from average spindle power signals which are readily available on machine tools. When tangential, radial and axial cutting force coefficients are needed, the forces need to be collected by piezoelectric sensors embedded to mechanical structures. The structural dynamics of sensor housings distort the force measurements at high spindle speeds. A Kalman filter is designed to compensate the structural modes of the sensor assembly when the spindle speed and its harmonics resonate one of the modes the measuring system. The FRF of the system is measured by a computer controlled impact modal test unit which is integrated to CNC. The impact head is instrumented with a piezo force sensor, and the vibrations are measured with a capacitive displacement sensor. The spring loaded impact head is released by a DC solenoid controlled by the computer. The impact force and resulting tool vibrations are recorded in real time, and the FRF is estimated automatically. The measured FRF and cutting force coefficient estimated from the spindle power are later used to predict the chatter free depth of cuts and spindle speeds. The machine integrated, smart machining system allows the operator to automatically select the chatter-free cutting conditions, leading to improved quality and productivity.

Table of Contents

Abstract ii

Table of Contents iii

List of Figures v

1 Introduction 1

2 Background and Literature Review 3

2.1 Identification of Cutting Force Coefficients 3

2.2 Automatic Transfer Function Measurement 6

2.3 Effective Dynamometer Bandwidth Improvement 8

3 Identification of Cutting Force Coefficients from Spindle Power 10

3.1 Introduction 10

3.2 Extraction of Cutting Torque from Spindle Load Sensor 11

3.2.1 Limitation of Torque Estimation using Load Meter 14

3.3 Identification of Tangential Cutting Force Coefficient 16

4 Automatic Impact Modal Testing of Milling Cutters 19

4.1 Automatic Spindle Orientation 21

4.2 Automatic Capacitive Sensor Calibration 24

4.3 Signal Processing of Automatic Hammer Signal 25

4.3.1 Force Measurement Windowing 25

4.3.2 Spectral Averaging 28

4.4 Hammer-Displacement Sensor Offset Compensation 31

4.5 Comparison of Automatic Hammer with Traditional Impact Test 33
4.6 Conclusion 36

5 Effective Dynamometer Bandwidth Improvement 37
5.1 Introduction 37
5.2 Dynamic Model of Cutting Force Measurement System 38
5.3 Experimental Results 47

6 Integrated Smart Machining System 55
6.1 Integration with Cutting Coefficient Prediction 55
6.2 Integration with Automated Transfer Function Measurement Unit 57
6.3 Chatter Prediction 59
6.3.1 Chatter Stability Law 59
6.4 Prediction of Stability Charts 61

7 Summary and Future Research 64
7.1 Limitations and Future Work 64

Bibliography 66

Appendices

Automatic Hammer CNC Integration Source Code 69

List of Figures

2.1 Automatic Hammer[17] 7

3.1 Load Meter Voltage in Air Cutting 11

3.2 Load Meter Sensitivity by Spindle Speed 13

3.3 Load Meter Torque Estimate vs. Measured Torque 14

3.4 Torque in Air Cutting 16

3.5 Load Meter Sensitivity by Spindle Speed 17

4.1 Automatic Hammer 20

4.2 Measurement Heights for Finding Flutes 21

4.3 Cosine-Taper Window for Force Signal Windowing 26

4.4 Automatic Hammer 32

4.5 Comparison of Automatic Hammer vs. Traditional Tap Testing (Cross vs. Cross) 33

4.6 Comparison of Automatic Hammer vs. Traditional Tap Testing (Cross vs. Direct) 34

4.7 Comparison of Automatic Hammer vs Traditional Tap Testing (Compensated vs. Direct) 35

4.8 Predicted Stability of Slot Machining 35

5.1 Kalman Filter Block Diagram 47

5.2 Dynamometer Force-Force Transfer Function 49

5.3 Dynamometer Force-Force Transfer Function 49

5.4 Kalman Filter for X-Axis at Tooth-Passing Frequency of 930 Hz 50

5.5 Kalman Filter for Y-Axis at Tooth-Passing Frequency of 930 Hz 50

5.6 X-Axis Cutting Test at 13950 RPM 51
5.7 Y-Axis Cutting Test at 13950 RPM 52
5.8 X-Axis Cutting Force Range by Tooth Passing Frequency(10th Percentile to 90th Percentile) 53
5.9 Y-Axis Cutting Force Range by Tooth Passing Frequency(10th Percentile to 90th Percentile) 54

6.1 CNC and Automatic Hammer Connection to Laptop Computer 56
6.2 Comparison of Stability Chart with Different K_r 62
6.3 Comparison of Stability Chart with State of the Art Approach 63

Chapter 1

Introduction

Present computer numerical controlled (CNC) machine tools have integrated tool setting, in-situ inspection, positioning error compensation, vision and communication with remotely located maintenance computers for on-line troubleshooting. However, machining process control and monitoring tasks have not been widely implemented in industry, despite intensive research efforts since 1980s, due to poor reliability of algorithms and sensors. Industry removed most tool condition monitoring systems since they frequently failed to detect the tool wear and breakage, and stopped the machines which led to poor productivity. The recent advances in electronics, sensors and computer technology created new opportunities to add intelligence to CNC machine tools. However, any new on-line process monitoring and control task must be reliable and should not hinder the production flow. This thesis proposes an automated approach to one problem commonly encountered in industry: chatter prediction.

A common problem in industry is the chatter vibrations during milling and wear of the spindles. If the selected depth of cut and spindle speeds in the NC program are incorrect, the machine tool becomes unstable and experiences severe vibrations, which lead to tool failure, damage to the machine tool spindle and workpiece. The machine tool vibrations can be avoided by predicting the critical depth of cut and speeds, and choosing a spindle speed and depth of cut in the stable region. This stable region depends on the frequency response of the machine at the tool tip and cutting force coefficients of work material to be machined. Each frequency response function and cutting force coefficient are strongly dependent on the particular tool geometry and work material, and their preparation in the data base prior to production is rather a costly process, and require scientific expertise.

This thesis presents robust sensors and on-line algorithms to measure the material cutting coefficients and dynamic stiffness changes of the spindle. From this data, the chatter free, stable depth of cuts and speeds are identified automatically. The Frequency Response Function is measured with a computer controlled impact testing unit, and cutting force coefficients of materials are identified automatically from the spindle power sensor when activated by the operator on demand. A communication protocol between the CNC and an external PC is designed, which allows command and data exchange between the machine tool control (CNC) and measurement (PC) units. An automated impact hammer is integrated to CNC machine tool to measure FRF at the tool tip. The operator can activate the FRF measurement from the panel through computer commands. The tool is positioned automatically at the impact testing station placed at the corner of the machine tool table. The measurement unit is activated and FRF of the tool is measured automatically. The operator is allowed to conduct trial test cuts with the same tool and specific work material. The spindle power is collected and sent to measurement computer. The power data, which is calibrated against cutting forces and torque, is used to identify cutting force coefficients automatically. The measured FRF and cutting force coefficients are used to predict chatter stability charts, which enable the operator to select optimal depth of cut and spindle speeds without violating the maximum torque and power limit of the machine tool.

Henceforth, the thesis is organized as follows. Chapter 2 presents related literature about on-line machine tool monitoring, control and chatter prevention methods. The estimation of cutting forces and cutting force coefficients from the machine spindle load meter is presented in Chapter 3. The integration of automated impact hammer to CNC machine tool and its reliability are explained in Chapter 4. The cutting force measurement technique developed in Chapter 3 are sufficient for measuring the average torque on the spindle, but cannot be used to reliably measure the instantaneous torque, or the cutting forces in the other two directions. If these forces are required, a dynamometer must be used. However, dynamometers distort the cutting forces if the spindle speed or tooth passing frequency is close to a mode of the dynamometer. This can occur during high-speed machining. Chapter 5 presents a method for compensating the dynamics of a dynamometer to make them more useful during high speed machining. The impact test and cutting force coefficients are integrated to a previously developed chatter stability and machine tool diagnostic system in Chapter 6. The thesis is concluded with a summary and future research issues in Chapter 7.

Chapter 2

Background and Literature Review

As machined parts become more complicated and the time given to develop them shortens, it becomes more important to decrease the length of the the machining process time of a part. Smart machining systems aim to improve productivity and part machining quality by integrating sensors to the machine tool, combined with physics based predictive models to provide actionable information to the machine tool operator. This allows a smart machining system to provide the first and every part correct; improve the response of the production system; realize rapid manufacturing; and provide data on an as-needed basis[8].

The objective of this thesis is to develop a system which can be integrated to machining centers to automatically determine chatter free, stable cutting conditions for a machining operation. The cutting coefficients, which relate feed per toot to the cutting force need to be determined. The frequency response function at the tip of the cutting tool must be measured, and this information is used with a model of the chatter process to determine the chatter-free cutting conditions.

A method of extending the effective bandwidth of dynamometers is also examined. Although dynamometers are not practical in the average machine shop, this technique allows less costly dynamometers to be used in laboratories to measure cutting forces at high spindle speed.

2.1 Identification of Cutting Force Coefficients

In order to predict the chatter-free pockets in a machining operation it is necessary to identify the cutting forces which will be generated for a given depth of cut (a). The tangential (F_t) and radial (F_r) cutting forces in milling are expressed as

$$F_t = K_{tc}ah(\phi) + K_{te}aF_r = K_{rc}ah(\phi) + K_{re}a \tag{2.1}$$

where (K_{tc}, K_{rc}) are the cutting force coefficients, and (K_{te}, K_{re}) are the edge force coefficients in tangential and radial directions, respectively. The chip thickness change periodically as $h = c \sin \phi$ where c is the feed per tooth, and ϕ is the immersion angle of the cutter which changes with the spindle speed (Ω)

$$\phi = \Omega t \tag{2.2}$$

Note that the chip is zero when the cutter is outside the material, $h = 0$ if $\phi < \phi_{st}$ or $\phi > \phi_{ex}$).

The cutting force coefficients are ideally identified from direct cutting force measurements carried out with dynamometers.

Tool wear can be correlated to the cutting force since as the cutting tool wear, the cutting force increases due to a larger friction load. There are different approaches used in force measurement for TCM: Direct measurement using a dynamometer, force plates and rings, force-measuring bearings, force and torque at spindles, pins and extension sensors, displacement sensors, and motor current and effective power.

Direct measurement using a piezoelectric dynamometer is the simplest method of force measurement and has a high bandwidth. However dynamometers are unsuitable for use in production due to high cost and a lack of overload-protection in case of a collision. Force plates and rings use piezoelectric sensors mounted between elements of the turret (in the case of turning machines), or behind the spindle flange (in the case of milling machines). So far results have been poor due to a high-degree of cross-talk, and force distortion due to the dynamics of the machine which may change over time. Some work has been done to compensate the machine dynamics in such systems using a Kalman filter[19]. Park et al. later extended the Kalman Filter compensation strategy to micro milling systems[6]. Spindle bearings are instrumented with strain gauges to measure the force. The output of these sensors must be low-pass filtered to eliminate the ball contact frequency. The torque can also be measured through a sensor placed directly in the machine tool spindle or tool holder. However, a complex system is required to measure the torque through the entire power range of the tool, and the measurement must be transmitted from the rotating spindle[5].

Pins and extension sensors measure the cutting force indirectly through the extension of machine elements. These have low sensitivity, so are not suitable for the task of measuring cutting coefficients. Similarly cutting force can also be measured indirectly by measuring the bending of the tool through non-contact displacement sensors. Such sensors have a high risk of interference in the form of chips, dirt and cooling fluid. In a similar vein, strain gauges have been mounted into a turning tool to measure the cutting forces[22]. This system had good sensitivity, but the measured force was a combination of the tangential force, feed force and radial force. This makes it suitable for tool condition monitoring, where the change in force is important, but unsuitable for measuring cutting coefficients, where the magnitude of each component is important.

The final method is to estimate the force from the current or power applied to the spindle or axis motors. This will be discussed in more depth below, however all such systems share some common disadvantages. Forces measured in this way include bearing and guideway friction components, which are often much higher than the cutting forces and may vary substantially based on the lubrication state and motor speed. Also, due to the large masses the output signal has a limited frequency bandwidth.

The cutting forces can be measured from the motor current[1], [16]. The feed and spindle drive servos have low frequency bandwidth and may not accurately capture cutting forces which are periodic at the spindle speeds times number of teeth on the cutter. However, the cutting force coefficients can be extracted from average cutting forces, which can be correlated to average current drawn by the spindle or motor current sensors. The prediction accuracy would improve if the friction force and inertial forces can be accurately modeled, and separated from the current consumed by cutting process when the machine has time varying velocities. Furthermore, this method is poor at determining the force in a direction if that axis is not moving due to the uncertainty in the static friction force, although empirical models have been developed for this case[14].

Spindle motor power[7] and spindle motor current[18] have been used to estimate the cutting torque. The bandwidth of the spindle motor is lower than the bandwidth of the axis feed drives. However, the friction which must be compensated is also lower. Since the measurement will be averaged to determine the cutting coefficient, the bandwidth of the sensor is less important than an accurate average measurement. This thesis will use spindle motor power to estimate the tangential cutting force coefficient.

The total torque drawn from the spindle motor is

$$T_c = \sum_n \frac{D}{2} F_{t,n}, \tag{2.3}$$

where D is the cutter diameter and $F_{t,n}$ is the cutting force of the nth tooth. The total power drawn by the spindle motor is

$$P_c = T_c \pi D \Omega \tag{2.4}$$

By measuring the average motor power ($\bar{P}_c$), the tangential cutting force coefficients (K_{tc}, K_{te}) can be identified during cutting with know depth of cut, speed and radial engagement condition. The radial cutting force coefficient (K_{rc}, K_{re}) cannot be predicted from the spindle power since the spindle power is related only to tangential force (i.e. torque). However, the radial and axial cutting force coefficients do not affect the absolute stability limits but mainly the width of the stability pockets.

Determining the instantaneous cutting forces from power measurements is almost impossible at practical spindle speeds due to low bandwidth of the drives. Instead, this thesis proposes to use average power to predict the fundamental parameter, the tangential cutting force coefficients, from the average motor power, which is available in CNC machines. The instantaneous cutting force can be reconstructed from the tangential cutting coefficient, feed per tooth and depth of cut.

2.2 Automatic Transfer Function Measurement

The principal methods of measuring the force to displacement frequency response function (FRF) measurement use instrumented hammers and shakers to apply a force and measure the response with an accelerometer or displacement sensor[9]. These techniques require lengthy manual setup and cannot be used directly in a smart machining system.

Several techniques of performing automatic frequency response or impact testing have been investigated. Piezoelectric sensors have been embedded into a structure to measure its frequency response function[12]. This technique cannot be used to directly measure the

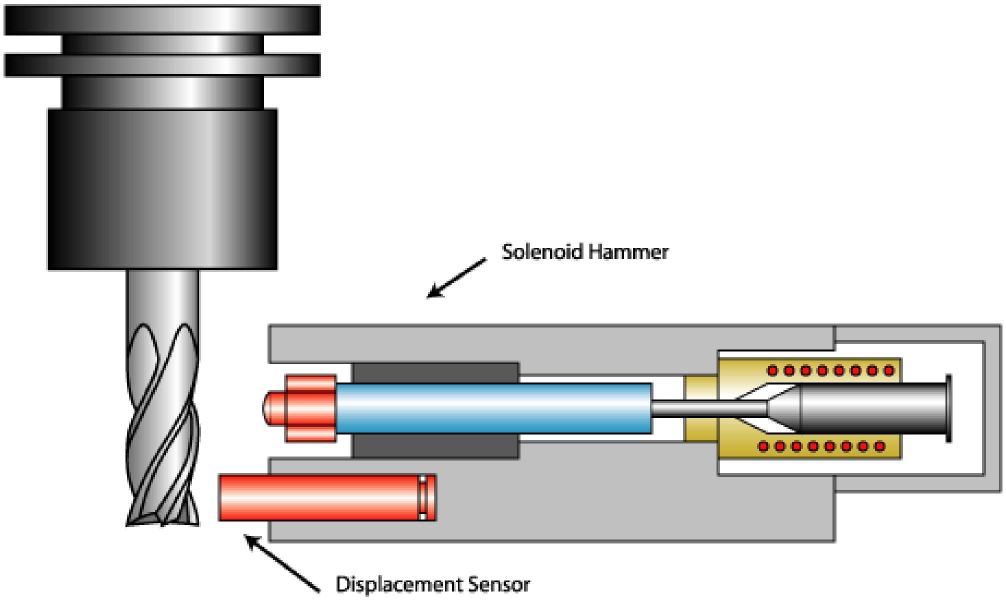

Figure 2.1: Automatic Hammer[17]

frequency response function to a force applied at a free surface like the tool tip. It could be used to find the natural frequencies of the spindle.

DC solenoid[20] and piezoelectric[23] powered impact hammers have been used in wave-velocity measurement of testing. For concrete velocity wave testing it is not necessary to measure the impact force; the DC solenoid did not have a force sensor. The piezoelectric sensor had a force sensor to measure the time of impact accurately. In this case filtering and windowing was needed to eliminate extraneous peaks which were observed to either side of the primary impact.

Bilne and Tse developed an solenoid-driven hammer[17] to measure the transfer function near the tip of a cutting tool (see figure 4.4). The hammer is activated by a pulse from an analog output on a computer. When the hammer is activated, the solenoid is energized, and the hammer strikes the tool. Once the hammer strikes the tool, the control electronics immediately de-energize the solenoid and the hammer is returned to the rest position by a spring. The force applied to the tool is measured by a piezoelectric force sensor mounted on the hammer, and the response is measured by the capacitive displacement sensor mounted below the hammer.

The advantages of this system over a traditional manual hammer are that it can deliver quick, repeatable strikes to the tool. The integrated displacement sensor allows the system to be run without requiring an external response sensor, such as an accelerometer. This hammer is used as the basis for the automatic frequency response measurement system described in Chapter 4.

2.3 Effective Dynamometer Bandwidth Improvement

The cutting forces are either measured by using table dynamometers or spindle integrated sensing systems, where both use piezoelectric sensors. Piezoelectric sensors detect the strain between the two mechanical interface surfaces, and naturally reflect the mass, springs and damping element chains in the embedded mechanical structure. As a result, the bandwidth of the force sensing systems is limited by the natural modes of the structure which can be well below the tooth passing frequencies used in milling operations. The aim of the smart measurement system is to remove the dynamic distortions from the measured force signals, so that the cutting forces and force coefficients are more accurately identified.

Park[19] developed a spindle integrated force sensor system for measuring the cutting forces at the tool tip by measuring the force sensors embedded in the spindle housing. He then used a Kalman filter to compensate for the dynamics of the spindle and spindle housing.He was able to expand the bandwidth of the spindle integrated sensing system to 1000 Hz by compensating the first three bending modes of the spindle structure. He later expanded the algorithms to compensate high frequency dynamics of micro end mills operating over 50,000 rev/min[6].

This thesis presents a robust Kalman filter method which is able to compensate the most dominant mode of the structure which is carrying the sensor. The method is illustrated on a table dynamometer which is used to measure cutting forces during high spindle speeds (i.e. 15,000 rev/min).

Although, the tool condition monitoring is quite a wide field, this thesis focuses only on automated FRF measurement, cutting force measurements at high speeds, and identification of cutting force coefficients from power signals available in CNC system. These methods are robust, practical and do not hinder the operation of the machine during production, hence they contribute to smart machining technology.

Chapter 3

Identification of Cutting Force Coefficients from Spindle Power

3.1 Introduction

The cutting force coefficients are used to predict cutting forces, torque, power and chatter stability to identify optimal machining operations on the machine tools. The cutting force coefficients are dependent on the material properties such as yield strength and hardness, and have traditionally been identified in laboratories by measuring cutting forces using dynamometers. The cutting forces in X, Y, and Z directions are measured with a dynamometer, and from the workpice material-tool geometry specific cutting coefficients in the radial, tangential and axial directions are calculated.

This thesis presents a method of predicting one of the cutting force coefficients from average power measurement sensors available on standard machine tools. Several test cuts are performed by the machine tool operator, and the power drawn by the spindle motor is measured using a computer connected to the CNC (the details of this connection are further explained in Chapter 6). The cutting torque is calculated from the spindle power. By combining the cutting torques calculated during several test cuts at different feed rates, the mechanistic cutting coefficients can be calculated, and entered into a cutting coefficient database. This eliminates the need for costly tests in a laboratory with a dynamometer.

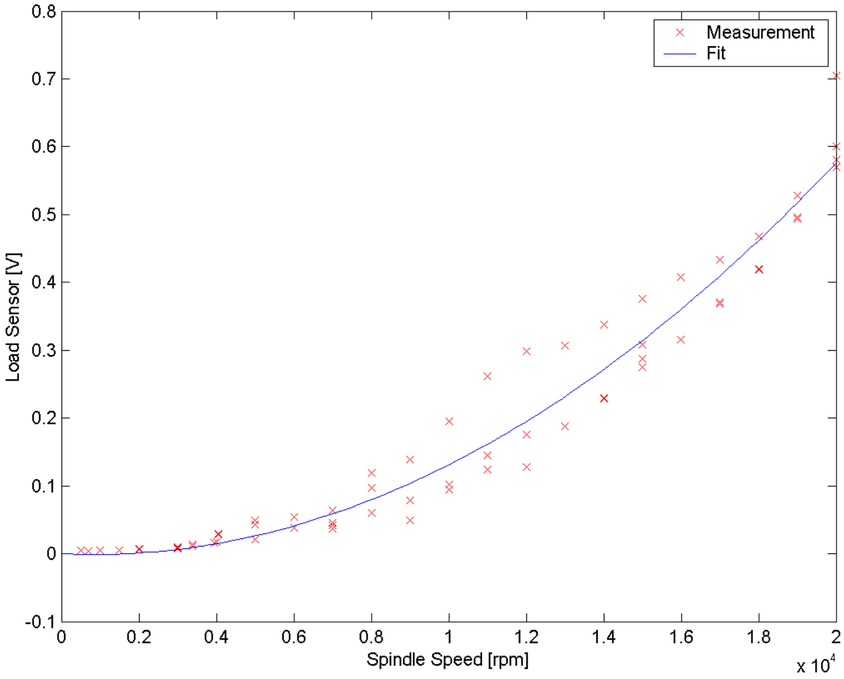

Figure 3.1: Load Meter Voltage in Air Cutting

3.2 Extraction of Cutting Torque from Spindle Load Sensor

The first step in cutting coefficient identification is to measure the cutting forces during a test cut. Most machines include load meters which display the power being drawn by the machine spindle, so the operator can know if the machine is approaching an overload condition. This voltage output of this sensor can be read by a computer connected to the CNC and used to estimate the torque on the machine tool, which provides enough information to estimate one of the three mechanistic cutting coefficients.

The load meter on the machine tool returns a voltage V which is proportional to the average power being drawn by the spindle drive motor

$$\begin{aligned} V &= KP, \\ &= KT\omega \end{aligned} \tag{3.1}$$

where T is the total torque applied to the spindle by the motor, ω is the angular speed of the spindle, and K is a constant. The applied torque consists of the torque T_c applied to cutting, and the torque lost to friction T_f

$$T = T_c + T_f. \tag{3.2}$$

The friction is assumed to consist of viscous and Coulomb components

$$T_f = b\omega + T_{cf}, \tag{3.3}$$

where b is the viscous friction constant and T_{cf} is the Coulomb friction torque.

Substituting equation 3.3 into 3.1 gives

$$V = K\omega \left(T_c + b\omega + T_{cf} \right). \tag{3.4}$$

Rearranging this equation gives the cutting torque in terms of load meter voltage and spindle speed

$$T_c(V, \omega) = \frac{V}{K\omega} - b\omega - T_{cf}. \tag{3.5}$$

There are three machine-tool specific coefficients which must be identified: the viscous friction b, the Coulomb friction T_{cf} and the load meter constant K.

To identify these coefficients, air cutting tests were performed on the target machine, a Mori-Seiki NMV-5000 machine tool. For an air-cutting test, the cutting torque $T_c = 0$, so

$$V = Kb\omega^2 + KT_{cf}\omega. \tag{3.6}$$

Air cutting tests were performed from 1000 to 20000 Hz, and the voltage measured by the spindle load meter was plotted (figure 3.1). From this data the voltage equivalent of the viscous and Coulomb friction

$$V_b = Kb \tag{3.7}$$

$$V_{Tf} = KT_{cf} \tag{3.8}$$

can be determined through least-squares fitting. For the Mori-Seiki machine tool tested, these parameters were found to be $Kb = 1.43 \times 10^{-7}$ V/rad/s^2, and $KT_{cf} = -2.51 \times 10^{-5}$ V/rad/s. Although the Coulomb friction is negative, this is not a problem for cutting torque prediction

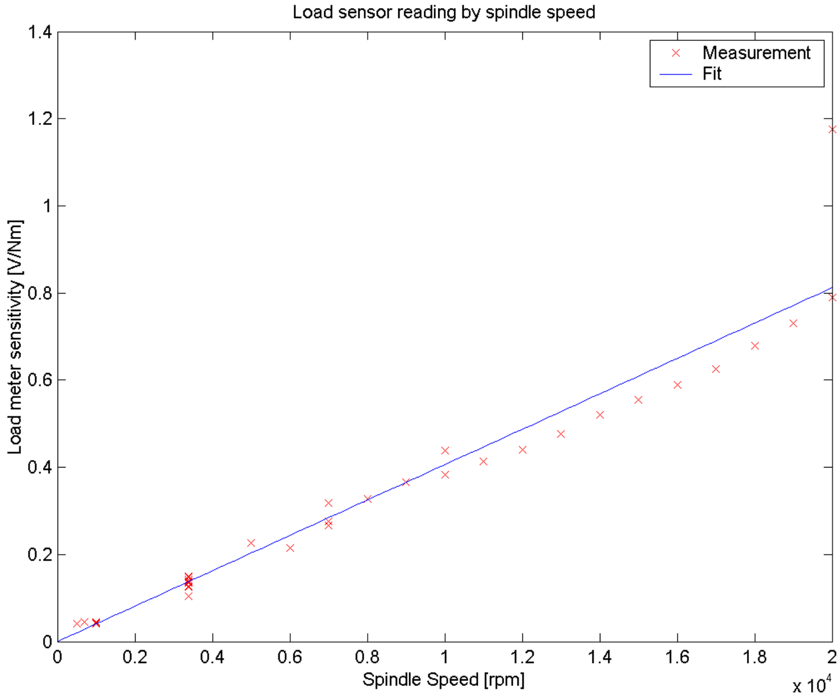

Figure 3.2: Load Meter Sensitivity by Spindle Speed

in practice (see section 3.2.1). To avoid the unphysical result of negative friction, the friction force is assumed to be zero below 2000rpm (209 rad/s).

To measure the load meter coefficient K, slot milling tests were conducted with spindle speeds starting at 1000 rpm in 1000 rpm increments up to the maximum spindle speed of the machine tool, $20,000$ rpm. The true average cutting torque (T_c) was measured using Kistler rotating dynamometer, while the average power was obtained from the spindle load meter of the machine. The spindle load caused by friction,

$$V_f = Kb\omega^2 + T_{cf}\omega, \tag{3.9}$$

was subtracted from the measured load, and the ratio of the cutting load to cutting torque torque

$$K\omega = \frac{V - V_f}{T_c} \tag{3.10}$$

was plotted in figure 3.2. The coefficient K was estimated by fitting a straight line to the data using least squares. For the Mori-Seiki NMV-5000 machine tool tested, K was found to be 3.88×10^{-4} V/W. By substituting K into KT_{cf} and Kb, the Coulomb and viscous

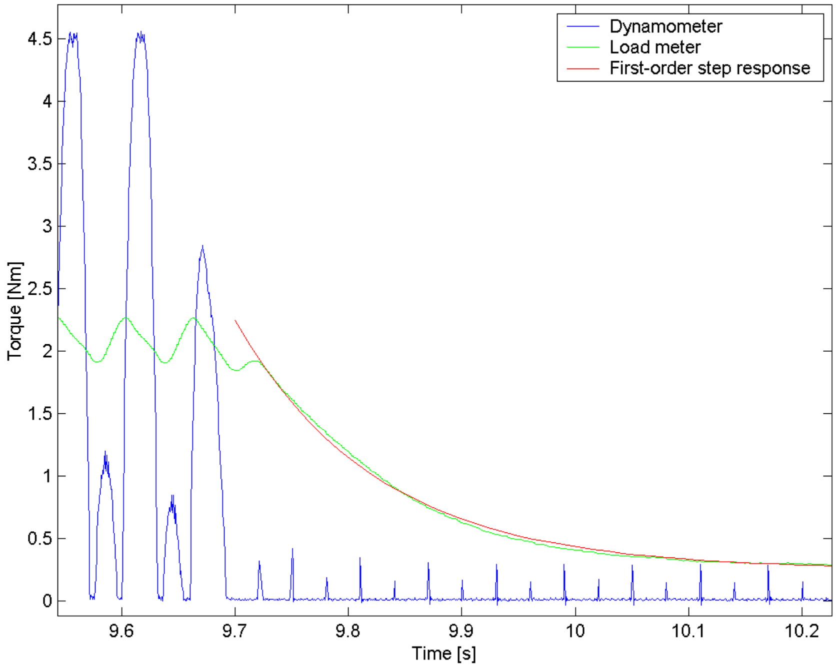

Figure 3.3: Load Meter Torque Estimate vs. Measured Torque

friction constants are found to be

$$\left.\begin{aligned} T_{cf} &= -0.0647 \text{ Nm} \\ b &= 3.70 \times 10^{-4} \text{ Nm/(rad/s)} \end{aligned}\right\} \omega > 209 \text{ rad/s, or } 2000 \text{ rpm} \tag{3.11}$$

3.2.1 Limitation of Torque Estimation using Load Meter

This method of estimating cutting torque has a couple of limitations stemming from the nature of the load meter sensor, which must be considered when planning cutting force coefficient identification tests. First, the data from the load meter is low-pass filtered, so this technique can only be used to obtain the average cutting torque, not the instantaneous cutting torque. Second, since the load meter measures power, the cutting force estimate is inaccurate at low speeds. Third, the friction estimate is inaccurate at low speeds.

The primary purpose of the machine tool load meter is to provide an indication to the operator when the machine is operating within safe limits. For this purpose, the important measure is the average power being applied to the spindle motor. In addition, the spindle has a large moment of inertia, and the bandwidth of the spindle velocity control loop is low, which means that the power reported by the load meter is a low-pass filtered version of the instantaneous cutting power. This can be seen in figure 3.3. This shows the torque and load-meter readings as the cutting tool leaves the workpiece. The torque measured by the dynamometer falls to zero immediately, while the load meter reading falls off exponentially. From this graph it appears the load-meter has approximately a first order response. To discover the approximate bandwidth of the load meter, the step response of a first order system was fitted to the load meter data. This gives a bandwidth of approximately 7 Hz for the load meter. To have a tooth-passing frequency less of less than 7 Hz, a two fluted tool must be rotating at less than 210 rpm. This is much lower than the practical spindle speeds on modern milling machines, making this method impractical for measuring the instantaneous cutting torques. It can still be used for measuring average cutting forces.

Another limitation arises because the torque is being estimated from the measured power. At low speeds, the power will be low, even if the torque is high, leading to inaccurate measurements. Consider two load meter measurements, one at V, and one at $V + \Delta V$, where ΔV is the quantization error of the load meter, which for the Mori-Seiki NMV-5000 is approximately 3×10^{-4} V. The difference in estimated torque for these two measurements,

$$\Delta T_c = \frac{1}{K\omega}\Delta V, \tag{3.12}$$

is the minimum cutting torque error. For the cutters used in this thesis, the average cutting torque was between $2-3$ Nm, so to get the quantization error below 10%, the spindle speed must be above 35 rpm, and to get the quantization error below 1% the spindle speed must be above 350 rpm.

The apparent negative Coulomb friction is the linear approximation used in the model. The real friction follows the Stribek curve, with viscous friction dominating over the majority of the spindle speeds except under 2000 rpm as can be seen from the torque measurements against the spindle speed shown in figure 3.4. To avoid the unphysical negative friction, friction is assumed to be zero below 2000 rpm. Since the spindle load method is less accurate at low spindle speed, so the error in the friction model at low spindle speed is unimportant in practice.

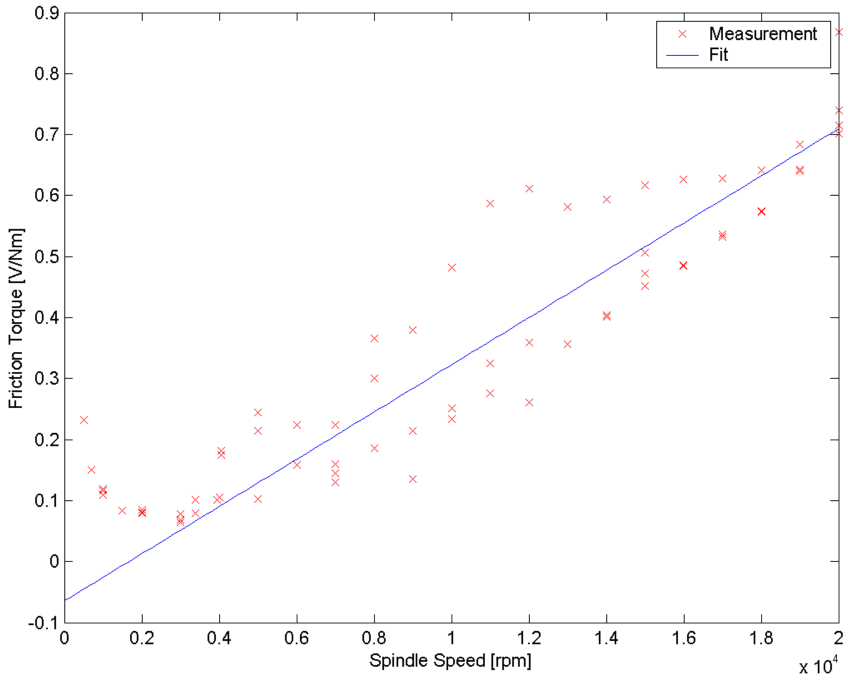

Figure 3.4: Torque in Air Cutting

3.3 Identification of Tangential Cutting Force Coefficient

The estimated average cutting torque from the spindle load meter needs to be correlated to cutting torque model based on metal cutting mechanics.

The tangential cutting force per unit depth of cut along the cutting edge j of a milling cutter is given by

$$dF_{t,j}(\phi) = K_{tc}h(\phi) + K_{te}, \tag{3.13}$$

where $h(\phi)$ is the chip thickness at the cutting edge angle ϕ[2]. The corresponding instantaneous cutting torque is

$$dT_{c,j}(\phi) = R(K_{tc}h(\phi) + K_{te})g(\phi), \tag{3.14}$$

where R is the radius of the milling tool, and $g(\phi)$ is a function which is unity if the tooth is in cut, and zero if it is out of cut. The chip thickness changes as cutter rotates,

$$h(\phi) = s_t \sin(\phi). \tag{3.15}$$

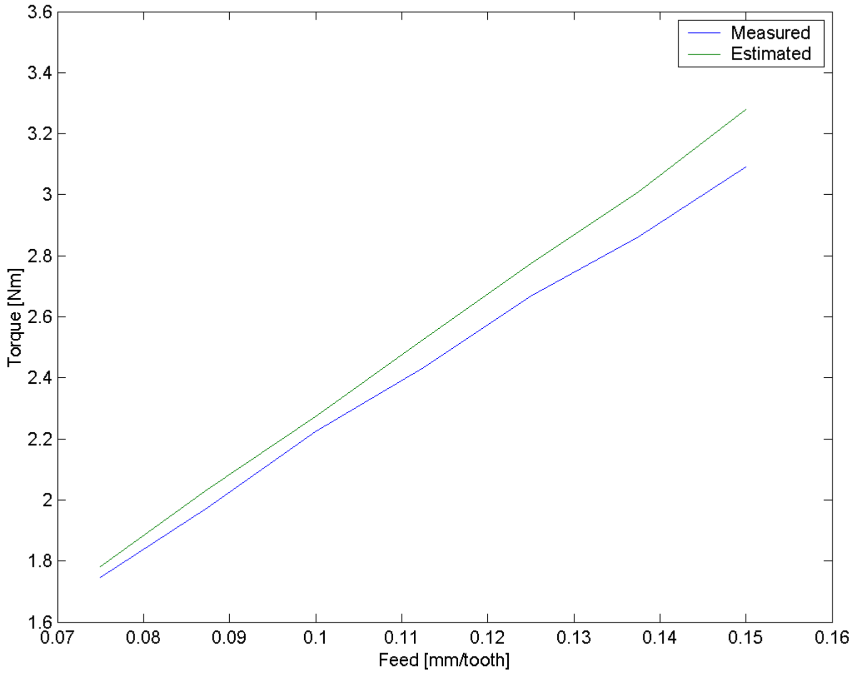

Figure 3.5: Load Meter Sensitivity by Spindle Speed

The average torque per unit depth of cut is thus

$$d\bar{T}_{c,j} = \frac{R}{2\pi}\int_0^{2\pi} g(\phi)\left[K_{tc}s_t\sin(\phi) + K_{te}d\phi\right]. \tag{3.16}$$

If the cutting tests are performed in slotting mode for simplicity,

$$g(\phi) = \begin{cases} 1 & \text{if } 0 \leq \phi < \pi \\ 0 & \text{if } \pi \leq \phi < 2\pi \end{cases}. \tag{3.17}$$

the average torque per unit length for a single tooth reduces to

$$d\bar{T}_{c,j} = \frac{RK_{tc}}{\pi}s_t + \frac{RK_{te}}{2}, \tag{3.18}$$

or a total average torque of

$$\bar{T}_c = \frac{RNaK_{tc}}{\pi}s_t + \frac{RNaK_{te}}{2}. \tag{3.19}$$

where a is the axial depth of cut.

Cutting tests were performed with varying feeds per tooth s_t, and the measured and estimated cutting torque was shown in figure 3.5. Straight lines were fit to both the measured torque from dynamometer and estimated torque from the load, and the slope and intercepts were matched to the coefficients in equation 3.19. The cutting coefficients were estimated to be

$$K_{tc} = 1040\text{N/mm}^2 \tag{3.20}$$

$$K_{te} = 9.81\text{N/mm} \tag{3.21}$$

from the load meter measurements. The cutting coefficients obtained from the rotating dynamometer measurements were

$$K_{tc} = 936\text{N/mm}^2 \tag{3.22}$$

$$K_{te} = 13.9\text{N/mm}, \tag{3.23}$$

which gives a value of K_{tc} which agrees within 11% with the estimation from the load meter model. The prediction accuracy is sufficient to use the model for on-line identification of tangential cutting force coefficients from cutting tests using the available load meters on the production machines.

The cutting force identification model can be integrated to CNC system for smart, automated, on line design of material machining data base in production floors.

Chapter 4

Automatic Impact Modal Testing of Milling Cutters

To predict the occurrence of unwanted unstable vibrations in machining (known as chatter), the dynamics, or frequency response function (FRF) of the machine tool must be measured. In practice this can be a time consuming and difficult operation, which requires a trained technician or engineer with specialized test equipment. This test must be repeated for every cutting tool used in manufacturing a part as each cutting tool will have a different FRF. If the machine tool could measure its own FRF, the process of calculating the stable cutting regions of a machine tool would be greatly reduced.

In order to automate this process an automatic hammer was developed which can be integrated to the controller of a machine tool. The automatic hammer can be activated from the front panel of the CNC by the operators. The chatter stability charts can be automatically calculated from the FRF measurements, and the operators can be advised to select chatter free conditions and the current health of the spindle can be reported to avoid damages to bearings.

The automatic hammer is a device invented by Matthew Tse and Ryan Bilne for their Master of Engineering project (2003)[17], with the goal of automating the process of obtaining the frequency response function of a tool/tool-holder combination. Although their device simplifies the measurement of the frequency response, the device itself is not enough to achieve the goal of one touch FRF measurement. This chapter expands their work to integrate the hammer with a CNC to produce a system which can perform high quality transfer function measurement completely automatically.

The automatic hammer (shown in figure 4.1) consists of a solenoid actuated plunger with a force sensor on the end to hit the tool, and a capacitive displacement sensor to measure

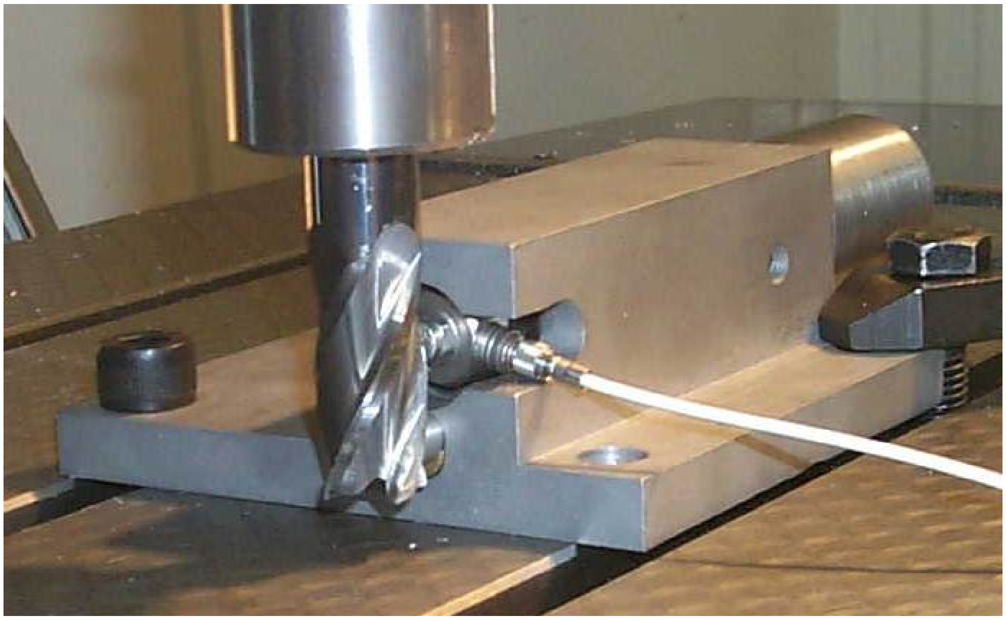

Figure 4.1: Automatic Hammer

the response. The force sensor and the solenoid are connected to a solenoid control circuitry which activates the solenoid in response to a signal transmitted from a connected PC. The circuitry energizes the solenoid until the hammer head on the solenoid hits the tool. The force and displacement are recorded by a computer, and the FRF between force and displacement is calculated.

Several difficulties have been identified with the use of automatic hammer on CNC machine tool:

- the tool must be oriented at the correct position and orientation,
- the capacitive displacement sensor must re-calibrated with each measurement,
- multiple impacts which destroy the measurement accuracy need to be minimized, and
- the offset between the impact force and measurement of vibration points must be considered.

This work proposes a series of algorithms which can be implemented on a personal computer (PC) connected to a CNC machine tool to automatically position the cutting tool and calibrate the displacement sensor. The accuracy of the developed system is then tested using a normal laptop computer connected to a Mori-Seiki NMV-5000 5-axis milling machine.

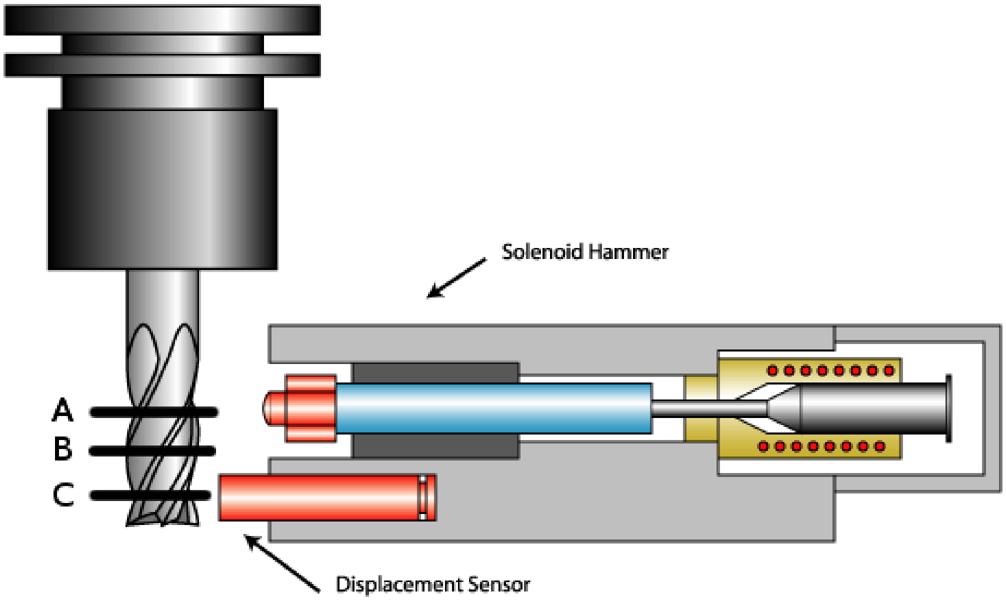

Figure 4.2: Measurement Heights for Finding Flutes

4.1 Automatic Spindle Orientation

In order to get an acceptable impact on the tool, it must be aligned so that the hammer strikes a surface of the tool which is approximately perpendicular to the direction of travel of the hammer, and the path of the hammer is free of interference from other parts of the tool. On an inserted cutter, this is not difficult since there is typically a large flat surface located behind the inserts. On a traditional endmill however the only acceptable location is behind the flute edge.

To position the tool correctly, the location of a flute at the height of hammer the must be found (marked A on figure 4.2). However, the displacement sensor cannot be moved to this location, since the tool would impact against the table. Instead, the flute is located at the bottom of the tool (C), and midway between A and C (B). Assuming the helix angle is constant, the location of the flute at point θ_A is

$$\theta_A = 2\theta_B - \theta_C, \tag{4.1}$$

where θ_B and θ_c are the locations of the flute at points B and C, respectively.

Locating the flute is a minimization problem; the flute is at the angle where the distance between the displacement sensor and the tool is minimized. To find the location of a flute, three points are found: a central point, where the sensor is close to the flute, and a point to the left and a point to the right of the central point, where the tool is further away from the sensor. Using the golden section search algorithm described below, the left, center and right points are moved towards each other, making sure that tool is always closer to the sensor at the central point than either the left or right points. This ensures that the tool is closest to the sensor at some point between the left and right points. Once the left and right points are within 1.5° of each other, the algorithm is terminated. This bound offers sufficient accuracy so the hammer will hit the flute. Additional accuracy is not required, and a smaller bound would require more measurements, increasing the length of time required to run the algorithm.

The technique used to locate the flute is the Golden Section Search[15]. This algorithm keeps track of three points: the left bound A, right bound B, and a point C between the two. A test point X is chosen in the larger of the regions between A and C, and between C and B. Suppose the point X is chosen between A and C. Then if $d(X) \leq d(C)$, the new bracketing triplet will be A, X, and C. Alternatively, if $d(C) < d(X)$, then the new bracket will be X, C, B.

The test point X is chosen to be a fraction of the distance between A and B as follows. Suppose C is a fraction of the distance between A and B

$$\frac{C-A}{B-A} = W. \tag{4.2}$$

Furthermore, suppose that the test point X is a fraction beyond A of

$$\frac{X-A}{B-A} = Z. \tag{4.3}$$

The next bracket will either be of length W, or of length $1-Z$. To maximize the worst-case performance, these two must be equal, or

$$W = 1 - Z. \tag{4.4}$$

Now assume the initial points A, B, and C were selected by the same strategy. Then W must be the same proportion of the whole interval, as Z is of the smaller interval $\overline{AB}$

$$W = \frac{Z}{W}. \tag{4.5}$$

Substituting in equation 4.4 and solving gives

$$W = \frac{1}{\phi} = \frac{\sqrt{5} - 1}{2}, \tag{4.6}$$

where the negative solution is ignored as physically impossible. Here ϕ is the golden ratio.

In order to apply this algorithm, the initial three bracket points must be found. The algorithm given in Numerical Recipes in C[21] is adopted here. First, the distance to the tool at two points θ_a, $\theta_b = \theta_a + 5.6°$. The largest of these two points becomes θ_1, the other θ_2. These two points are measured, and the distance to the tool at some number of other points

$$\theta_n = \theta_{n-1} + \begin{cases} \phi\left(\theta_{n-1} - \theta_{n-2}\right) \text{if } \phi\left(\theta_{n-1} - \theta_{n-2}\right) < 22.5° \\ 22.5° \text{otherwise} \end{cases}, \tag{4.7}$$

are also measured, for $n = 2, 3, \ldots$. This continues until the distance to the tool at the final point θ_N is greater than the distance at θ_{N-1}. Then, the three bracketing points will be θ_{N-2}, θ_{N-1}, and θ_N.

If the bracketing interval is chosen before the maximum step size of 22.5° is reached, the central point will be located so that the golden section algorithm is optimal. If the maximum step size of 22.5° is reached, the interval found will no longer be optimal for the golden section search; however a maximum step size is required because the golden section search algorithm assumes there is only one local maximum inside the bracketed region. If there are more than one local maximum, the algorithm could converge to a local maximum which is not necessarily the global maximum. In order to ensure there will be only one local maximum in the bracket, it is necessary to find the maximum bracket size such that there is only one local maximum in any such bracket.

The maximum step size must be carefully chosen. On one hand, if the maximum step size is too small, the bracketing algorithm will often achieve the maximum step size, which will slow down both the bracketing and the minimization algorithms. On the other hand, if the maximum step size is too large, the chosen bracket may include multiple local maxima, causing the algorithm to fail to correctly locate the flute edge.

4.2 Automatic Capacitive Sensor Calibration

The calibration for the capacitive sensor is only valid when the target is a thick metal sheet that is flat within an area roughly 30% larger than the target of the capacitive sensor. Due to the flutes and the curvature of the tool, a milling tool does not satisfy these requirements, so a custom calibration method must be done before measuring each tool. In addition, the calibration is designed such that at the end of the calibration the tool is positioned such that the voltage output from the capacitive sensor is approximately zero. The capacitive sensor has a non-linear sensitivity, and is more sensitive the closer the tool is to the sensor, so positioning the tool such that the voltage output of the displacement sensor is zero ensures maximum sensitivity while ensuring the tool does not impact against the sensor.

In order to calibrate and zero the sensor, the voltage at the initial distance is first measured. The machine is then jogged 50 μm, and the voltage is measured again. This is used to obtain an estimate of the sensitivity

$$S = \frac{\Delta V}{\Delta x}. \tag{4.8}$$

Then an estimate of the distance required to zero the voltage of the sensor

$$\Delta x_{remaining} = \frac{V}{S}. \tag{4.9}$$

If the distance remaining is less than 200 μm, then the CNC is jogged half the estimated remaining distance; otherwise it is jogged 100 μm. This slow approach ensures that the tool does not impact against the displacement sensor. This is repeated until the voltage is more than -0.1 V. The last sensitivity measurement is taken as the sensitivity of the sensor for the FRF measurement.

There are two major potential sources of error in this measurement. First, the CNC may have low positioning accuracy, so it may move a different amount than what is instructed. Secondly, there may be noise in the displacement sensor measurement. This noise was found to be negligible in the flute finding algorithm; however, the sensitivity is the derivative of the data, which will tend to magnify the effect of noise. In order to measure the error of the CNC axis, a laser interferometer was set up to measure the displacement along the same axis as the automatic hammer was measuring. The CNC was then programmed to move between

10μm and 100μm, and the distance actually traveled was measured by the interferometer. In all cases, it was found that the distance actually traveled by the CNC agreed with the instructed distance to within ±100nm, which is the display precision of the interferometer.

4.3 Signal Processing of Automatic Hammer Signal

The goal of signal processing for the automatic hammer setup is to reduce the effect of noise and systematic errors on the measured FRFs. There are two sources of noise in the automatic hammer system which are eliminated using digital signal processing. First, it is known that the force is zero except during the impact. The force data is multiplied by a window function which sets the force to zero away from the impact. Second, random noise is a significant problem, especially in the displacement sensor readings. So several measurements are taken and averaged as described by Ewins[9].

4.3.1 Force Measurement Windowing

The purpose of windowing is usually to reduce signal distortion (leakage) which occurs when the discrete Fourier transform of a non-periodic function is taken. In the automatic hammer system, this is not a problem. The measurement time is sufficiently long that the system has a chance to settle back to its steady state value after each impact, so it can be treated as a periodic system. Instead, the goal is to reduce the noise in the measured signal.

For an impact test, the applied force $f(t)$ is large for a small time; however the noise will occur over the entire measurement. Therefore, the overall noise in the signal can be significantly reduced by setting the force signal to zero outside the small region where the actual force signal exists. This is especially important for the automatic hammer. Since some of the measurement circuitry is run from the same circuitry as the solenoid actuator, the current drawn by the solenoid causes noise to appear in the force signal when the solenoid is activated.

There is a similar problem with the displacement measurement. Although there is no cross-talk between the solenoid driver circuit and the displacement circuit, since the response has an exponential falloff, it is mostly limited to within the first 0.2 s after the impact.

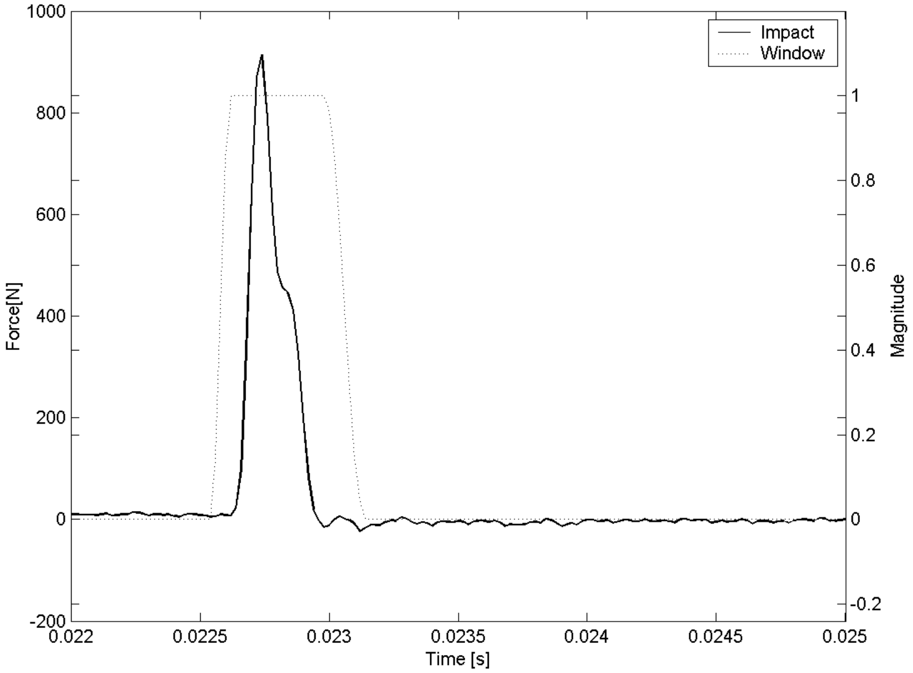

Figure 4.3: Cosine-Taper Window for Force Signal Windowing

The noise reduction algorithm is presented as follows. The discrete Fourier transform (DFT) of the measured signal f_n is

$$U_k = \sum_{n=0}^{N-1} f_n e^{\frac{2\pi i}{N}kn}. \tag{4.10}$$

where N is the total number of samples, and k is the number of the harmonic. Assuming that the measurement error is a Gaussian process with zero mean, the error at each point is

$$\begin{aligned}\epsilon_{U_k}^2 &= \sum_{m=0}^{N-1} \left| \frac{\partial}{\partial f_m} \sum_{n=0}^{N-1} f_n e^{\frac{2\pi i}{N}kn} \right|^2 \epsilon_{f_m}^2 \\ &= N\epsilon_f^2, \end{aligned} \tag{4.11}$$

where $epsilon_{f_m}$ is the measurement error at point m. Since the noise is assumed to be a stationary process, the error ϵ_{f_m} at each measurement point is identical. This error is called ϵ_f. The measured signal is windowed as

$$w_n = u_n w_n, \tag{4.12}$$

where w_n is the window function. For the windowed signal,

$$W_k = \sum_{n=0}^{N-1} w_n f_n e^{\frac{2\pi i}{N} kn}. \tag{4.13}$$

The error in the windowed signal is

$$\begin{aligned} \epsilon_{W_k}^2 &= \sum_{m=0}^{N-1} \left| \frac{\partial}{\partial f_m} \sum_{n=0}^{N-1} w_n f_n e^{\frac{2\pi i}{N} kn} \right|^2 \epsilon_{f_m}^2 && (4.14) \\ &= \epsilon_f^2 \sum_{n=0}^{N-1} w_n^2. && (4.15) \end{aligned}$$

As a result, the error is decreased by

$$\frac{\epsilon_{W_k}}{\epsilon_{U_k}} = \sqrt{\frac{1}{N} \sum_{n=0}^{N-1} w_n^2} \tag{4.16}$$

when the window is used.

For the force signal windowing, a cosine taper window was chosen. Such a window is shown with a typical force signal in figure 4.3. A cosine-taper window is defined as

$$W(t) = \begin{cases} \frac{1}{2}\left[\cos\left(\pi \frac{t-t_{ps}}{t_{ps}-t_{ts}}\right) + 1\right] & \text{if } t >= t_{ts} \text{ and } t < t_{ps} \\ 1 & \text{if } t >= t_{ps} \text{ and } t < t_{pe} \\ \frac{1}{2}\left[\cos\left(\pi \frac{t-t_{pe}}{t_{te}-t_{pe}}\right) + 1\right] & \text{if } t >= t_{pe} \text{ and } t < t_{te} \\ 0 & \text{otherwise} \end{cases} \tag{4.17}$$

There are four important points: the start of the initial cosine taper t_{ts}, the start of the plateau t_{ps}, the end of the plateau t_{pe}, and the end of the final cosine taper t_{te}. These points are automatically selected by finding the maximum force, and then finding the next point where the force falls to 10% of its peak value. The distance to this point is called $t_{shoulder}$. The end of the plateau is selected to be

$$t_{pe} = t_{max} + \frac{3}{2} t_{shoulder}, \tag{4.18}$$

and the end of the window is

$$t_{te} = t_{pe} + t_{shoulder}. \tag{4.19}$$

The last point before the force peak where the force rises above 10% of the peak value is also found, and the same procedure is followed to find t_{ps} and t_{ts}.

This method of determining the force window size was chosen to ensure that the force window was unity wherever the force sensor was in contact with the machine tool, it hugged the force signal as tightly as possible so as much of the noise as possible is eliminated, and it had a smooth drop off between 1 and 0.

A typical force signal produced by the automatic hammer has a width of 0.4 ms, with a 50 000 Hz sampling rate, and sample length of 0.2 s, which gives an error 6 % of the error of the unwindowed signal.

4.3.2 Spectral Averaging

In order to reduce the effect of noise on the calculated transfer function, several force and displacement measurements have been made. The Fourier transforms of the force $F_n(j\omega)$ and displacement $X_n(j\omega)$ of each measurement are calculated, and the auto- and cross-spectra of each measurement are calculated:

$$S_{ff,n} = |F_n|^2\,, \tag{4.20}$$

$$S_{xx,n} = |X_n|^2\,, \tag{4.21}$$

$$S_{fx,n} = F_n^* X_n. \tag{4.22}$$

The average spectrum over all M measurements is then calculated

$$\bar{S}_{ff} = \frac{1}{M}\sum_{n=1}^{M} S_{ff,n} \tag{4.23}$$

$$\bar{S}_{fx} = \frac{1}{M}\sum_{n=1}^{M} S_{fx,n} \tag{4.24}$$

$$\bar{S}_{xx} = \frac{1}{M}\sum_{n=1}^{M} S_{xx,n}. \tag{4.25}$$

These spectra are then used to calculate the measured transfer function one of two ways. Either

$$H_{m1} = \frac{\bar{S}_{fx}}{\bar{S}_{ff}}, \tag{4.26}$$

or

$$H_{m2} = \frac{\bar{S}_{xx}}{\bar{S}^*_{fx}} \tag{4.27}$$

In the noise free case these equations both simplify to the normal equation for calculating the transfer function

$$H_m = \frac{\bar{X}}{\bar{F}}. \tag{4.28}$$

If the transfer function is calculated using equation 4.28, the error at each point will be

$$\begin{aligned} \epsilon^2_{H_m} &= \sum_{m=1}^{M}\sum_{n=0}^{N-1}\left|\frac{\partial}{\partial x_{mn}}\frac{\bar{X}}{\bar{F}}\right|^2 \epsilon^2_{x_n} + \sum_{m=1}^{M}\sum_{n=0}^{N-1}\left|\frac{\partial}{\partial f_{mn}}\frac{\bar{X}}{\bar{F}}\right|^2 \epsilon^2_{f_n} \\ &= \frac{N}{M}\frac{1}{\left|\bar{F}\right|^2}\left[\epsilon^2_x + \left|H_m\right|^2 \epsilon^2_f\right] \end{aligned} \tag{4.29}$$

If the transfer function is calculated using equation 4.26, the error in the transfer function measurement is

$$\begin{aligned} \epsilon^2_{H_m} &= \sum_{m=1}^{M}\sum_{n=0}^{N-1}\left|\frac{\partial}{\partial x_{mn}}\frac{\bar{S}_{fx}}{\bar{S}_{ff}}\right|^2 \epsilon^2_{x_n} + \sum_{m=1}^{M}\sum_{n=0}^{N-1}\left|\frac{\partial}{\partial f_{mn}}\frac{\bar{S}_{fx}}{\bar{S}_{ff}}\right|^2 \epsilon^2_{f_n} \\ &= \frac{1}{\bar{S}^2_{ff}}\left[\sum_{m=1}^{M}\sum_{n=0}^{N-1}\left|\frac{\partial \bar{S}_{fx}}{\partial x_{mn}}\right|^2 \epsilon^2_{x_n} + \sum_{m=1}^{M}\sum_{n=0}^{N-1}\left|\frac{\partial \bar{S}_{fx}}{\partial f_{mn}} - H_{m1}\frac{\partial \bar{S}_{ff}}{\partial f_{mn}}\right|^2 \epsilon^2_{f_n}\right] \end{aligned} \tag{4.30}$$

but

$$\begin{aligned} \frac{\partial \bar{S}_{fx}}{\partial x_{mn}} &= \frac{\partial}{\partial x_{mn}}\frac{1}{M}\sum_{p=1}^{M} F^*_p \sum_{q=0}^{N-1} x_{pq} e^{-\frac{2\pi i k}{N}q} \\ &= \frac{F^*_m}{M} e^{-\frac{2\pi i k}{N}n}. \end{aligned} \tag{4.31}$$

Similarly,

$$\frac{\partial \bar{S}_{fx}}{\partial f_{mn}} = \frac{X_m}{M} e^{\frac{2\pi i k}{N}n}. \tag{4.32}$$

Finally,

$$\begin{aligned}
\frac{\partial \bar{S}_{ff}}{\partial f_{mn}} &= \frac{\partial}{\partial f_{mn}} \frac{1}{M} \sum_{p=1}^{M} \left(\sum_{q=0}^{N-1} f_{pq} e^{-\frac{2\pi i k}{N} q} \right)^{*} \left(\sum_{r=0}^{N-1} f_{pr} e^{-\frac{2\pi i k}{N} r} \right) \\
&= \frac{\partial}{\partial f_{mn}} \frac{1}{M} \sum_{p=1}^{M} \sum_{q=0}^{N-1} \sum_{r=0}^{N-1} f_{pq} f_{pr} e^{-\frac{2\pi i k}{N}(r-q)} \\
&= \frac{1}{M} \left[\sum_{q=0}^{N-1} f_{mq} e^{-\frac{2\pi i k}{N}(n-q)} + \sum_{r=0}^{N-1} f_{mr} e^{-\frac{2\pi i k}{N}(r-n)} \right] \\
&= \frac{2}{M} \sum_{q=0}^{N-1} f_{mq} \cos\left(\frac{2\pi k}{N} (n-q) \right) \\
&= \frac{2}{M} \left[\cos\left(\frac{2\pi n}{N} k \right) \sum_{q=0}^{N-1} f_{mq} \cos\left(\frac{2\pi q}{N} k \right) + \sin\left(\frac{2\pi n}{N} k \right) \sum_{q=0}^{N-1} f_{mq} \sin\left(\frac{2\pi q}{N} k \right) \right] \\
&= \frac{2}{M} \left[\cos\left(\frac{2\pi n}{N} k \right) \Re\left[F_m\right] + \sin\left(\frac{2\pi n}{N} k \right) \Im\left[F_m\right] \right]. \qquad (4.33)
\end{aligned}$$

Substituting equations 4.31 through 4.33 into equation 4.30 gives

$$\begin{aligned}
\epsilon_{H_{m1}}^2 = \frac{1}{M^2 \bar{S}_{ff}^2} &\left(\sum_{m=1}^{M} \sum_{n=0}^{N-1} S_{ff,m} \epsilon_x^2 \right. \\
&+ \sum_{m=1}^{M} \sum_{n=0}^{N-1} \left[S_{xx,m} - 4\Re\left[X_m^* H_{m1} e^{-\frac{2\pi k}{N} n} \right] \right. \\
&\cdot \left(\cos\left(\frac{2\pi n}{N} k \right) \Re\left[F_m\right] + \sin\left(\frac{2\pi n}{N} k \right) \Im\left[F_m\right] \right) \\
&\left. \left. + 4\left|H_{m1}\right|^2 \left(\cos\left(\frac{2\pi n}{N} k \right) \Re\left[F_m\right] + \sin\left(\frac{2\pi n}{N} k \right) \Im\left[F_m\right] \right)^2 \right] \epsilon_f^2 \right). \qquad (4.34)
\end{aligned}$$

In order to simplify this equation, approximate

$$\sum_{n=0}^{N-1} f\left(\frac{n}{N} \right) \approx \int_0^N f\left(\frac{n}{N} \right) dn \qquad (4.35)$$

for large N. Since

$$\int_0^T \cos\left(2\pi k \frac{x}{T} \right) \sin\left(2\pi k \frac{x}{T} \right) dx = 0 \qquad (4.36)$$

$$\int_0^T \cos^2\left(2\pi k \frac{x}{T} \right) dx = \frac{T}{2} \qquad (4.37)$$

$$\int_0^T \sin^2\left(2\pi k \frac{x}{T} \right) dx = \frac{T}{2} \qquad (4.38)$$

for any integer k, equation 4.34 reduces to

$$\epsilon_{H_{m1}}^2 \approx \frac{N}{M}\frac{|H_{m1}|^2}{\bar{S}_{ff}}\left[\frac{1}{|H_{m1}|^2}\epsilon_x^2 + \frac{\epsilon_f^2}{C_{fx}}\right], \tag{4.39}$$

where C_{fx} is the coherence, defined as

$$C_{fx} = \frac{|\bar{S}_{fx}|^2}{\bar{S}_{xx}\bar{S}_{ff}}, \tag{4.40}$$

which is roughly an indication of the error in the signal, and varies between 0 and 1. Similarly, the error in H_{m2} is

$$\begin{aligned}\epsilon_{H_{m2}}^2 &= \frac{1}{|\bar{S}_{fx}|^2}\left[\sum_{m=1}^{M}\sum_{n=0}^{N-1}\left|\frac{\partial \bar{S}_{xx}}{\partial x_{mn}} - H_{m2}\frac{\partial \bar{S}_{fx}^*}{\partial x_{mn}}\right|^2 \epsilon_{x_n}^2 + \sum_{m=1}^{M}\sum_{n=0}^{N-1}\left|H_{m2}\frac{\partial \bar{S}_{fx}^*}{\partial f_{mn}}\right|^2\right] \\ &\approx \frac{N}{M}\frac{|H_{m2}|^2}{\bar{S}_{xx}}\left[\frac{\epsilon_x^2}{C_{fx}} + |H_{m2}|^2\epsilon_f^2\right].\end{aligned} \tag{4.41}$$

By comparing equations 4.29, 4.39 and 4.41 a couple of observations can be made. First, in all cases the error increases with the square root of the number of the frequency range, and decreases with the square root of the number of measurements. Although increasing the number of measurements will always increase the accuracy of the measurement, as more measurements are made the improvements will decrease. When using the automatic hammer, an increased number of measurements can be used since each measurement takes less time.

Equation 4.26 is preferred over 4.27 because for the automatic hammer the force measurement has lower error than the displacement measurement, and in equation 4.26 the error in the force measurement ϵ_f contributes more to the overall error than the error in the displacement ϵ_x.

4.4 Hammer-Displacement Sensor Offset Compensation

To predict the chatter-free cutting conditions, it is necessary to obtain the frequency response function (FRF) from a force applied at the tool tip to the displacement at the tool tip. However, the automatic hammer does not apply the force at the tool tip, it applies the force approximately 15 mm above the tool tip (see figure 4.4. This will increase the apparent

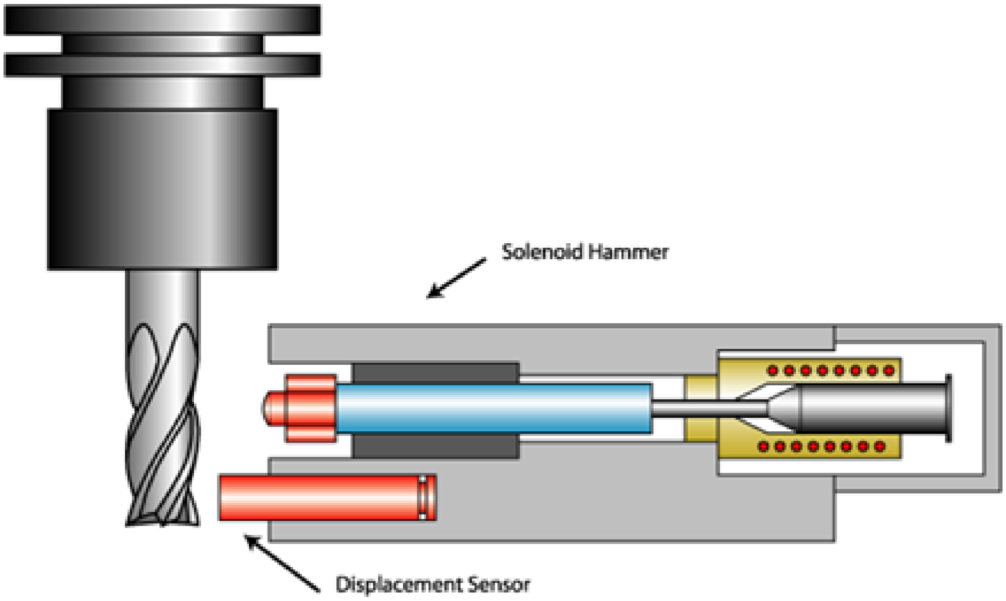

Figure 4.4: Automatic Hammer

stiffness of the cutting tool. As a result, the predicted chatter stability obtained using this FRF will be higher than the real chatter stability.

This apparent increase in stiffness can be compensated under the assumptions that

1. the most flexible part of the spindle/tool holder/tool structure is the tool,
2. the tool can be approximated as a beam, and
3. the bending of the tool is quasi-static.

Under these assumptions, the stiffness of the tool at the tool tip is

$$\frac{X_a}{F_a} = \frac{L^3}{3EI}, \tag{4.42}$$

where L is the stick-out of the tool from the toolholder, E is Young's modulus, and I is the area moment of inertia of the tool. The stiffness of the tool for a force applied higher on the tool and the displacement measured at the tool tip is

$$\frac{X_a}{F_b} = \frac{{l_b}^2}{6EI}\left(3L - l_b\right) \tag{4.43}$$

where $l_b = L - \epsilon$ where ϵ is the offset between the displacement sensor and force sensor in the automatic hammer. Solving equation 4.43 for EI and substituting into equation 4.42

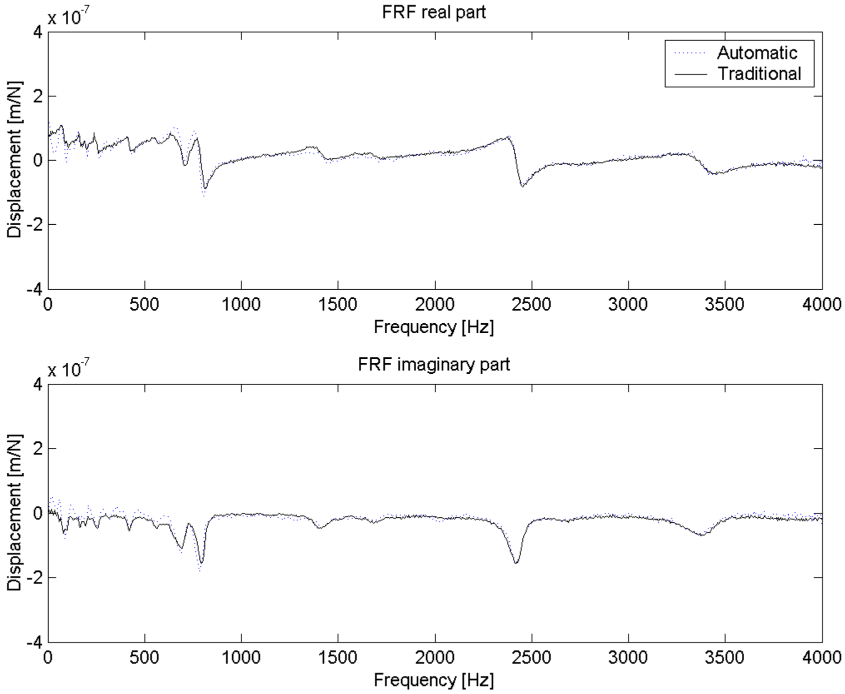

Figure 4.5: Comparison of Automatic Hammer vs. Traditional Tap Testing (Cross vs. Cross)

gives

$$\frac{X_a}{F_a} = \frac{2L^3}{{l_b}^2\,(3L - l_b)}\frac{X_a}{F_b}. \tag{4.44}$$

This equation is used to obtain an approximation of the direct transfer function by substituting the measured cross transfer function into X_a/F_b. The effect of this compensation can be seen by comparing figure 4.6 with figure 4.7. The FRF with the correction applied matches more closely with the FRF measured manually.

4.5 Comparison of Automatic Hammer with Traditional Impact Test

A test of the full automatic hammer system was performed, including edge detection and automatic calibration. This was compared with a FRF measurment performed with a manual hammer striking the tool at the same height as the automatic hammer, and measuring the response with a capacitive sensor positioned at the tool tip (see figure 4.5). In this case the

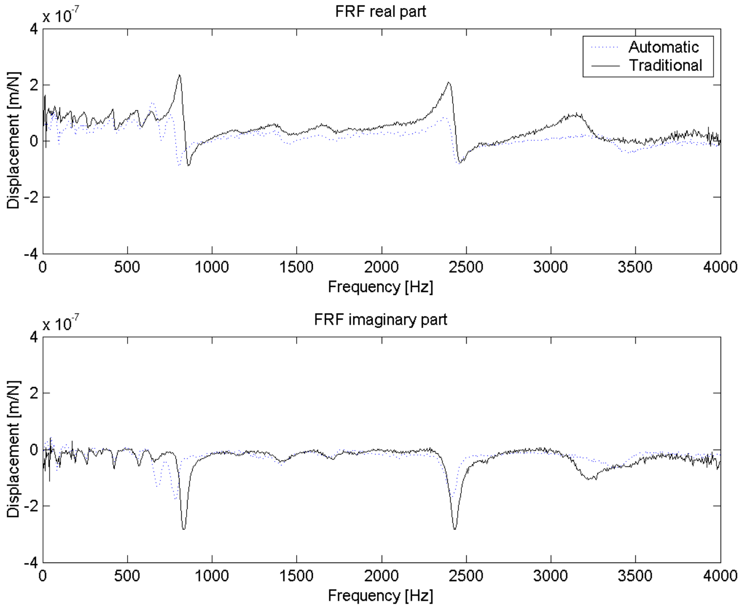

Figure 4.6: Comparison of Automatic Hammer vs. Traditional Tap Testing (Cross vs. Direct)

automatic hammer and traditional methods give very similar results.

The FRF measured using the automatic hammer was then compared with the FRF of the machine was then measured with a traditional tap testing method: an instrumented hammer to apply and measure the impulse, and an laser sensor to measure the response. In this case, both the instrumented hammer and laser sensor were positioned at the tool tip. Tests were done using a typical setup for each method, and a comparison without hammer-displacement sensor offset compensation is shown in figure 4.6. Since the automatic hammer is much quicker than the manual hammer, the transfer function obtained by the automatic hammer was averaged over 20 measurements, while the FRF obtained by the traditional method was only averaged over 5 measurements.

The increased stiffness due to the offset between the hammer and the displacement sensor can easily be observed near the mode at 2400 Hz. By contrast, the stiffness of this mode is much closer in the corrected FRF (figure 4.7).

The effect of this correction can be seen in the stability chart (figure 4.8). In all three cases (corrected automatic hammer FRF, uncorrected automatic hammer FRF and traditional

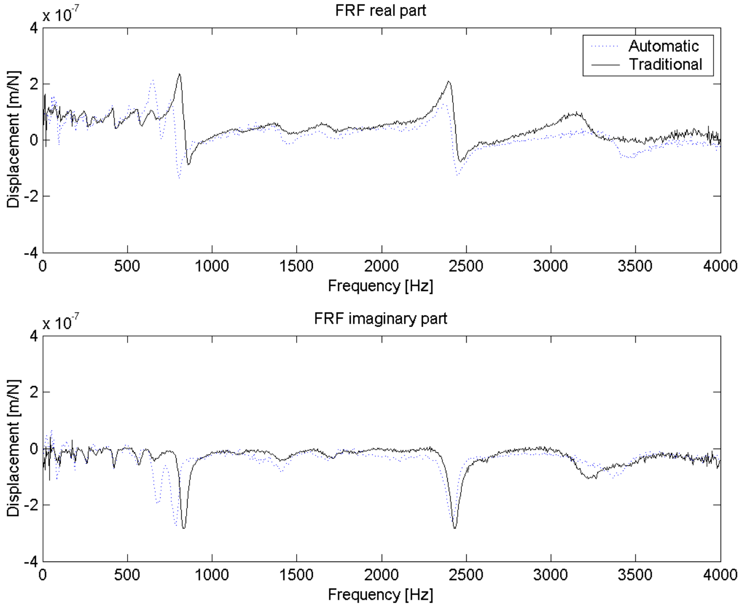

Figure 4.7: Comparison of Automatic Hammer vs Traditional Tap Testing (Compensated vs. Direct)

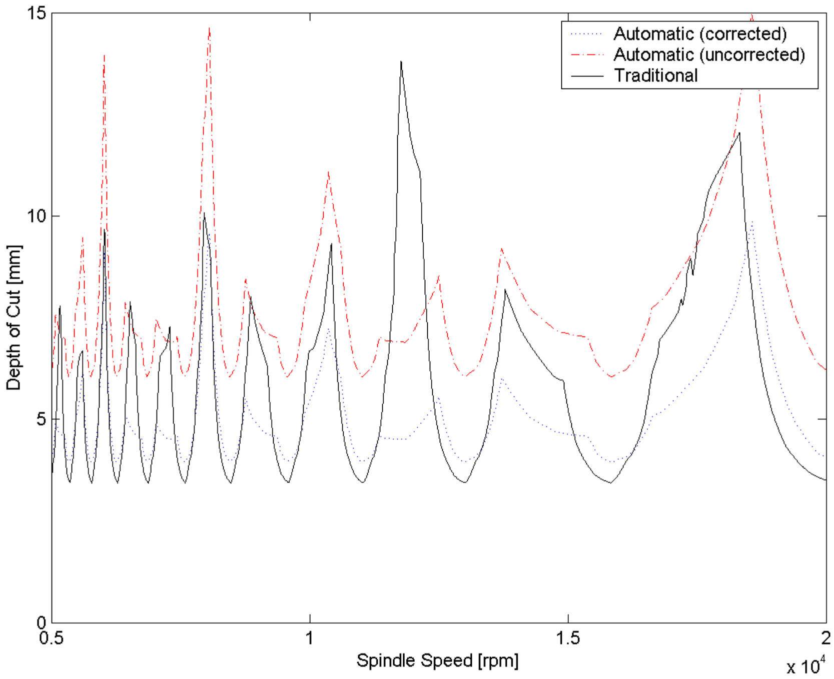

Figure 4.8: Predicted Stability of Slot Machining

method FRF) the stable cutting pockets occur at the same spindle speeds. However, due to the apparent increased stiffness, the uncorrected automatic hammer measurements have a much higher absolute stability limit compared with both the corrected and traditional FRF measurements. The stability limit inside the more stable pockets matches more closely with the uncorrected measurement than the corrected measurement. This is because the maximum stable depth of cut in these regions is determined principally by the spindle modes. This is unsurprising as these modes were not targetted by the correction algorithm, and the assumptions used when developing the correction algorithm do not apply in this region.

4.6 Conclusion

Since the automatic hammer is completely automated, it can measure the frequency response function of the machine tool much quicker than the traditional method. It can also be used by a machine tool operator with minimal training. However, the traditional method measures the direct FRF at the tool-tip, while the automatic method measures a cross-FRF, with the impact located approximately 15 mm above the response measurement point. When this cross-FRF is used to approximate the direct transfer FRF, it is similar to the direct FRF at low frequencies. At high frequencies the two FRF differ substantially, with the FRF measured by the automatic hammer not detecting modes above 3000 Hz. The FRF measured by the automatic hammer can be used as a starting point for locating the stable cutting regions.

Chapter 5

Effective Dynamometer Bandwidth Improvement

5.1 Introduction

The instantaneous cutting force measurements are used to identify material cutting constants, to trace the cutting load history on the machine tool during machining a workpiece, and to monitor the condition and cutting performance of the tools used.

Spindle load meters have only limited bandwidth (less than 10Hz), and can not measure instantaneous cutting forces at spindle speeds above a few hundred revolutions per minute. It is necessary to use cutting force dynamometers mounted on the table or integrated to the spindle. Integrating a dynamometer to the machine tool adds flexibility to the structure. If the periodic cutting force frequency is near a resonance frequency of the sensing system, the measured cutting forces are distorted and incorrectly recorded. The lowest natural frequency of the sensing system can be as low as 200 Hz depending on the mass of the workpiece mounted on the dynamometer, or spindle which houses the force cells. If a force is applied at this frequency, the workpiece and dynamometer will vibrate, and the measured force will be distorted compared to the applied force. In order to recover the real applied force from the distorted measured force, a state space model of the dynamometer is developed, and a Kalman filter is derived to recover the applied force, similar to the technique proposed by Park for his spindle integrated force sensor system[19].

In this technique, the dynamics of the mechanical system are first identified by applying a known force impact to the dynamometer using an instrumented hammer and measuring the response to get a frequency response function (FRF). The FRF is then used to construct a state space system which captures the dynamics of the dynamometer. This system is then modified as described below such that the applied force on the dynamometer is treated as another state. A Kalman filter is then designed to estimate the state of this model, and from that state an estimate of the applied force can be extracted.

5.2 Dynamic Model of Cutting Force Measurement System

The dynamometer is modeled as a linear time invariant (LTI) system which has an applied force F_a as the input and produces a measured force F_m. The dynamometer dynamics can be modeled as a transfer function of the form

$$\frac{F_m}{F_a} = H(s) = \sum_{k=1}^{\infty} \frac{K_k}{s^2 + 2\zeta_k \omega_{n,k} s + \omega_{n,k}^2}. \tag{5.1}$$

For simplicity, assume that the modes are ordered by natural frequency, $\omega_{n,1} \leq \omega_{n,2} \leq \dots$. In general any structure will have an infinite number of modes. Here the dynamics of the dynamometer are approximated by a model with a finite number of modes, M, which is valid for frequencies $0 < \omega < \omega_{n,M}$. For any mode

$$\begin{aligned} H_k(j\omega) &= \frac{K_k}{-\omega^2 + j2\zeta_k \omega_{n,k} \omega + \omega_{n,k}^2} \\ &\approx \frac{K_k}{\omega_{n,k}^2} \qquad \text{when } \omega \ll \omega_{n,k}, \end{aligned} \tag{5.2}$$

since if $\omega \ll \omega_{n,k}$,

$$\left|\omega^2 + j2\zeta_k \omega_{n,k} \omega\right| \ll \omega_{n,k}^2 \tag{5.3}$$

Therefore, in the domain of the reduced model, the contributions from higher modes can be approximated by a constant

$$K_0 = \sum_{k=M+1}^{\infty} \frac{K_k}{\omega_{n,k}^2}. \tag{5.4}$$

This gives an approximate transfer function of the dynamometer,

$$H(s) \approx K_0 + \sum_{k=1}^{M} \frac{K_k}{s^2 + 2\zeta_k \omega_{n,k} s + \omega_{n,k}^2} \text{for } 0 < \omega < \omega_{n,M} \tag{5.5}$$

To determine the transfer function of the dynamometer, the frequency response function (FRF) of the dynamometer is measured by striking it with an instrumented hammer. The input force from the hammer and output force from the dynamometer are measured. The measured FRF is fit to the above model using a non-linear least-squares method, with the additional constraint that the static gain of the system is known to be 1 since the both the hammer and dynamometer are calibrated to produce a force in Newtons, so

$$K_0 = 1 - \sum_{k=1}^{M} K_k. \tag{5.6}$$

To design a Kalman filter to compensate these dynamics, a state space system with these input-output characteristics must be designed. There is no unique state space system with certain system. Indeed, there are infinitely many state space systems with the same dynamics. The two standard forms are the controllable canonical form and the observable canonical form. Since the extended system must be observable, the observable canonical form is most appropriate. To write the state space model of the system in observable canonical form, the transfer function of the system must first be written as a system with a common denominator

$$H(s) = \frac{b_0 + b_1 s + \cdots + b_n s^n}{a_0 + a_1 s + \cdots + s^m}. \tag{5.7}$$

This can be written in observable canonical form as

$$\dot{\mathbf{x}} = \overbrace{\begin{bmatrix} -a_{m-1} & 1 & 0 & \cdots & 0 \\ -a_{m-2} & 0 & 1 & & \\ \vdots & & & \ddots & \vdots \\ -a_0 & 0 & 0 & \cdots & 0 \end{bmatrix}}^{=\mathbf{A}} x + \overbrace{\begin{bmatrix} b_n \\ b_{n-1} \\ \vdots \\ b_0 \end{bmatrix}}^{=\mathbf{B}} F_a \tag{5.8}$$

$$y = \underbrace{\begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}}_{=\mathbf{C}} \mathbf{x}. \tag{5.9}$$

For a typical mechanical system however, this process will have poor numerical properties. A useful measure of the numerical stability of any matrix U is the conditioning number,

$$\kappa(\mathbf{U}) = \|\mathbf{U}\| \left\|\mathbf{U}^{-1}\right\|, \tag{5.10}$$

where any norm can be used. In this thesis infinity norm is used for simplicity. For the observable canonical form,

$$\left\|\mathbf{A}\right\|_{\infty} = \max\left\{a_0, a_1 + 1, \dots a_{m-1} + 1\right\}. \tag{5.11}$$

Since for the modes of interest of the dynamometer $\omega_i \gg \zeta_j \quad \forall i, j$,

$$\left\|\mathbf{A}\right\|_{\infty} = a_0 = \prod_{k}^{M} \omega_k^2. \tag{5.12}$$

Similarly,

$$\begin{aligned}
\left\|\mathbf{A}^{-1}\right\|_{\infty} &= \frac{\left\|\begin{matrix} 0 & 0 & \cdots & 1 \\ a_0 & 0 & & -a_{m-1} \\ 0 & a_0 & & -a_{m-2} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & -a_1 \end{matrix}\right\|}{|a_0|} && \text{using Cramer's rule} \qquad (5.13) \\
&= \max_{k>0}\left\{\frac{1}{a_0}, 1 - \frac{a_k}{a_0}\right\} && \approx 1. \qquad (5.14)
\end{aligned}$$

This gives a conditioning of the state transition matrix A of

$$\kappa(\mathbf{A}) \approx a_0. \tag{5.15}$$

A system is said to be ill-conditioned if the reciprocal of the condition number approaches the machine precision, which is 10^{-12} for double precision[21]. The natural frequencies of a dynamometer are $\omega_n \approx 10^3$, so the observable canonical form will be ill-conditioned with more than two modeled modes.

Instead, the state-space model is constructed by composing the state space models of several mode individually. Split the transfer function into the sum of transfer functions for each mode

$$H(s) = \sum_{k=1}^{M} H_k(s), \tag{5.16}$$

where

$$H_k(s) = \frac{K_k}{s^2 + 2\zeta_k\omega_{n,k}s + \omega_{n,k}^2}. \tag{5.17}$$

Each individual mode has a corresponding canonical observable form

$$\mathbf{\dot{x}_k} = \overbrace{\begin{bmatrix} -2\zeta_k\omega_{n,k} & 1 \\ -\omega_{n,k}^2 & 0 \end{bmatrix}}^{=\mathbf{A_n}} \mathbf{x_k} + \overbrace{\begin{bmatrix} 0 \\ K_k \end{bmatrix}}^{=B_n} F_a \tag{5.18}$$

$$y_k = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{=C_n} \mathbf{x_k} \tag{5.19}$$

The total response of the system y is given as:

$$y = \sum_{k=1}^{M} y_k + K_0 F_a. \tag{5.20}$$

Let $\mathbf{x}$ be the total state of the system

$$\mathbf{x}' = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \tag{5.21}$$

then a state space model of the system can be written as

$$\mathbf{\dot{x}}' = \overbrace{\begin{bmatrix} -2\zeta_1\omega_{n,1} & 1 & 0 & 0 & \cdots & 0 & 0 \\ -\omega_{n,1}^2 & 0 & 0 & 0 & & 0 & 0 \\ 0 & 0 & -2\zeta_k\omega_{n,k} & 1 & & 0 & 0 \\ 0 & 0 & -\omega_{n,k}^2 & 0 & & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & & & & -2\zeta_k\omega_{n,k} & 1 \\ 0 & 0 & \cdots & & & -\omega_{n,k}^2 & 0 \end{bmatrix}}^{=\mathbf{A}'} \mathbf{x} + \overbrace{\begin{bmatrix} 0 \\ K_1 \\ 0 \\ K_2 \\ \vdots \\ 0 \\ K_M \end{bmatrix}}^{=\mathbf{B}'} F_a \tag{5.22}$$

$$y = \underbrace{\begin{bmatrix} 1 & 0 & 1 & 0 & \cdots & 1 & 0 \end{bmatrix}}_{=\mathbf{C}'} \mathbf{x}' + \underbrace{K_0}_{=\mathbf{D}'} F_a. \tag{5.23}$$

For this new state space model,

$$\|\mathbf{A}'\|_\infty = \max \omega_{n,k}^2, \tag{5.24}$$

and

$$\begin{aligned}\left\|\mathbf{A}'^{-1}\right\| &= \left\|\begin{matrix}\mathbf{A_1}^{-1} & 0 & \cdots & 0\\ 0 & \mathbf{A_2}^{-1} & & \\ \vdots & & \ddots & \vdots \\ 0 & & \cdots & \mathbf{A_M}^{-1}\end{matrix}\right\| \\ &= \max_k\left\{\frac{\omega_{n,k}}{\zeta_k}\right\}. \end{aligned} \tag{5.25}$$

This gives a condition number approximately

$$\kappa(\mathbf{A}') \approx \max_k\left\{\omega_{n,k}^2\right\}. \tag{5.26}$$

This form is well conditioned for any number of modes as long as the natural frequency of all modeled modes $\omega_{n,k} < 10^6$ rad/s.

In order to construct a Kalman filter, it is necessary to write the system in state space form with known inputs and outputs, and unknown process and measurement noise. For the Kalman filter to be optimal, the process and measurement noise must both be uncorrelated zero mean white noise processes with a known variance. However, in the case of milling, the force applied to the dynamometer is not a white-noise. A model must be developed which models the applied force as white-noise passed through some filter[13]. To obtain this filter, the applied force is written as a periodic function $F_p(t)$ multiplied by some slowly varying envelope $A(t)$

$$F_a(t) = A(t)F_p(t). \tag{5.27}$$

Expanding F_p as a Fourier series gives

$$F_a = \sum_{n=-\infty}^{\infty} a_n(t)e^{\frac{2\pi i n}{T}t}, \tag{5.28}$$

where $A(t)$ is considered approximately constant. Defining

$$F_{p,n} = a_n(t)e^{\frac{2\pi i n}{T}t}, \tag{5.29}$$

and

$$G_{p,n} = iF_{p,n}, \tag{5.30}$$

gives

$$\dot{F}_{p,n} = \frac{2\pi n}{T} G_{p,n} + \dot{a}_n e^{\frac{2\pi i n}{T} t} \tag{5.31}$$

$$\dot{G}_{p,n} = -\frac{2\pi n}{T} F_{p,n} + i\dot{a}_n e^{\frac{2\pi i n}{T} t}, \tag{5.32}$$

for $n \neq 0$ and

$$\dot{F}_{p,0} = \dot{a}_0 \tag{5.33}$$

Since $\dot{a}_n$ is not known a priori, $\dot{a}_n e^{\frac{2\pi i n}{T} t}$ can be treated as a white-noise source with variance

$$Q_n = \dot{a}_n^2 = \text{constant}. \tag{5.34}$$

This has the form of a state-space model with an infinite number of states. For the purpose of the dynamometer compensation, this model is approximated as a model with $2N{+}1$ states, the DC component and the first N harmonics. The tests done in this thesis have been performed with a 4-fluted tool, so $N = 12$ was chosen, with the fundamental frequency being the spindle frequency. The spindle frequency was chosen as the fundamental frequency instead of the tooth passing frequency to account for any runout in the measurement. By using 12 harmonics, the third harmonic of the tooth passing frequency is included, which is required to properly capture the cutting forces[19].

$$\begin{aligned}
\dot{\mathbf{x}}_{\mathbf{f}} &= \overbrace{\frac{2\pi}{T}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 & & 0 & 0 \\ \vdots & & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & 0 & & 0 & N \\ 0 & 0 & 0 & 0 & 0 & \cdots & -N & 0 \end{bmatrix}}^{=\mathbf{A_f}} \mathbf{x_f} + \overbrace{\mathbf{I}}^{=\mathbf{B_f}} \mathbf{w} \\
F_a &= \underbrace{\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & \cdots & 1 & 0 \end{bmatrix}}_{=\mathbf{C_f}} \mathbf{x_f}.
\end{aligned} \tag{5.35}$$

The dynamometer model is expanded to a composite model which includes the force model. The state vector of the composite model contains the states of both the dynamometer model and the applied force model

$$\mathbf{z} = \begin{bmatrix} \mathbf{x}' \\ \mathbf{x_f} \end{bmatrix}. \tag{5.36}$$

The expanded system is then

$$\dot{\mathbf{z}}(t) = \begin{bmatrix} \mathbf{A}' & \mathbf{B}'\mathbf{C_f} \\ \mathbf{0} & \mathbf{A_f} \end{bmatrix} \mathbf{z}(t) + \begin{bmatrix} \mathbf{0} \\ \mathbf{B_f} \end{bmatrix} \mathbf{w}(t) \tag{5.37}$$

$$\mathbf{F_m} = \begin{bmatrix} \mathbf{C}' & \mathbf{D}'\mathbf{C_f} \end{bmatrix} \mathbf{z}(t) + v(t). \tag{5.38}$$

This gives a new state space system with state space matrices

$$\begin{gathered} \hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A}' & \mathbf{B}'\mathbf{C_f} \\ \mathbf{0} & \mathbf{A_f} \end{bmatrix} \quad \hat{\mathbf{B}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \\ \hat{\mathbf{C}} = \begin{bmatrix} \mathbf{C}' & \mathbf{D}'\mathbf{C_f} \end{bmatrix} \quad \hat{D} = 0, \end{gathered} \tag{5.39}$$

which is disturbed by some white noise $\begin{bmatrix} \mathbf{0} & \mathbf{B_f} \end{bmatrix}^T \mathbf{w}(t)$ and $v(t)$, each with impulsive correlation

$$E\left\{ \begin{bmatrix} \mathbf{0} \\ \mathbf{B_f} \end{bmatrix} \mathbf{w}(t)\mathbf{w}(\zeta)^T \begin{bmatrix} \mathbf{0} & \mathbf{B_f} \end{bmatrix} \right\} = \mathbf{Q}\delta(t-\zeta) \tag{5.40}$$

$$E\left\{v(t)v(\zeta)\right\} = R\delta(t-\zeta). \tag{5.41}$$

$v(t)$ and $\mathbf{w}(t)$ are also assumed to be mutually uncorrelated.

To estimate the applied force it is necessary to find a linear filter which is driven by the measured force F_m and produces a state estimate $\hat{\mathbf{z}}$. An estimate of the applied force $\hat{F}_a$ can be extracted from the estimated state. The filter will be optimized by minimizing the function

$$J_t = E\left\{\tilde{\mathbf{z}}(t)\tilde{\mathbf{z}}(t)^T\right\} \tag{5.42}$$

where

$$\tilde{\mathbf{z}}(t) = \hat{\mathbf{z}}(t) - \mathbf{z}(t) \tag{5.43}$$

is the estimation error. This is the Kalman filter problem.

To derive the solution to this problem, the derivation of the Kalman filter given in Goodwin[13] is followed. Begin by considering a time-varying state-space model

$$\dot{\mathbf{x_t}}(t) = \mathbf{A_t}(t)\mathbf{x_t}(t) + \mathbf{w_t}(t)y_t(t) = \mathbf{C_t}(t)\mathbf{x_t}(t) + v_t(t) \tag{5.44}$$

where $\mathbf{w_t}(t)$ and $v_t(t)$ have zero mean and are uncorrelated, and

$$E\left\{\mathbf{w_t}(t)\mathbf{w_t}(\zeta)^T\right\} = \mathbf{Q_t}\delta(t-\zeta) \tag{5.45}$$

$$E\left\{v(t)v(\zeta)\right\} = R_t\delta(t-\zeta). \tag{5.46}$$

For this model, compute the covariance $\bar{\mathbf{P}}(t) = E\left\{\mathbf{x_t}(t)\mathbf{x_t}(\zeta)^T\right\}$ given the initial covariance $\bar{\mathbf{P_0}} = E\left\{\mathbf{x_t}(0)\mathbf{x_t}(0)^T\right\}$. The solution to equation 5.44 is

$$\mathbf{x_t}(t) = \boldsymbol{\phi_t}(t,0)\mathbf{x_{0t}} + \int_0^t \boldsymbol{\phi_t}(t,\tau)\mathbf{w_t}(\tau)d\tau \tag{5.47}$$

where the state transition matrix $\boldsymbol{\phi_t}(t,\tau)$ is the matrix such that

$$\mathbf{x_t}(t) = \boldsymbol{\phi_t}(t,\tau)\mathbf{x_t}(\tau). \tag{5.48}$$

Squaring the solution given in equation 5.48 and taking the expectation gives

$$\bar{\mathbf{P}}(t) = \boldsymbol{\phi_t}(t,0)\bar{\mathbf{P}}_\mathbf{0}\boldsymbol{\phi_t}(t,0)^T + \int_0^t \boldsymbol{\phi_t}(t,\tau)\mathbf{Q_t}(\tau)\boldsymbol{\phi_t}(t,\tau)^T d\tau. \tag{5.49}$$

Differentiating this equation gives

$$\frac{d\bar{\mathbf{P}}(t)}{dt} = \mathbf{A_t}(t)\bar{\mathbf{P}}(t) + \bar{\mathbf{P}}(t)\mathbf{A_t}(t)^T + \mathbf{Q_t}(t). \tag{5.50}$$

Returning to the original problem of obtaining an estimate, $\hat{\mathbf{z}}(t)$, of the state $\mathbf{z}(t)$. Assume the filter has the form:

$$\frac{d\hat{\mathbf{z}}(t)}{dt} = \hat{\mathbf{A}}\hat{\mathbf{z}}(t) + \mathbf{K}(t)\left[F_m(t) - \hat{\mathbf{C}}\hat{\mathbf{z}}(t)\right], \tag{5.51}$$

where $\mathbf{K}(t)$ is a gain matrix yet to be determined. Subtracting equation 5.37 from equation 5.51 gives

$$\frac{d\tilde{\mathbf{z}}(t)}{dt} = \left(\hat{\mathbf{A}} - \mathbf{K}(t)\hat{\mathbf{C}}\right)\tilde{\mathbf{z}}(t) + \mathbf{K}(t)v(t) - \mathbf{w}(t). \tag{5.52}$$

Comparing equation 5.52 with 5.44 gives

$$\mathbf{x_t}(t) = \tilde{\mathbf{z}}(t); \qquad \mathbf{A_t}(t) = \left(\hat{\mathbf{A}} - \mathbf{K}(t)\hat{\mathbf{C}}\right); \qquad \mathbf{w_t}(t) = \mathbf{K}(t)v(t) - \mathbf{w}(t). \tag{5.53}$$

Therefore

$$\frac{d\mathbf{P}(t)}{dt} = \left(\hat{\mathbf{A}} - \mathbf{K}(t)\hat{\mathbf{C}}\right)\mathbf{P}(\mathbf{t}) + \mathbf{P}(t)\left(\hat{\mathbf{A}} - \mathbf{K}(t)\hat{\mathbf{C}}\right)^T + \mathbf{K}(t)\mathbf{R}\mathbf{K}(t)^T + \mathbf{Q}. \tag{5.54}$$

To determine the gain $\mathbf{K}(t)$ which minimizes equation 5.54, let

$$\mathbf{K}(t) = \hat{\mathbf{K}}(t) + \tilde{\mathbf{K}}(t), \tag{5.55}$$

where

$$\hat{\mathbf{K}}(t) = \mathbf{P}(t)\hat{\mathbf{C}}^T\mathbf{R}^{-1}. \tag{5.56}$$

Substituting equation 5.55 into equation 5.54 gives

$$\begin{aligned}\frac{d\mathbf{P}(t)}{dt} = &\left(\hat{\mathbf{A}} - \hat{\mathbf{K}}(t)\hat{\mathbf{C}}\right)\mathbf{P(t)} + \mathbf{P(t)}\left(\hat{\mathbf{A}} - \hat{\mathbf{K}}(t)\hat{\mathbf{C}}\right)^T \\ &+ \hat{\mathbf{K}}(t)\mathbf{R}\hat{\mathbf{K}}(t)^T + \tilde{\mathbf{K}}(t)\mathbf{R}\tilde{\mathbf{K}}(t)^T + \mathbf{Q}.\end{aligned} \tag{5.57}$$

It can clearly be seen that $\dot{\mathbf{P}}(t)$ is minimized if $\tilde{\mathbf{K}}(t) = 0$. Thus, $\hat{\mathbf{K}}(t)$ is the optimal time varying filter gain. At a steady state condition,

$$\begin{aligned}\mathbf{0} = &\left(\hat{\mathbf{A}} - \hat{\mathbf{K}}(t)\hat{\mathbf{C}}\right)\mathbf{P(t)} + \mathbf{P}_\infty\left(\hat{\mathbf{A}} - \hat{\mathbf{K}}(t)\hat{\mathbf{C}}\right)^T \\ &+ \hat{\mathbf{K}}(t)\mathbf{R}\hat{\mathbf{K}}(t)^T + \tilde{\mathbf{K}}(t)\mathbf{R}\tilde{\mathbf{K}}(t)^T + \mathbf{Q}\end{aligned} \tag{5.58}$$

$$\hat{\mathbf{K}}_\infty = \mathbf{P}_\infty\hat{\mathbf{C}}^T\mathbf{R}^{-1}. \tag{5.59}$$

The steady state Kalman gain $\hat{K}_\infty$ is used instead of the time varying gain $\hat{\mathbf{K}}(t)$ to avoid recomputing the Kalman gain at every time.

To summarize, the estimated applied force is

$$\dot{\hat{\mathbf{z}}} = \hat{\mathbf{A}}\hat{\mathbf{z}}(t) + \hat{\mathbf{K}}\left(F_m - \hat{\mathbf{C}}\hat{\mathbf{z}}(t)\right) \tag{5.60}$$

$$\hat{F}_a = \begin{bmatrix}\mathbf{0}_{1\times 2M} & \mathbf{C_f}\end{bmatrix}\hat{\mathbf{z}} \tag{5.61}$$

where the optimal gain $\hat{\mathbf{K}}$ satisfies

$$\hat{\mathbf{K}} = \mathbf{P}\hat{\mathbf{C}}^T\mathbf{R}^{-1} \tag{5.62}$$

and $\mathbf{P}$ is the solution to

$$\mathbf{0} = \mathbf{Q} - \mathbf{P}\hat{\mathbf{C}}^T\mathbf{R}^{-1}\mathbf{C}\mathbf{P} + \mathbf{P}\hat{\mathbf{A}}^T + \hat{\mathbf{A}}\mathbf{P}. \tag{5.63}$$

The full system is show in figure 5.1.

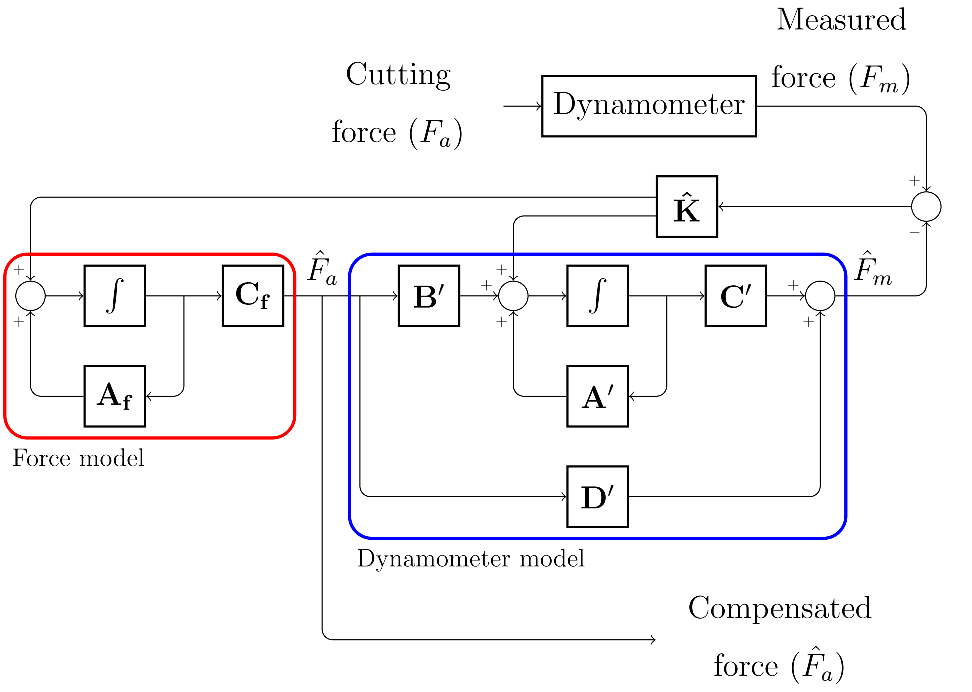

Figure 5.1: Kalman Filter Block Diagram

5.3 Experimental Results

One method which could be used to test the dynamometer force compensation results would be to measure the force with both a dynamometer and another sensor, and compare the force measured by the dynamometer before and after compensation with the real applied force measured by the other sensor.

There are several practical problems with method. Primarily, the reference sensor must have a large enough bandwidth such that forces near the first mode of the sensor to be compensated are measured undistorted by the reference sensor. In addition, the reference dynamometer must be small enough (or the dynamometer to be compensated must be large enough) that the reference dynamometer can be mounted on top of the test dynamometer.

For these reasons, using a reference dynamometer was impractical. Instead, an indirect method of testing the Kalman filter was used. The tangential force on each tooth of a cutting tool is

$$F_t(\phi) = K_{tc}ah(\phi) + K_{te}a \tag{5.64}$$

when the tooth is cutting, where $h(\phi)$ is the chip thickness, which is approximately

$$h(\phi) = c \sin \phi. \tag{5.65}$$

The cutting coefficients K_{tc} and K_{te} do not depend on the spindle speed. A similar relationship holds for the radial cutting force $K_r(\phi)$. These forces can be transformed into the X and Y axis

$$F_x(\phi) = -F_t \cos \phi - F_r \sin \phi \tag{5.66}$$

$$F_y(\phi) = F_t \sin \phi - F_r \cos \phi \tag{5.67}$$

The cutting forces as a function of tool angle are independent of the spindle speed. To measure the effectiveness of the Kalman filter, the cutting forces at a low speed $F_{xl}(t), F_{yl}(t)$ are measured. These forces are converted to force as a function of angle using the relation

$$\phi = \Omega t. \tag{5.68}$$

The forces are measured at a spindle speed close to the resonance, and converted to forces as a function of angle, and compared with the low speed forces. If the Kalman filter performs as expected, the two forces will match.

The transfer function of the X- and Y-axes of the target dynamometer, and an associated models are shown in figures 5.2 and 5.3, respectively. For both axes there is one dominant mode; however, the dynamometer model includes several of the higher frequency modes as well. It was found that if these higher-frequency modes are not included, the mismatch between the model and the real dynamics can cause the an amplification of high-frequency noise.

In order to design a Kalman filter, both the variance of the measurement noise and the process noise must be known. The measurement noise can be estimated by measuring the variance of the dynamometer with no applied forces. The variance of the applied force is more difficult to measure, so instead of attempting to measure it, this is treated as a tunable parameter, which can be used to adjust the bandwidth of the generated Kalman filter. With the model developed previously, a Kalman filter is designed to compensate the dynamics of both the X and Y axes. Since the noise model depends on the spindle frequency, a separate

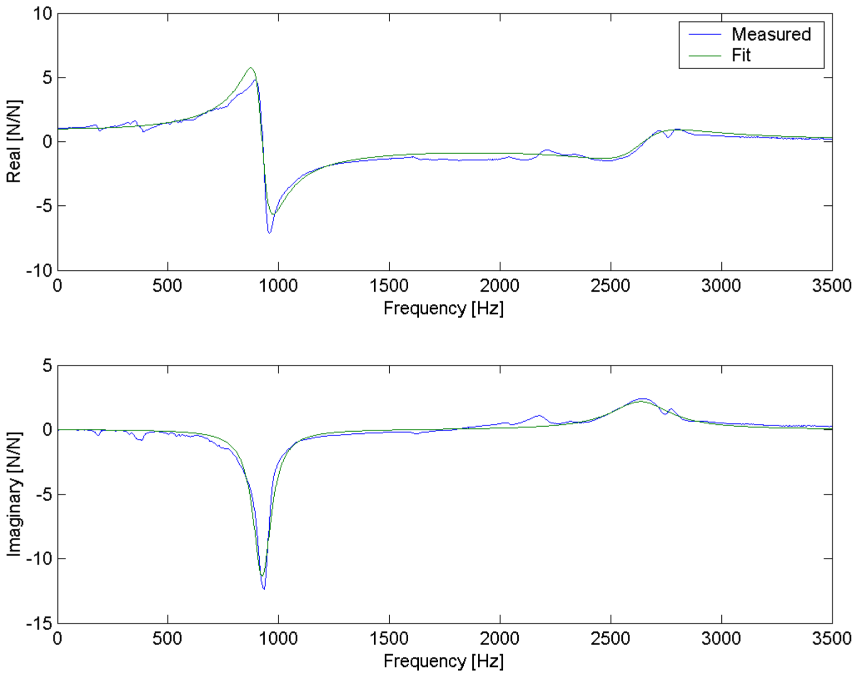

Figure 5.2: Dynamometer Force-Force Transfer Function

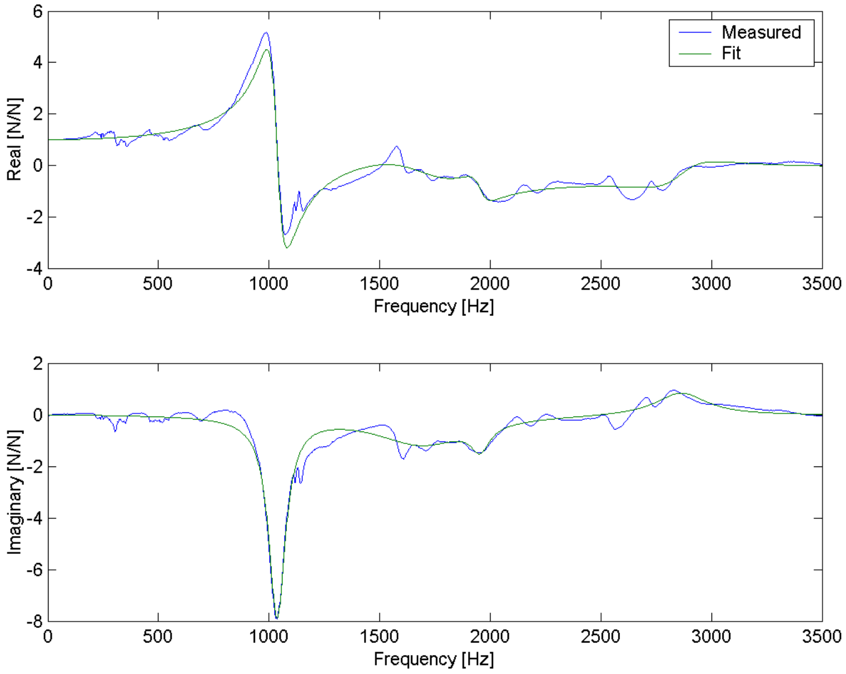

Figure 5.3: Dynamometer Force-Force Transfer Function

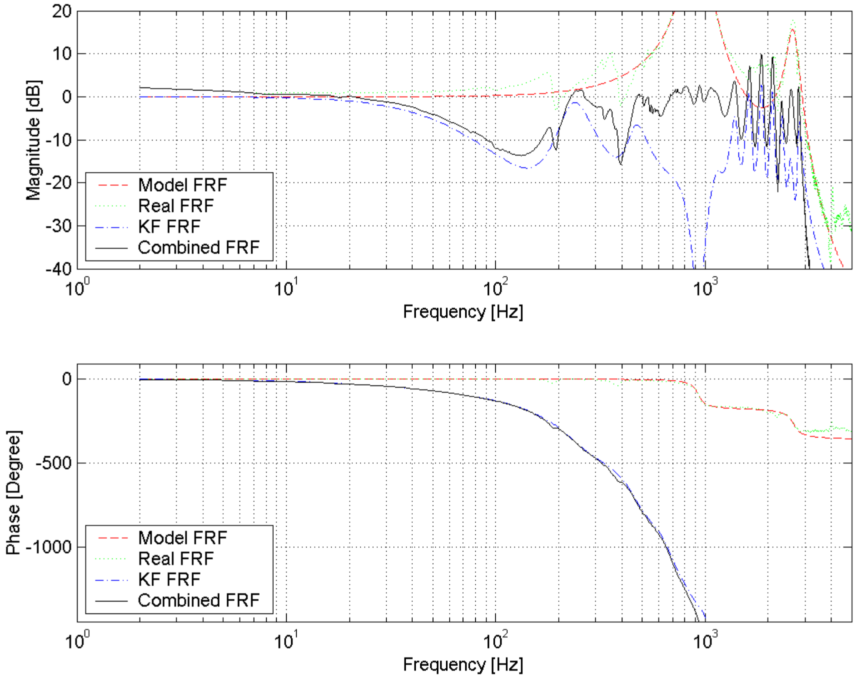

Figure 5.4: Kalman Filter for X-Axis at Tooth-Passing Frequency of 930 Hz

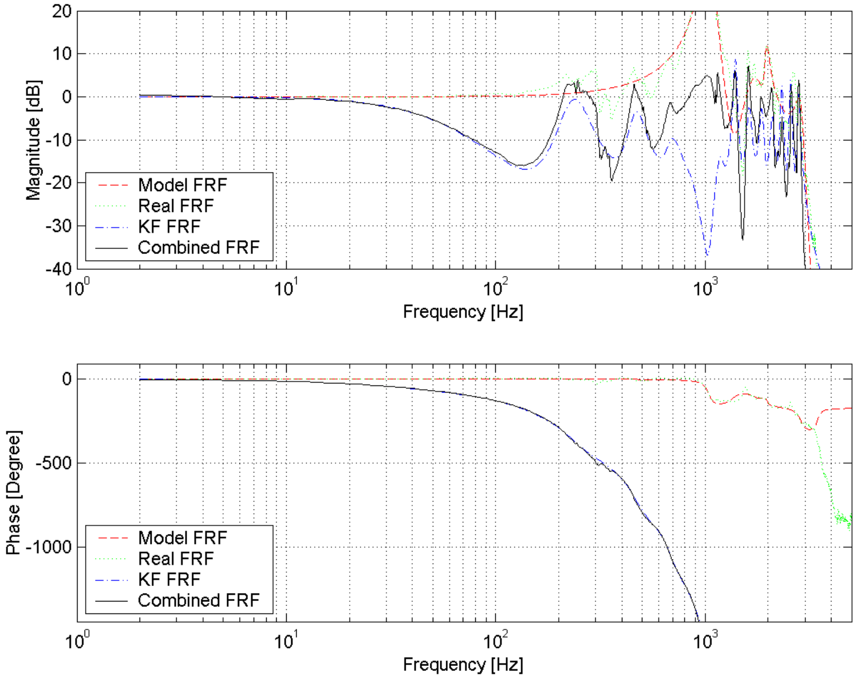

Figure 5.5: Kalman Filter for Y-Axis at Tooth-Passing Frequency of 930 Hz

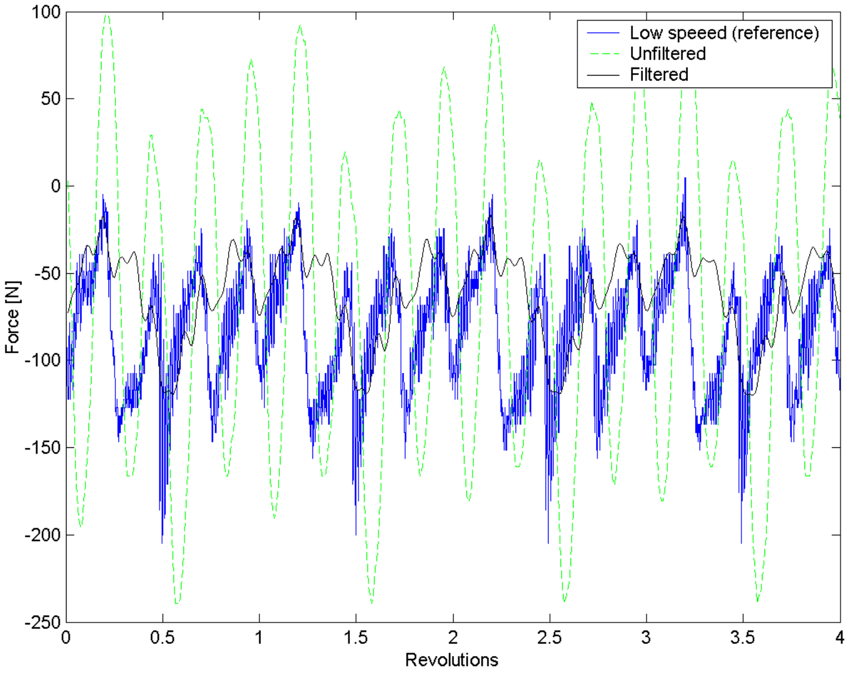

Figure 5.6: X-Axis Cutting Test at 13950 RPM

Kalman filter must be designed for each spindle frequency. Typical filters are shown in figures 5.4 and 5.5.

In order to test the effectiveness of the Kalman filter, two slotting cutting tests are performed with a four fluted cylindrical endmill. One test is performed with a spindle speed of 750 rpm, giving a tooth passing frequency of 50 Hz. Another test is performed with a spindle speed of 13950 rpm, a tooth passing frequency of 930 rpm, which corresponds to the natural frequency of the X axis of the dynamometer. The first, low frequency, test is undistorted by any dynamometer dynamics. The second test is compensated by the Kalman filter compensation algorithm, and both the uncompensated and compensated cutting forces are compared with the low speed reference test. The results are shown in figures 5.6 and 5.7.

To compare the effectiveness of the dynamometer compensation algorithm at different spindle speeds, tests were done at six different spindle speeds between 750 rpm and 18000 rpm with a four fluted tool. This gives tooth passing frequencies from 50 Hz to 1200 Hz. To compare the effectiveness of the compensation algorithm at different frequencies, the 10th and 90th percentiles were plotted as a function of tooth passing frequency in figures 5.8 and

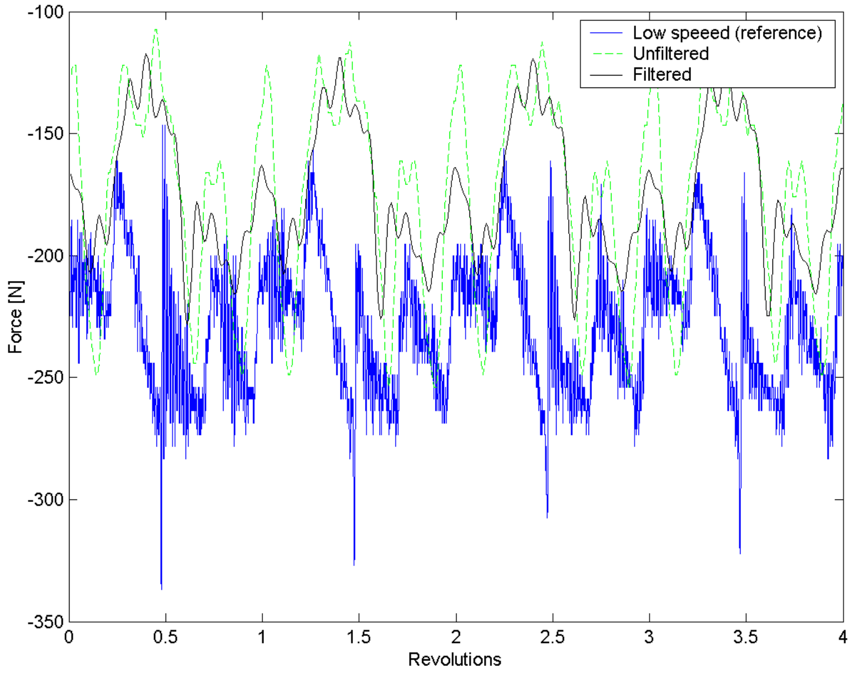

Figure 5.7: Y-Axis Cutting Test at 13950 RPM

5.9. This shows the range of force signal, which is expected to be constant. The 10th and 90th percentiles were chosen instead of the minimum and maximum to minimize the effect of noise. Although the range of the compensated cutting forces remains approximately constant over all tooth-passing frequencies, the average cutting force decreases as the spindle speed increases. This may occur because the cutting coefficients are decreasing, as the spindle speed increases, rather than staying constant as predicted.

In summary, the Kalman filter used in this chapter is capable of reproducing the applied force signal with the correct overall amplitude. It performs less well at correctly estimating the waveform of the applied force. This is because in order to accurately reproduce the cutting force waveform it is necessary to compensate the first several harmonics of the applied force, which lie beyond the bandwidth of the sensor. Also, for the dynamometer used, the region beyond the principal mode was difficult to model making compensation impossible.

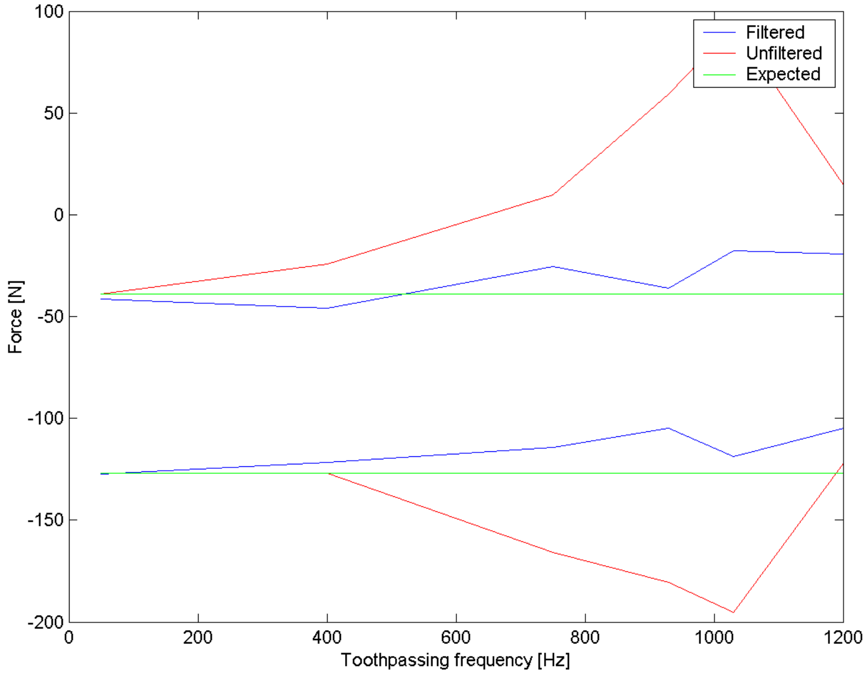

Figure 5.8: X-Axis Cutting Force Range by Tooth Passing Frequency(10th Percentile to 90th Percentile)

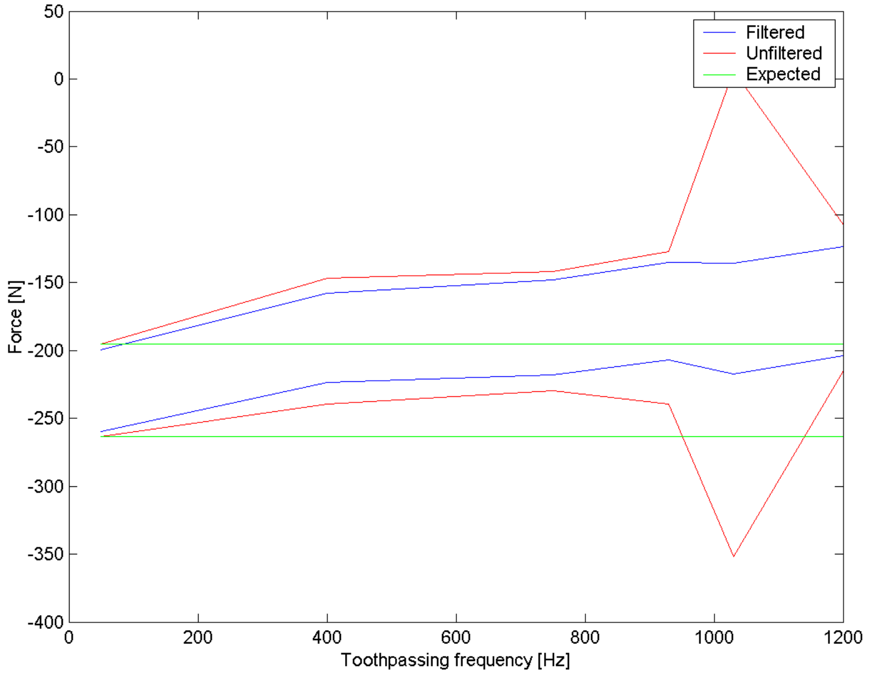

Figure 5.9: Y-Axis Cutting Force Range by Tooth Passing Frequency(10th Percentile to 90th Percentile)

Chapter 6

Integrated Smart Machining System

The automated cutting force coefficient identification from spindle power or dynamometers is integrated with the frequency response function of the tool measured automatically with the computer controlled hammer. The objective is to embed a smart machining system to a CNC, which allows the operator to automatically identify chatter free spindle speeds and cutting depths.

To produce an automatic smart machining system, it is necessary to integrate the algorithms developed in this thesis into a CNC machining system, so the necessary steps can be performed with the press of a button. In a production system, these algorithms would run on the CNC, or the front panel. This is impractical in a research setting. Instead, a prototype system was developed which placed the algorithms on a PC, which communicated with the CNC to execute the CNC-side commands and perform data collection from the CNC.

The prototype system consists of a Mori-Seiki NMV5000 milling machine using a Fanuc 30i CNC, a standard laptop PC with a data acquisition card connected to the Fanuc CNC via ethernet network (figure 6.1. The PC is able to send commands and read data from the Fanuc CNC using both the Fanuc FOCAS libraries[10], and the Fanuc Servo Guide software[11], which is built on top of the FOCAS libraries. The FOCAS libraries are used by the operator's panel to communicate with the CNC, so integrating the components developed in this thesis into the front panel of the machine tool is possible.

6.1 Integration with Cutting Coefficient Prediction

To measure the load meter voltage necessary to perform the cutting coefficient identification, the Fanuc Servo Guide software was used to measure the spindle load meter current. The process is as follows:

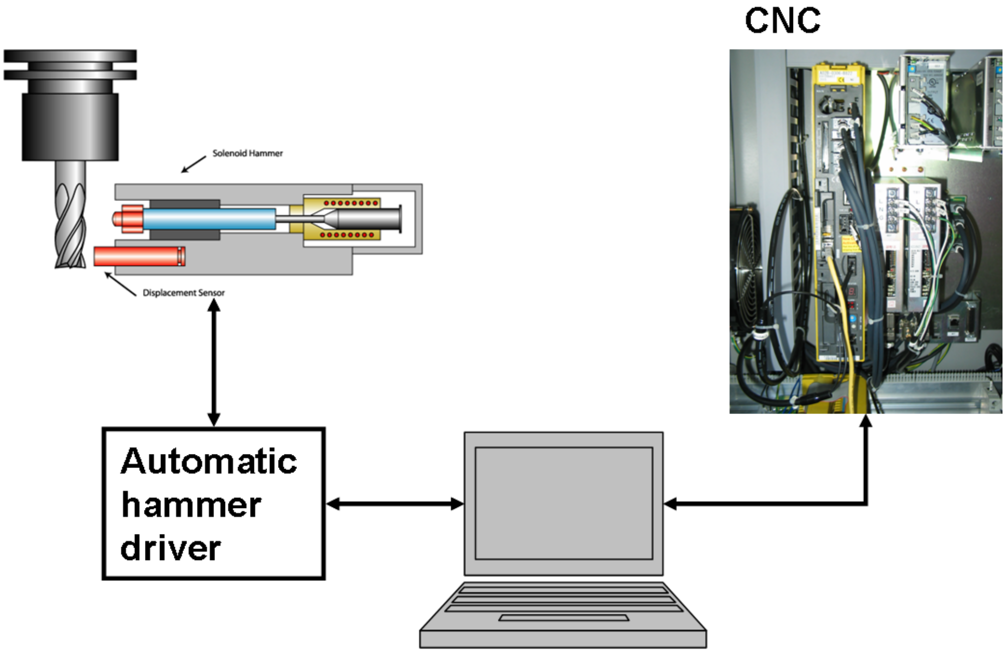

Figure 6.1: CNC and Automatic Hammer Connection to Laptop Computer

- A test slotting cut is programmed into the CNC, with the beginning of the cut at the start of a NC code block, with a cut at least a second long.

- The Fanuc Servo Guide software is set up to measure the spindle power (channel labeled LMDAT) with a start trigger at the block given in step 6.1.

- The test cut is run, and the spindle power measurements are collected.

- Steps 6.1 to 6.1 are repeated at different feeds per tooth.

- The data from the above steps is used to calculate the tangential cutting coefficient, as described in chapter 3.

Although the experimental setup used for this thesis uses the Fanuc Servo Guide, it is not necessary to use this software. The Servo Guide software uses undocumented functions of the Fanuc FOCAS libraries for its data acquisition routines. If documenation for these routines can be obtained, this procedure could be automated into a self-contained program.

6.2 Integration with Automated Transfer Function Measurement Unit

To implement the procedure for positioning, calibrating and actuating the automatic hammer described in chapter 4 it is necessary to be able to send commands to the CNC, and perform actions when those commands have been executed. For this, the Direct Numerical Control (DNC) facility of the Fanuc controller is used. This facility is intended to allow part programs to be sent directly from a PC. The Fanuc FOCAS libraries have functions which allow a DNC program to be generated dynamically. This facility enables automatic execution of the algorithms described in chapter 4.

The program for automated transfer function measurements is a C++ program which is divided into four sections: main loop, flute finding, capacitive sensor calibration, and the frequency response function (FRF) measurement. Full source code is provided in Appendix 7.1.

The main loop is contained entirely in the `main()` function. It establishes the connection to the CNC, and enters the `cnc_dncstart2()` function, which waits for the operator to start the machine in DNC mode. Once the `cnc_dncstart2()` function returns, the next steps of the algorithm are executed in order (flute finding, sensor calibration and FRF measurement). Once the FRF has been measured, the loop sends the program halt ocde (`M30`) to the CNC and terminates DNC mode by calling the `cnc_dncend2()` function. The operator can now change to another mode and continue machining, or switch tools and measure the FRF of another tool.

In order to execute the flute finding and sensor calibration algorithms, it is necessary to send commands to the CNC. This is done through the `run_cmd()` function, which in turn calls the FOCAS function `cnc_dnc2()`. The `cnc_dnc2()` command takes three parameters: a handle which identifies the CNC, the text of the CNC commands to be executed, and a pointer to an integer which contains the number of characters to be sent to the CNC. Once the `cnc_dnc2()` function returns, this integer contains the number of characters which were actually transmitted to the CNC. It is possible that only a part of the command string will be transmitted to the CNC, in which case the `cnc_dnc2()` function must be called again with the rest of the string of commands. This retransmission is performed by the `run_cmd()` function, so the rest of the program does not have to handle partial transmissions.

The flute finding code implements the algorithms described in section 4.1. In order to implement these algorithms, the distance from the capacitive sensor to the tool surface at a given angle must be measured. This is performed by

- Issuing a command to the CNC to rotate the spindle to the target angle. On the machine used, this is done by issuing the commands

  ```
  #1133=angle
  M119
  ```

  where `angle` is the desired angle θ expressed as a number from 0-4095

  $$\texttt{angle} = \frac{4096\theta}{2\pi}. \tag{6.1}$$

- Waiting for 200 ms for the CNC to begin acting on the command.

- Measuring the spindle speed and waiting for the spindle to come to rest. The spindle is assumed to have come to rest once the spindle speed is zero in five measurements in a row.

- The average voltage output by the capacitive sensor over 10 ms is measured using a data acquisition card (DAQ). This voltage is used as the distance between the capacitive sensor and the tool surface.

The tool positioning required for sensor calibration is performed similarly. For the axis positions it is possible to measure the distance remaining, so distance remaining is used to determine when the command has finished executing, instead of when the machine comes to rest.

- Issue a command to move the CNC by the amount required for the calibration algorithm

  ```
  Xdist
  ```

where `dist` is replaced with the desired offset.

- Wait until the distance remaining to travel is zero.

- The average voltage output by the capacitive sensor over 10 ms is measured using a data acquisition card (DAQ). This voltage is used as the distance between the capacitive sensor and the tool surface.

Finally, an external program (MalTF.exe) is started to perform the frequency response function measurement. This program accepts a command line flag to communicate the calculated displacement sensor sensitivity. In the future the frequency response function measurement, and tool positioning and sensor calibration code could be more tightly integrated, although this isn't necessary. The current level of integration achieves the objective of automatic frequency response function measurement.

6.3 Chatter Prediction

The tangential cutting coefficient measured using the method developed in chapter 3 is now combined with the frequency response function measured with the automatic FRF measurement system from chapter 4 to provide an estimate of the chatter-free cutting conditions. To calculate these chatter-free cutting conditions, the experimentally proven and widely used chatter stability method presented by Altintas and Budak[3] is used in the proposed system. The stability theory is briefly described here.

6.3.1 Chatter Stability Law

The dynamic milling force is expressed by

$$\begin{bmatrix} F_x(t) \\ F_y(t) \end{bmatrix} = \frac{1}{2} a K_t \begin{bmatrix} a_{xx}(t) & a_{xy}(t) \\ a_{yx}(t) & a_{yy}(t) \end{bmatrix} \begin{bmatrix} \Delta x(t) \\ \Delta y(t) \end{bmatrix} \tag{6.2}$$

where Δx and Δy are the vibrations between successive tooth periods. a_{ij} are the directional dynamic milling force coefficients, given as

$$a_{xx} = \sum_{j=0}^{N-1} -g(\phi_j)\left[\sin 2\phi_j + K_r\left(1 - \cos 2\phi_j\right)\right] \tag{6.3}$$

$$a_{xy} = \sum_{j=0}^{N-1} -g(\phi_j)\left[(1 + \cos 2\phi_j) + K_r \sin 2\phi_j\right] \tag{6.4}$$

$$a_{yx} = \sum_{j=0}^{N-1} g(\phi_j)\left[(1 - \cos 2\phi_j) - K_r \sin 2\phi_j\right] \tag{6.5}$$

$$a_{xx} = \sum_{j=0}^{N-1} g(\phi_j)\left[\sin 2\phi_j - K_r\left(1 + \cos 2\phi_j\right)\right], \tag{6.6}$$

where N is the number of teeth, and ϕ_j is the angular position of tooth j, and $g(\phi_j)$ is a function indicating if the tooth is in cut. It is unity if the tooth is cutting, and zero if the tooth is out of cut.

The dynamics of the machine tool spindle/tool holder/cutter combination can be identified from the frequency response function, and written as

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} \phi_{xx} & \phi_{xy} \\ \phi_{yx} & \phi_{yy} \end{bmatrix} \begin{bmatrix} F_x(t) \\ F_y(t) \end{bmatrix}. \tag{6.7}$$

By substituting this into 6.2, the eigenvalue problem

$$\left[\left[I\right] - \frac{1}{2}K_t a\left(F(t) - F(t+T)\right)\left[A(t)\right]\left[\Phi\right]\right] = 0 \tag{6.8}$$

is obtained. To solve this, some simplifications must be made. First, $[A(t)]$ is rewritten as

$$\left[A(t)\right] = \left[A_0\right] + \left[A'(t)\right], \tag{6.9}$$

where $[A_0]$ is the average value of $[A(t)]$. $[A'(t)]$ is ignored for an initial estimate. This equation will be critically stable if

$$\det\left[\left[I\right] - \frac{1}{2}K_t a\left(1 - e^{-i\omega_c T}\right)\left[A_0\right]\left[\Phi(i\omega_c)\right]\right] = 0, \tag{6.10}$$

where ω_c is the principal chatter frequency of the system. Let Λ be the eigenvalues of $\left[A_0\right]\left[\Phi(i\omega_c)\right]$. By comparison of the eigenvalue formula with equation 6.10,

$$\Lambda = -\frac{N}{4\pi}a_{lim}K_t(1 - e^{-i\omega_c T}). \tag{6.11}$$

Solving for a_{lim}, ensuring that it is a real number, gives

$$a_{lim} = -\frac{2\pi\Re[\Lambda]}{NK_t}(1+\kappa^2), \tag{6.12}$$

where

$$\kappa = \frac{\Im[\Lambda]}{\Re[\Lambda]}. \tag{6.13}$$

The spindle speed corresponding to each chatter frequency can also be calculated as

$$n = \frac{60}{N}\frac{\omega_c}{\pi - 2\tan^{-1}\kappa + 2k\pi} \tag{6.14}$$

in rpm, where k is chosen such that the spindle speed is positive.

From these equations, the chatter stability limit for a cutting process can be determined from the number of teeth N, the engagement profile of the cutter, the dynamics of the machine represented by the frequency response function (FRF) Φ, and the mechanistic cutting force coefficients of the cutter K_t and K_r.

6.4 Prediction of Stability Charts

In a smart machining system, the number of teeth and engagement angles are provided by the process planner. An estimate of the machine dynamics is measured by the automatic hammer system described in chapter 4, and K_t is estimated from the spindle load as described in chapter 3. The only missing parameter is K_r, which can be identified cutting force measurements with a dynamometer, but not from spindle power measurements.

The value of K_r depends on the geometry of the tool, and material properties of the workpiece material. Because of this, it will be different for each tool/workpiece combination. However, it is possible to get an estimate of the stability lobes without knowing the value of K_r. Figure 6.2 shows two different stability charts, one with $K_r = 0.25$ and one with $K_r = 0.5$, both during slotting. This represents the practical extremes in K_r variation. Although the two stability lobes differ substantially in critical depth of cut, they both share some important features. First, the absolute stable depth of cut is 6 mm in both cases. Second, the stability lobes have very similar natural frequencies in both cases. In the smart machining system, a K_r value of 0.3 will be used, as this is a common K_r value.

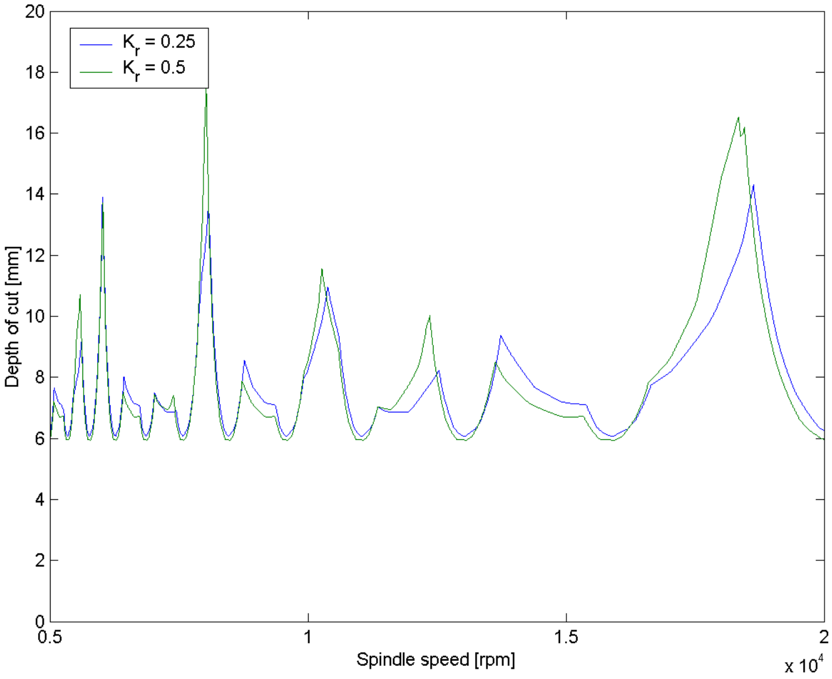

Figure 6.2: Comparison of Stability Chart with Different K_r

As a comparison, a stability chart was generated using the traditional instrumented hammer and accelerometer tap-testing method, and the cutting coefficients were obtained from a database through the orthogonal-to-oblique cutting coefficient transformation[4]. This was compared with the stability chart generated when these parameters were obtained as described in this thesis, and the result is shown in figure 6.3. Again, although the predicted maximum stable depth of cut varies widely at certain spindle speeds, the gross features, such as spindle speed of stable pockets, are comparable. The proposed smart machining strategy, which uses an automated hammer and low-cost estimation of cutting force coefficient from available spindle power, is robust and practical to use on production machines.

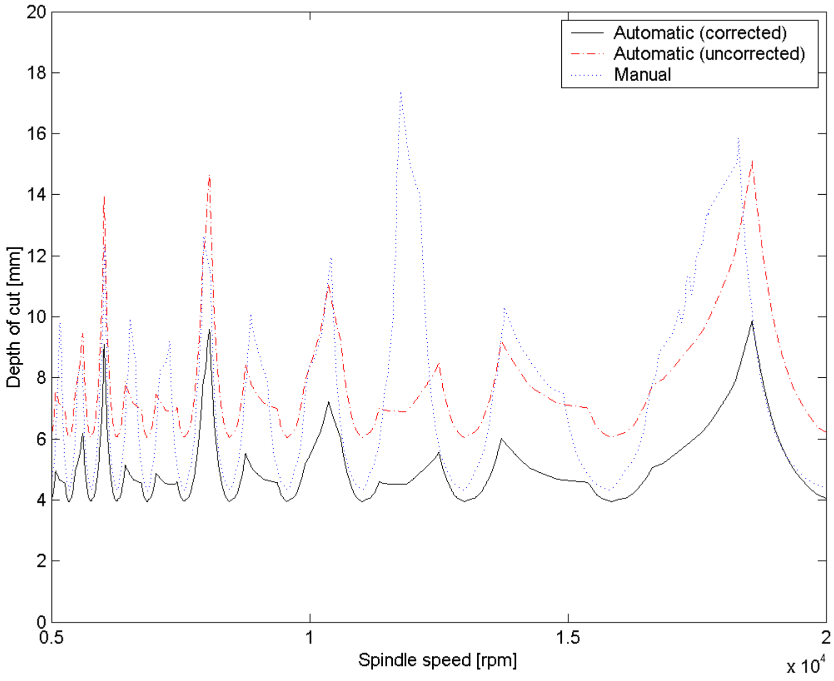

Figure 6.3: Comparison of Stability Chart with State of the Art Approach

Chapter 7

Summary and Future Research

In this thesis the construction of a system for automatically determining the stable, chatter-free cutting conditions for a cutting operation was investigated. This system consists of four parts. First is a method to interface with a Computer Numerically Controlled (CNC) milling machine and determine the tangential cutting coefficient from the spindle load meter readings. Second, a solenoid-actuated automatic frequency response measurement unit, which is integrated to the CNC machine to automatically position the cutting tool correctly and automatically calibrate the displacement sensor. Third, a method for compensating the dynamics of a dynamometer to obtain force readings near the natural frequency of the dynamometer with improved accuracy at high spindle speeds. Finally, using the frequency response measurement and the cutting coefficients obtained using the methods presented previously to calculate the chatter-free regions of a machine tool.

7.1 Limitations and Future Work

The readily available load meter of machine tools was used to measure the cutting coefficient. However many other signals, including spindle However many other signals, including spindle and axis current and axis currents are also readily available on the CNC machine tools. Since the cutting conditions can be chosen when measuring the cutting coefficients, it is possible that by combining the output of several different sensors and using several cutting tests with different cutting conditions a more accurate measurement of the cutting coefficients could be obtained. The ball screw-drives have position dependent high friction which clouds the force estimation from the drive current. However, linear motor driven axis do not have ball screw and nut, which are the largest contributors to the friction. Experience indicates that the current signals, which are directly proportional to the cutting forces, have higher bandwidth, i.e. 100 Hz in an average machine tool. Tangential, radial and axial cutting force coefficients can be identified from the current signals of linear motor drives, as opposed to obtaining only tangential cutting force coefficient from spindle motor power.

The weakest link in the smart machining system developed in this thesis is the automatic transfer function measurement unit. Due to the vertical offset between the impact and response measurement locations, the dynamic stiffnesses of tool modes differ substantially from the direct measurements which are needed to accurately predict the chatter-free regions. Positioning the response sensor on the opposite side of the tool would allow measuring the direct transfer function; however the current capacitive sensor cannot accommodate the wide range of tool diameters if placed at a fixed location opposite the hammer.

There are a couple of improvements which could be made to the Kalman filter. In order to account for the periodic nature of the applied force, the applied force was decomposed into its Fourier Series, and an multivariable ordinary differential equation which generates the first N harmonics of that Fourier series was designed. This was necessary because the Kalman filter was designed as a continuous-time Kalman filter, and there is no differential equation which generates a time delay necessary to implement the true equation for a periodic equation

$$F_a(t) = F_a(t - T). \tag{7.1}$$

If the Kalman filter was designed as a discrete-time Kalman filter however, it is trivial to write a difference equation for equation 7.1. Another area for improvement is the design of the Kalman filter. Using the approach presented in this thesis, a new Kalman filter must be designed for each spindle speed. The model could be extended to include the spindle speed as an estimated state. This would make the model non-linear, so a non-linear extension of the Kalman filter such as the Extended Kalman Filter or Unscented Kalman Filter would be required.

Bibliography

[1] Y. Altintaş. Prediction of cutting forces and tool breakage in milling from feed drive current measurements. *Journal of Engineering for Industry*, 1992.

[2] Y. Altintaş and A.A. Ber. Manufacturing automation: metal cutting mechanics, machine tool vibrations, and CNC design. *Applied Mechanics Reviews*, 54:B84, 2001.

[3] Y. Altintaş and E. Budak. Analytical prediction of stability lobes in milling. *CIRP Annals - Manufacturing Technology*, 44(1):357 – 362, 1995.

[4] E. Budak, Y. Altintaş, and E. J. A. Armarego. Prediction of milling force coefficients from orthogonal cutting data. *Journal of Manufacturing Science and Engineering*, 118(2):216–224, 1996.

[5] G. Byrne, D. Dornfeld, I. Inasaki, G. Ketteler, W. König, and R. Teti. Tool condition monitoring (TCM)the status of research and industrial application. *CIRP Annals-Manufacturing Technology*, 44(2):541–567, 1995.

[6] J. Chae and S.S. Park. High frequency bandwidth measurements of micro cutting forces. *Int. J. of Machine Tools and Manufacture*, 47:1433–1441, 2007.

[7] N. Constantinides and S. Bennett. An investigation of methods for the on-line estimation of tool wear. *International Journal of Machine Tools and Manufacture*, 27(2):225 – 237, 1987.

[8] L. Deshayes, L. Welsch, A. Donmez, R. Ivester, D. Gilsinn, R. Rhorer, E. Whitenton, and F. Potra. Smart machining systems: Issues and research trends. In *Innovation in Life Cycle Engineering and Sustainable Development*, pages 363–380. Springer Netherlands, 2006.

[9] D. J. Ewins. *Modal Testing: Theory, Practice and Application (Mechanical Engineering Research Studies Engineering Design Series).* Taylor & Francis Group, February 2003.

[10] Fanuc. *Fanuc Open CNC: FOCAS1/FOCAS2 CNC/PMC Data window library.*

[11] Fanuc. *Fanuc Servo Guide*, 2001-2006.

[12] V. Giurgiutiu and A. N. Zagrai. Embedded self-sensing piezoelectric active sensors for on-line structural identification. *Transactions - American Society of Mechanical Engineers Journal of Vibration and Acoustics*, 124(1):116–125, 2002.

[13] G.C. Goodwin, S.F. Graebe, and M.E. Salgado. *Control system design.* Prentice Hall Upper Saddle River, NJ, 2001.

[14] Y.H. Jeong and D.W. Cho. Estimating cutting force from rotating and stationary feed motor currents on a milling machine. *International Journal of Machine Tools and Manufacture*, 42(14):1559 – 1566, 2002.

[15] J. Kiefer. Sequential minimax search for a maximum. *Proceedings of the American Mathematical Society*, 4(3):502–506, 1953.

[16] T.Y. Kim and J. Kim. Adaptive cutting force control for a machining center by using indirect cutting force measurements. *International Journal of Machine Tools and Manufacture*, 36(8):925 – 937, 1996.

[17] M. Tse and R. Bilne. Design of an Automated Frequency Response Test Unit. M.Eng Report, 2003.

[18] Y. Oh, W. Kwon, and C. Chu. Drilling torque control using spindle motor current and its effect on tool wear. *International Journal of Advanced Manufacturing Technology*, 24(5-6):327 – 334, 2004.

[19] S.S. Park and Y. Altintaş. Dynamic compensation of spindle integrated force sensors with kalman filter. *Journal of Dynamic Systems, Measurement, and Control*, 126(3):443–452, 2004.

[20] J. S. Popovics, W. Song, J. D. Achenbach, J. H. Lee, and R. F. Andre. One-sided stress wave velocity measurement in concrete. *Journal of Engineering Mechanics*, 124(12):1346–1353, 1998.

[21] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, UK, 1st edition, 1988.

[22] M. Santochi, G. Dini, G. Tantussi, and M. Beghini. A sensor integrated tool for cutting force monitoring. *ANNALS-CIRP*, 46:49–52, 1997.

[23] J.-H. Tong, T.-T. Wu, and C.-K. Lee. Fabrication of a piezoelectric impact hammer and its application to the *In-situ* nondestructive evaluation of concrete. *Japanese Journal of Applied Physics*, 41(Part 1, No. 11A):6595–6600, 2002.

Appendix A

Automatic Hammer CNC Integration Source Code

```
// PositionTool.cpp : Defines the entry point for the console application.
//

#include <tchar.h>
#include <stdio.h>
#include <stdlib.h>
#include <30i/fwlib32.h>
#include <malloc.h>

#include <limits>
#include <cassert>
#include <algorithm>
#include <numeric>
#include <vector>

#include <NIDAQmx.h>

#define WIN32_LEAN_AND_MEAN
#include <windows.h>

#define PROG_NUM ""
```

```
using namespace std;

int cur_axes;

void read_address(char *address, unsigned short *port)
{
        char line[80];
        do {
                printf("CNC␣address:␣");
                fgets(line, 80, stdin);
        } while (sscanf(line, "%[^:/?#@!$&'()*+,;=␣]:%hd␣",
           address, port) != 2);
}

void read_channel(char *channel)
{
        char line[80];
        do {
                printf("Cap␣sensor␣channel:␣");
                fgets(line, 80, stdin);
        } while (sscanf(line, "%s␣", channel) != 1);
}

short run_cmd(short fhnd, const char *line)
{
        short ret;
        long len, n;

        len = (long)strlen(line);
        while(len) {
                n = len;
                ret = cnc_dnc2(fhnd, &n, (char *)line);

                switch(ret)
                {
                case EW_BUFFER:
                        continue;
                case EW_OK:
                        len -= n;
                        line += n;
                        break;
                case EW_RESET:
                        ret = cnc_dncend2(fhnd, DNC_CANCEL);
                        if (ret) return ret;
                        return EW_RESET;
                }
        }

        return EW_OK;
}

void wait_for_axis_move(short hndl)
{
        bool is_done;
        ODBAXIS axis_dist;

        Sleep(200);
```

```
    do {
        cnc_distance(hndl, ALL_AXES, sizeof(ODBAXIS), &
           axis_dist);

        is_done = true;
        for (int i = 0; i < cur_axes; ++i)
        {
            if (axis_dist.data[i])
            {
                is_done = false;
                break;
            }
        }
    } while (!is_done);
}

char channel[80];

double measure_dist()
{
    int32 ai_ret;
    TaskHandle ai_task;

    ai_ret = DAQmxCreateTask("DistanceRead", &ai_task);

    if (ai_ret)
    {
        return numeric_limits<double>::signaling_NaN();
    }
```

```
ai_ret = DAQmxCreateAIVoltageChan(ai_task,
        channel,
        NULL,
        DAQmx_Val_Cfg_Default,
        -10.0,
        10.0,
        DAQmx_Val_Volts,
        NULL);

if (ai_ret)
{
        return numeric_limits<double>::signaling_NaN();
}

ai_ret = DAQmxCfgSampClkTiming(ai_task,
        NULL,
        50000,
        DAQmx_Val_Rising,
        DAQmx_Val_FiniteSamps,
        512);

if (ai_ret)
{
        return numeric_limits<double>::signaling_NaN();
}

ai_ret = DAQmxExportSignal(ai_task, DAQmx_Val_SampleClock,
    ”PFI7”);

if (ai_ret)
```

```
        {
                return numeric_limits<double>::signaling_NaN();
        }

        double sample_array[512];
        long samples_read;

        ai_ret = DAQmxReadAnalogF64(ai_task,
                DAQmx_Val_Auto,
                -1,
                DAQmx_Val_GroupByChannel,
                sample_array,
                sizeof(sample_array)/sizeof(sample_array[0]),
                &samples_read,
                NULL);

        assert(samples_read == 512);

        double sample_avg = accumulate(sample_array,
                sample_array + 512,
                0.0)/512.0;

        DAQmxClearTask(ai_task);

        return sample_avg;
}

double measure_at(short hndl, short angle)
{
        short ret;
        char cmd[256];

        angle %= 4096;
        if (angle < 0) angle += 4096;

#if 0
        ret = cnc_wrmacro(hndl, 1133, 10, angle%4096, 0);

        if (ret)
        {
                return numeric_limits<double>::signaling_NaN();
        }
#else
        sprintf(cmd, ”#1133=%d\nM119\n”, angle);
#endif

        if (ret = run_cmd(hndl, cmd))
        {
                return numeric_limits<double>::signaling_NaN();
        }

        // need to wait a while here.

        // wait for spindle to stop
        ODBACT2 sspeed;

        Sleep(200);

        int speeds[5];
        int k = 0;
```

```
    memset(speeds, 0xcd, sizeof(speeds));
    do {
        if (ret = cnc_acts2(hndl, 1, &sspeed))
        {
            fprintf(stderr, "Error␣reading␣actual␣
               spindle␣speed:␣%d\n", ret);
            cnc_freelibhndl(hndl);
            return 1;
        }
        speeds[k++] = sspeed.data[0];
        if (k > 4) k = 0;
    } while(speeds[0] || speeds[1] || speeds[2] || speeds[3]
       || speeds[4]);

    return measure_dist();
}

const double phi = (1 + sqrt(5.0))/2.0;



void bracket_max(short hndl, short *left, short *centre, short *
   right)
{
#if 1
    double fa, fb, fc;
    short ax, bx, cx;

    ax = 0;
    bx = 32;
```

```
fa = measure_at(hndl, ax);
fb = measure_at(hndl, bx);

if (fb < fa) {
        swap(fa, fb);
        swap(ax, bx);
}

cx = bx + phi*(bx - ax);

fc = measure_at(hndl, cx);
double fu;
short u, ulim;
while (fb <= fc)
{
        double r = (bx - ax)*(fb - fc);
        double q = (bx - cx)*(fb - fa);

        double rq = r - q;
        if (cx > bx) {
                ulim = bx + 64;
        } else {
                ulim = bx - 64;
        }

        if (abs(r - q) < 1e-3)
        {
                u = cx + phi*(cx - bx);
                fu  = measure_at(hndl, u);
        } else {
```

```
u = bx - ((bx - cx)*q - (bx - ax)*r)/(r -
   q);

if ((bx - u)*(u - cx) > 0)
{
        fu = measure_at(hndl, u);
        if (fu > fc)
        {
                ax = bx;
                fa = fb;
                bx = u;
                fb = fu;
                continue;
        } else if (fu < fb) {
                cx = u;
                fc = fu;
                continue;
        }
        u = cx + phi*(cx - bx);
        fu = measure_at(hndl, u);
} else if ((cx - u)*(u - ulim) > 0) {
        fu = measure_at(hndl, u);
        if (fu > fc) {
                bx = cx;
                cx = u;
                u = cx+phi*(cx - bx);
                fb = fc;
                fc = fu;
                fu = measure_at(hndl, u);
        }
```

```
                } else if ((u - ulim)*(ulim - cx) > 0) {
                    u = ulim;
                    fu = measure_at(hndl, u);
                } else {
                    u = cx + phi*(cx - bx);
                    fu = measure_at(hndl, u);
                }
            }

            ax = bx;
            bx = cx;
            cx = u;
            fa = fb;
            fb = fc;
            fc = fu;
        }

        if (ax < bx && ax < cx)
        {
            *left = ax;
            if (bx < cx) {
                *centre = bx;
                *right = cx;
            } else {
                *centre = cx;
                *right = bx;
            }
        } else if (bx < cx) {
            *left = bx;
            if (ax < cx) {
```

```
                        *centre = ax;
                        *right = cx;
                } else {
                        *centre = cx;
                        *right = ax;
                }
        } else {
                *left = cx;
                if (ax < bx) {
                        *centre = ax;
                        *right = bx;
                } else {
                        *centre = bx;
                        *right = ax;
                }
        }

        assert(*right - *left < 4096);
#else
    double m = -11.0;
        short mx = -1;
        for (int i = 0; i < 4096; i += 32)
        {
                double measure = measure_at(hndl, i);
                if (measure > m) { m = measure; mx = i; }
        }

        *left = mx - 32;
        *centre = mx;
        *right = mx + 32;
```

```
#endif
}

short find_max(short hndl, short left, short centre, short right)
{
        short x1, x2;
        short x0 = left;
        short x3 = right;

        const double r = phi - 1.0, c = 1.0 - r;

        if (abs(right - centre) > abs(centre - left))
        {
                x1 = centre;
                x2 = centre + c*(right - centre);
        }
        else
        {
                x2 = centre;
                x1 = centre - c*(centre - left);
        }

        double f0, f1, f2, f3;
        f1 = measure_at(hndl, x1);
        f2 = measure_at(hndl, x2);

        while (abs(x3 - x0) > 5)
        {
                if (f2 > f1)
                {
```

```
            x0 = x1;
            x1 = x2;
            x2 = r*x1 + c*x3;
            f0 = f1;
            f1 = f2;
            f2 = measure_at(hndl, x2);
        }
        else
        {
            x3 = x2;
            x2 = x1;
            x1 = r*x2 + c*x0;
            f3 = f2;
            f2 = f1;
            f1 = measure_at(hndl, x1);
        }
    }

    if (f1 > f2)
    {
        return x1;
    }
    else
    {
        return x2;
    }
}

template <class T>
inline T sign(const T &t)
```

```
{
        if (t < T()) return T(-1); else return T(1);
}

double calibrate_and_zero(short hndl)
{
        double delta_dist;
        double prev_dist = measure_dist();



        if (prev_dist > -0.1)
        {
                // already zeroed -- need to find some other way
                   to calibrate
                delta_dist = -0.1;
        }
        else if (prev_dist < -5)
        {
                delta_dist = 0.1;
        }
        else
        {
                delta_dist = 0.05;
        }

        double dist;
        double gain = 0.0;
        char cmd[256];
        short ret;
        do {
                sprintf(cmd, "X%.3f\n", delta_dist);
                run_cmd(hndl, cmd);

                wait_for_axis_move(hndl);

                dist = measure_dist();
                if (abs(prev_dist) < 9.5)
                {
                        gain = abs(delta_dist/(dist - prev_dist));
                            // mm/V

                        delta_dist = -dist*gain/2;

                        if (abs(delta_dist) > 0.10)
                        {
                                delta_dist = sign(delta_dist)*0.1;
                        }
                }
                else
                {
                        delta_dist = -0.1*sign(dist);
                }

                prev_dist = dist;
        } while (abs(dist) > 0.1);

        return gain;
}
```

```
int _tmain(int argc, _TCHAR* argv[])
{
    char address[256];
    unsigned short port;
    unsigned short hnd;
    short ret;
    ODBSYS sysinfo;
    int max_axes;

    read_address(address, &port);
    read_channel(channel);

    if (ret = cnc_allclibhndl3(address, port, 10, &hnd))
    {
        fprintf(stderr, ”Error␣number␣%hd\n”, ret);
        return ret;
    }

    if (ret = cnc_sysinfo(hnd, &sysinfo))
    {
        fprintf(stderr, ”Error␣number␣%hd\n”, ret);
        cnc_freelibhndl(hnd);
        return ret;
    }

    printf(”Controlled␣axes:␣%.2s/%d␣(current/max)\n”, sysinfo
       .axes, sysinfo.max_axis);
    max_axes = sysinfo.max_axis;
    if (sysinfo.axes[0] >= ’0’ && sysinfo.axes[0] <= ’9’)
    {
        cur_axes = 10*(sysinfo.axes[0] - '0') + (sysinfo.
            axes[1] - '0');
}
else
{
        cur_axes = sysinfo.axes[1] - '0';
}

if (cur_axes < 0 || cur_axes > max_axes)
{
        fprintf(stderr, "Invalid␣number␣of␣controlled␣axes
            :␣%d\n", cur_axes);
        cnc_freelibhndl(hnd);
        return 1;
}
printf("Controlled␣axes:␣%d/%d␣(current/max)\n", cur_axes,
    max_axes);

printf("Kind␣of␣CNC:␣␣␣␣␣%.2s", sysinfo.cnc_type);
if ((sysinfo.addinfo & 0x02) == 0x02) {
        printf("i");
}
switch(sysinfo.mt_type[1])
{
case 'M':
        if (sysinfo.mt_type[0] == 'M')
        {
                printf("␣2-path␣machining␣centre\n");
        }
        else
        {
                printf(”␣machining␣centre\n”);
        }
        break;
case ’T’:
        switch (sysinfo.mt_type[0])
        {
        case ’T’:
                printf(”␣2/3␣path␣lathe\n”);
                break;
        case ’M’:
                printf(”␣millturn\n”);
                break;
        case ’␣’:
                printf(”␣lathe\n”);
                break;
        }
        break;
case ’P’:
        printf(”␣punch␣press\n”);
        break;
case ’L’:
        printf(”␣laser\n”);
        break;
case ’W’:
        printf(”␣wire␣cut\n”);
        break;
default:
        printf(”␣unknown\n”);
}
```

```
printf("Series:␣␣␣␣␣␣␣␣␣␣%.4s\n", sysinfo.series);
printf("Version:␣␣␣␣␣␣␣␣␣%.4s\n", sysinfo.version);

while (1)
{
        char prognum[256];

        while (1)
        {
                /* wait for DNC connect */
                ret = cnc_dncstart2(hnd, prognum);
                switch (ret)
                {
                case EW_REJECT:
                        continue;
                case 0:
                        break;
                default:
                        fprintf(stderr, "Error␣while␣
                           waiting␣for␣DNC␣connect:␣%d\n",
                            ret);
                        if (ret == -1) {
                                char buf[32];
                                printf("DNC␣session␣in␣
                                   progress.␣␣Force␣
                                   takeover␣(y/n)?␣");
                                fgets(buf, sizeof(buf),
                                   stdin);
                                if (!stricmp(buf, "y\n"))
                                {
                                        cnc_dncend2(hnd,
                                           DNC_READERR);
                                        continue;
                                }
                        }
                        cnc_freelibhndl(hnd);
                        return 1;
                }

                if (strcmp(prognum, PROG_NUM))
                {
                        fprintf(stderr, "Unrecognized␣
                           program␣number␣%s.\n", prognum)
                           ;
                        cnc_dncend2(hnd, DNC_OPENERR);
                        continue;
                }

                short left, centre, right;

                const short nteeth = 4;

                bracket_max(hnd, &left, &centre, &right);

                short low_pos = find_max(hnd, left, centre
                   , right);
                if (low_pos <= −9.5) {
                        printf("Error:␣cutter␣not␣in␣range
                            ␣of␣displacement␣sensor.\n");
```

```
            goto prog_end;
}

if (run_cmd(hnd, ”G91\n”) == EW_RESET)
    continue;
if (run_cmd(hnd, ”G01F60Z−8.73\n”) ==
    EW_RESET) continue;

wait_for_axis_move(hnd);

left = low_pos;
right = low_pos + 4096/nteeth;
centre = (left + right)/phi;

short high_pos = find_max(hnd, left,
    centre, right);

if (run_cmd(hnd, ”Z8.73\n”) == EW_RESET)
    continue;

wait_for_axis_move(hnd);

char cmd[256];
sprintf(cmd, ”#1133=%d\nM119\n”, 2*
    high_pos − low_pos);
if (run_cmd(hnd, cmd) == EW_RESET)
    continue;

double gain = calibrate_and_zero(hnd);
```

```
            sprintf(cmd, ".\\MalTF.exe␣/c␣autohammer.
                cfg␣/set␣OSensor[0].Gain=%g", gain
                /1000.0);

            system(cmd);


prog_end:
            if (run_cmd(hnd, "M02\n%\n") == EW_RESET)
                continue;

            cnc_dncend2(hnd, DNC_NORMAL);
        }
    }

    return 0;
}
```