AN ABSTRACT OF THE THESIS OF

Christopher Wilfred Indrarto for the degree of Master of Science in Mechanical Engineering presented on July 15,2022

Title: Active and Passive Structural Vibration Mitigation for High-Speed Machine Tools.

Abstract approved:

Burak Sencer

This thesis presents novel feedback and feedforward control system design strategies for active to passive vibration mitigation of residual vibrations on high speed machine tools. Residual vibrations on high-speed machine tools are triggered when the machine axes (table) undergo large accelerations, which induce sudden inertial forces and excite the lightly damped structural modes of the machine tool structure. The proposed active vibration mitigation design approach utilizes an accelerometer, and feed the spindle (tool-tip) acceleration and velocity back to the motion controller to dampen them. On the other hand, the passive vibration mitigation approach shapes the frequency spectrum of reference motion trajectories (commands) to avoid triggering residual vibrations of the feed drive transmission system. Design of the proposed active and passive techniques require accurate modeling of the machine tool's structural dynamics. Hence, time and frequency domain identification methods are presented. In order to facilitate a practical tuning strategy, a convex-optimization based iterative tuning approach is also presented. The proposed active and passive vibration mitigation design method are tested experimentally and show significant improvements in the command tracking and vibration suppression performance.

Active and Passive Structural Vibration Mitigation for High-Speed Machine Tools.

by
Christopher Wilfred Indrarto

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented July 15, 2022
Commencement June 2023

Master of Science thesis of Christopher Wilfred Indrarto presented on July 15, 2022

APPROVED:

Major Professor, representing Mechanical Engineering

Head of the School of Mechanical, Industrial, Manufacturing Engineering

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Christopher Wilfred Indrarto, Author

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor, Dr. Burak Sencer, for providing guidance and support throughout my studies and research.

I would also like to thank my colleagues at the Manufacturing and Process Control Laboratory, as well as the faculty and staff of Mechanical and Manufacturing Engineering at Oregon State University.

Finally, I would like to thank my family for their continuous support throughout my studies.

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

Page

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

Figure Page

LIST OF TABLES

1. Introduction

Computer Numerical Control (CNC) milling machines are commonly used in manufacturing industries. They are able to produce consistent quality products with very tight geometric tolerances. However, their production speed is often limited by the desired accuracy of the product, i.e., a higher production speeds typically need larger error tolerances. This error is due to the structural resonances of the machine components such as spindle columns, ball-screw shafts, and the machine frame itself. These structural resonances dictate the upper limit of the achievable control bandwidth. In addition, vibration modes can be excited by the reference acceleration profile as well as external disturbance forces, such as cutting force, which result in uneven surface on the product.

Many studies have been done to address the limitations of CNC milling machines. Proposed methods include techniques to design and implement feedforward control [1],[2] , acceleration feedback control [3], [4], [5], [6], [7], [8], [9] trajectory pre-filter [10], and combinations of multiple methods [11],[12]. Most studies utilized the methods mentioned above to increase the motor or workpiece tracking performance [1], [3], [10], vibration damping of a specific structural resonance [5], [7], [11], and robustness against external disturbances [5]. However, few studies have been done regarding the relative vibrations between multiple structural resonances. These relative vibrations between certain structural resonances, especially between work piece and cutting tool as shown in Figure 1, directly affect the product’s surface roughness. Hence, additional studies on minimizing relative vibrations are needed.

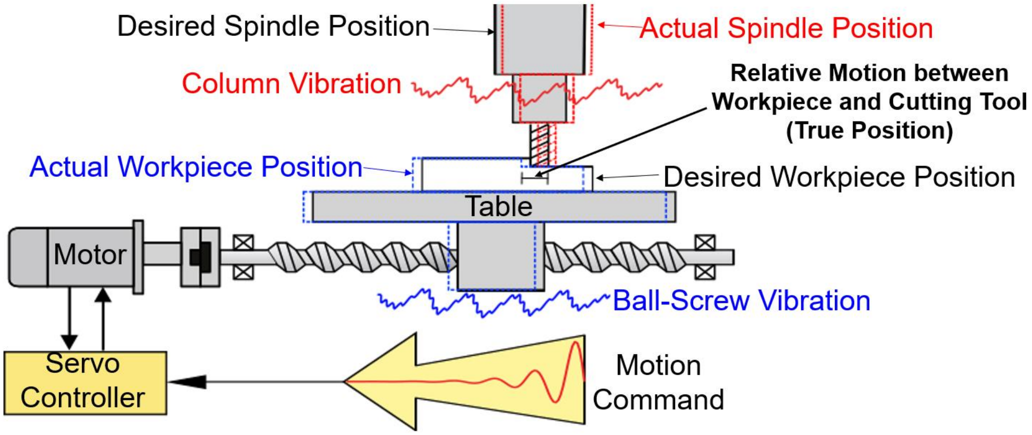

Figure 1. Illustration of workpiece – cutting tool relative vibration on NC Milling Machine

This thesis proposes novel tuning techniques for acceleration feedback and trajectory modification method to minimize the vibration of the motor, workpiece, spindle and the relative vibration between the work piece and the tool. These methods require position measurement of the tool, which can be achieved either by placing accelerometers or by installing a laser sensor on the spindle column. Systematic frameworks that utilize frequency domain and pole placement approaches to tune acceleration feedback, as well as optimization algorithm to tune the optimal trajectory pre-filter are presented. Furthermore, stability analysis on optimization algorithm is also presented.

The remainder of this thesis is divided into 5 chapters. Chapter 2 consists of a literature review on some previous studies on this research topic such as feedforward control, acceleration feedback control, trajectory pre-filter and combinations of those methods. Chapter 3 explains dynamic modelling of the machine both utilizing frequency and time domain approaches. This explanation is followed by acceleration feedback design in chapter 4. Then, trajectory pre-compensation design method is presented on chapter 5, followed by iterative tuning stability analysis on chapter 6. Then, conclusion is presented on chapter 7.

2. Literature Review

2.1. Introduction

Consider a commonly used position feedback controller used in industrial settings:

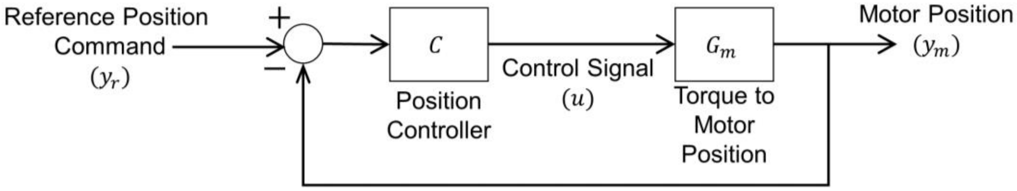

Figure 2. Common Feedback Controller

where C is the controller and G_m is the representation of the machine or torque to motor dynamics. A considerable amount of research has been done modify the common controller to reduce a specific structural vibration—which originated from the spindle column or ball screw drive. Various modification methods, ranging from active to passive compensation techniques have been implemented to the control scheme to mitigate structural vibration. These methods include acceleration feedback control [3], [4], [5], [6], [7], [8], [9] feedforward control [1], [2] , trajectory pre-filter [10] and combinations of these methods [11], [12]. This section of the thesis focuses on studies that has been done on these methods.

2.2. Acceleration Feedback Control

Acceleration feedback is one of the most commonly used methods used to mitigate structural vibration as it modifies the feedback controller which adds robustness against changes on the system. It requires acceleration measurement data from the structure of interest, which is usually obtained through accelerometers or derivation of position encoder measurement. Figure 3 represents the common control scheme for acceleration feedback.

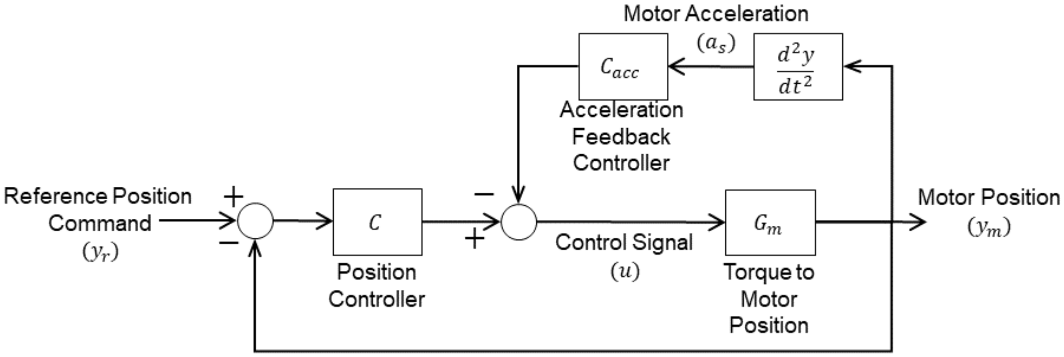

*Figure 3***.** *Common Acceleration Feedback Control Scheme*

where C_{acc} is the acceleration feedback controller. Through actively changing the virtual mass of the system based on the real-time acceleration signal, the modified controller is able to reduce the effect of disturbance forces. However, high controller bandwidth is usually required to avoid phase lag and ensure the overall controller is stable with respect to its stability margins. To compensate for this limitation, Butler [3] utilized a loop shaping method to tune and split the acceleration feedback controller. Butler’s proposed method allows the performance of closed loop position controller to be minimally affected by the acceleration feedback at all frequency while having increased disturbance rejection at selected frequencies. Figure 4 shows the proposed method by Butler.

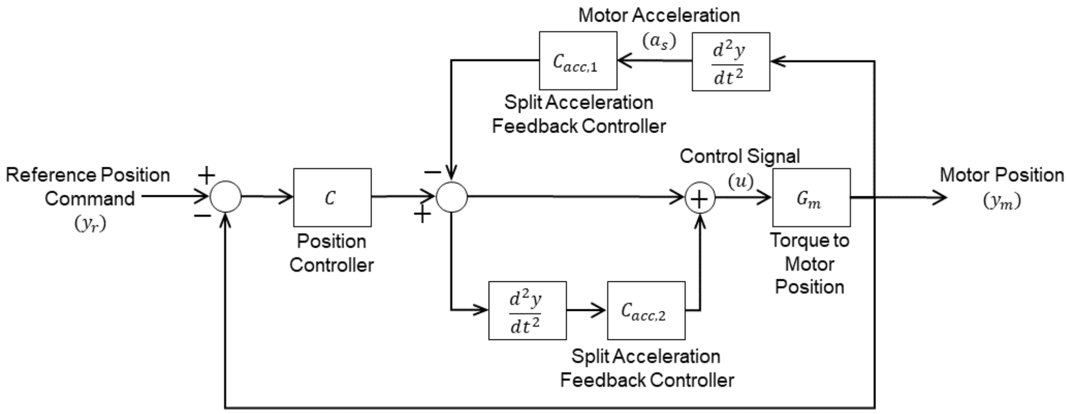

Figure 4. Butler’s Acceleration Feedback Control Scheme

Other types of acceleration feedback methods also existed in the literature such as the method to mitigate the relative vibration between 2 structures by Kai [9] and the acceleration feedback method where the feedback signal is taken from the load side instead of the motor side by Marushita [8]. Despite being robust to slight changes in the system, the stability of the existing controller is still affected by the addition of acceleration feedback, which might lead it to instability. In contrast, there are other methods which sacrifice the robustness of the controller in exchange for retaining the stability of the original controller, namely feedforward controls and trajectory pre-filtering techniques. By avoiding altering the feedback loop, the stability of the existing controller is maintained.

2.3. Feedforward Control

Unlike acceleration feedback, feed forward controller doesn’t modify the feedback loop and thus the stability of the original controller is preserved. The main idea of feedforward control is to achieve accurate tracking performance or passive vibration mitigation by compensating for a known system behavior [13]. Therefore, feedforward control generally requires an accurate model of the system which is shown in Eq(1). Figure 5 shows the general feedforward control scheme.

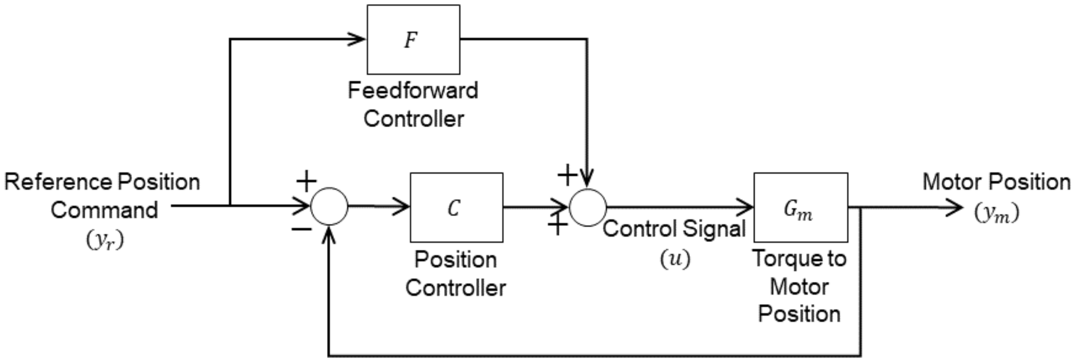

Figure 5. Common Feedforward Control Scheme

$$\frac{y_m}{y_r} = \frac{(c+F)G_m}{1+CG_m} = 1$$
$$F = \frac{1}{G_m} \tag{1}$$

where F is the feed forward controller. To construct the optimal feed forward F on a system with multiple resonances, Hosseinabadi *et al.* [5] proposed a high-order system identification procedure to accurately model a system with multiple resonances. Hosseinabadi's proposed procedure utilized the Gauss – Newton optimization algorithm on a high order transfer function to minimize the differences between identified model transfer function and actual system frequency response function (FRF). The proposed high order transfer function used for the identification procedure is the multiplication of a set of second order transfer function, each corresponds to every dominant resonance frequencies of the system as shown below.

$$F=\prod_{k=1}^{N}\frac{b_{0,k}s^{2}+b_{1,k}s+b_{2,k}}{s^{2}+2\zeta_{k}\omega_{n,k}s+\omega_{n,k}^{2}} \tag{2}$$

where ω_n and ζ is the natural frequency and the damping coefficient for each of the resonances respectively. $b=[b_0\, b_1\, b_2]$ is the numerator parameters which is updated by the Gauss – Newton optimization algorithm. Since the fitting algorithm is done in frequency domain, it is required to have the frequency response function of the system which can be obtained through multiple means which will be explained in section 3 of this thesis. Model parameters identified from the identification procedure are then used to construct an accurate feedforward controller.

On the other hand, van der Meulen *et al* [1] proposed a feedforward structure and tuning procedure that does not require an accurate system model, which also bypasses the need to have a frequency response function of the system. His proposed method is to feed an additional signal obtained from a combination of reference velocity, acceleration, jerk and snap, each multiplied by a constant. Figure 6 shows his proposed feedforward method. Each constant is tuned simultaneously with Newton's optimization algorithm to minimize the dynamic tracking error, which is the difference between the actual motor position and the reference position.

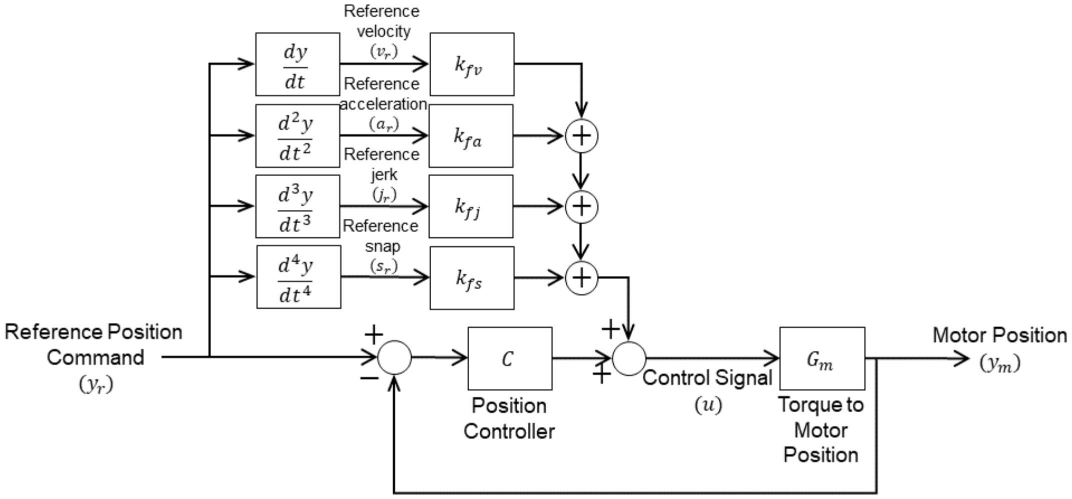

Figure 6. Meulen's Feedforward Control Scheme

In contrast to acceleration feedback control, feedforward control doesn't interfere with the stability of closed loop position controller. However, feedforward control's main idea is to compensate for errors that is both predictable and repeatable. Hence, any mismatch between the model and the actual system, as well as any changes to the system during cutting process, cannot be compensated and results in imperfect error mitigation.

2.4. Trajectory Pre-compensation

Both previous methods are sometimes not applicable in commercial machines due to the need to modify the controller directly. Some of the commercial machines don't allow the user to access and modify its controller, leaving the only modifiable parameter in the system is the reference trajectory. Therefore, instead of injecting a compensation signal directly to the torque command in feedforward control, trajectory pre-compensation modifies the reference trajectory to compensate for any predictable and repeatable error between the actual motion and the desired motion. Figure 7 shows the trajectory pre-filter control scheme.

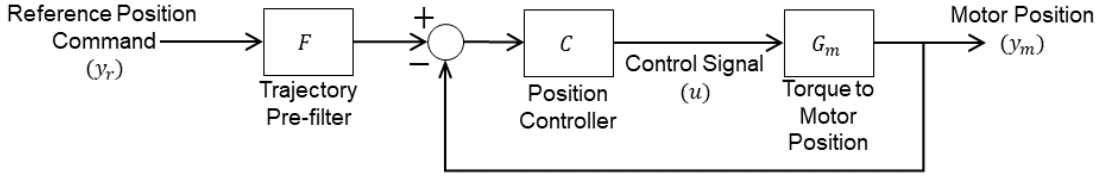

Figure 7. Trajectory Pre-filter Control Scheme

where F is the trajectory pre-filter. Dumanli *et al.* [10] proposed a trajectory pre-filter method to modify the trajectory such that the difference between motor position and reference position i.e. motor tracking error, is significantly reduced. Similar to van der Meulen's tuning method, Dumanli's proposed tuning method utilized Newton's optimization algorithm, along with time domain experiment measurement data to obtain the optimal pre-filter parameters.

Similar to feedforward control, due to the lack of real-time feedback of the acceleration, trajectory pre-compensation method can only compensate for predictable and repeatable error. Therefore, any changes that happen to the system during the cutting process cannot be compensated and tracking error will occur.

However, unlike feedforward control, trajectory pre-compensation method doesn't require the user to modify the controller. Therefore, trajectory pre-compensation is sometimes preferred over feedforward controller especially if there is a lack of access to the machine's control system.

2.5. Conclusion

Accurate dynamic positioning can be achieved by modifying the overall control system to compensate for structural vibrations, especially low frequency vibrations

(<60Hz). A lot if control modification techniques has been proposed in literature to improve the dynamic positioning accuracy of machine. However, many of them requires accurate mathematical modelling or is not robust to external disturbances and changes on the system. Moreover, close to none of the proposed technique address the relative vibrations between multiple structural vibrations.

3. Dynamic Modelling of Torque to Motor Dynamics

3.1. Introduction

One of the most common high-speed machine tool used in industrial standards are the 3-axis CNC milling machine. Figure 8 shows an example of an actual 3-axis CNC milling machine while Figure 9 is a constructed 3D model of a 3-axis CNC milling machine.

Figure 8. Example of a 3-Axis CNC Milling Machine

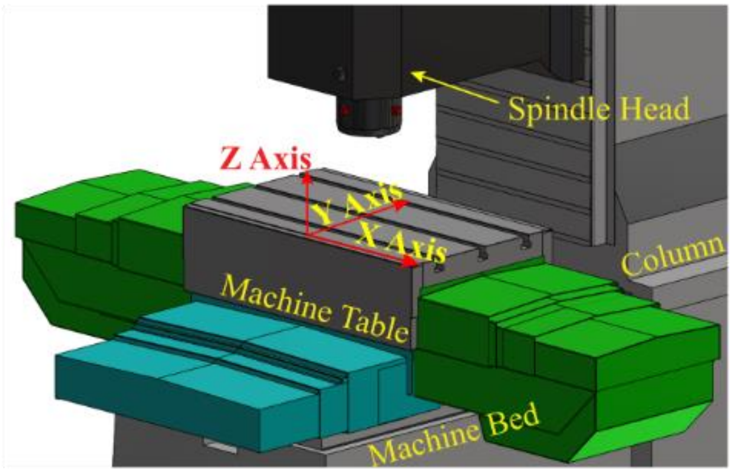

Figure 9. 3D Model of 3-Axis CNC Milling Machine

These machine tools typically utilize ball screw driven mechanism to drive its axis. Figure 10 shows the visual representation of a ball screw drive mechanism.

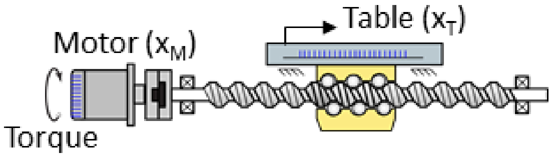

Figure 10. Ball Screw Drive Mechanism

As torque command is sent to the motor, the motor will rotate which rotates the gear box and the ball screw drive. The ball screw translates the rotary motion to linear motion of the table. This mechanism is controlled by a closed loop controller to ensure the system moves as desired.

The design procedure of control system often requires accurate mathematical model of the machine as it has direct effects to the controller performance as well as the stability of the controller with respect to its margins. The most common way of representing the system behavior is through a Frequency Response Function (FRF) [14]. In this chapter, systematic procedure to obtain FRF of a CNC machine, as well as transfer function identification algorithm in time and frequency domain will be explained.

The basic multi degree of freedom model structure will be explained in section 3.2. Although there are a few methods to tune the mathematical model of the machine, 2 methods will be explained in this thesis; frequency domain identification that utilize the frequency response function (FRF) in section 3.3.1. and time domain identification in section 3.3.2. Then, the experimental results of the aforementioned methods will be presented in section 3.4.

3.2. Multi Degree of Freedom (MDOF) Model

Most high-speed motion system utilized the ball screw drive systems. However, ball screw drives mechanism suffers from translational and torsional vibration modes. These vibrations are mainly caused by the flexibility of the ball screw shaft, bearings, and couplings. Since the torsional vibration mode is typically on a much higher frequency than the translational mode, it is usually attenuated by the controller. On the other hand, translational mode typically dictates the stability bandwidth which limit the dynamic positioning accuracy of the system.

Since the first vibrational modes contributes the most to the limitation of control bandwidth, a popular model used to estimate the machine's behavior is a simple 2 degree of freedom (DOF) transfer function [15]. Figure 11 shows the simplified two mass spring system of a ball screw drive. Eq(3) shows the equation of motion for Figure 11.

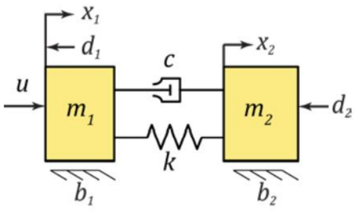

Figure 11. Two mass spring model of a flexible ball screw drive

$$\begin{aligned} m_1\ddot{x}_1 &= -b_1\dot{x}_1 + k\left(x_2 - x_1\right) + c\left(\dot{x}_2 - \dot{x}_1\right) + u + d_1 \\ m_2\ddot{x}_2 &= -b_2\dot{x}_2 + k\left(x_1 - x_2\right) + c\left(\dot{x}_1 - \dot{x}_2\right) + d_2 \end{aligned} \tag{3}$$

where m_1 and m_2 represent the equivalent motor and table inertias, k is the stiffness of the ball screw drives, b_1 and b_2 are the viscous friction on the motor and table respectively, and c is the viscous friction of the ball screw. u is the input to the motor, which can be in voltage or torque. d_1 and d_2 are disturbance forces applied on the motor and table respectively. This spring mass model can be further simplified under 2 assumptions; the model only operates on low frequencies and the controller feedback will be taken from the motor side. Under these 2 assumptions, the ball-screw system can be modelled as a rigid body mass translating on a viscous medium. This results in a transfer function representation below.

$$u = m_1\ddot{x}_1 + b_1\dot{x}_1$$
$$G_m = \frac{x_1}{u} = \frac{1}{m_1 s^2 + b_1 s} \tag{4}$$

The transfer function presented in Eq(4) is used to model the system. Identification procedure is required to obtain the correct transfer function parameters. Hence, the next section explains 2 commonly used method for transfer function parameter identification method; frequency domain and time domain identification.

3.3. Parameter Identification of Torque to Motor Dynamics Transfer Function

3.3.1. Frequency Domain Identification

Frequency Response Function identification, which can be commonly defined as FRF identification, is an identification technique that focuses on constructing and tuning transfer function parameters which has a very similar frequency response to that of the actual system [14]. Frequency response of a system is a quantitative measure of how the system will behave, both in magnitude and phase, as a function of frequency. To simplify the definition, it is the input to output relationship in terms of magnitude and phase of the system for any given frequencies. Figure 12 presents a systematic framework of how a frequency response of a system can be obtained.

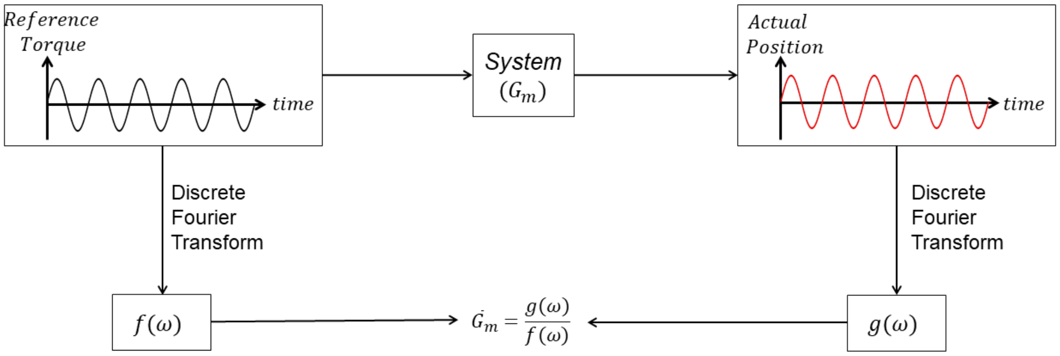
Reference Torque
time
System
(G_m)
Actual Position
time
Discrete Fourier Transform
Discrete Fourier Transform
$f(\omega)$
$\dot{G}_m = \frac{g(\omega)}{f(\omega)}$
$g(\omega)$

Figure 12. Procedure of obtaining the System's FRF

By sending sine waves of multiple frequencies within a frequency range, especially with considerably small frequency step (e.g. every 1 Hz from 5 Hz to 40 Hz), the input and output relationship for each frequency steps can be obtained. Hence, FRF of the system within that frequency range can be constructed. Figure 13 shows an example FRF of a commercial machine's spindle dynamics.

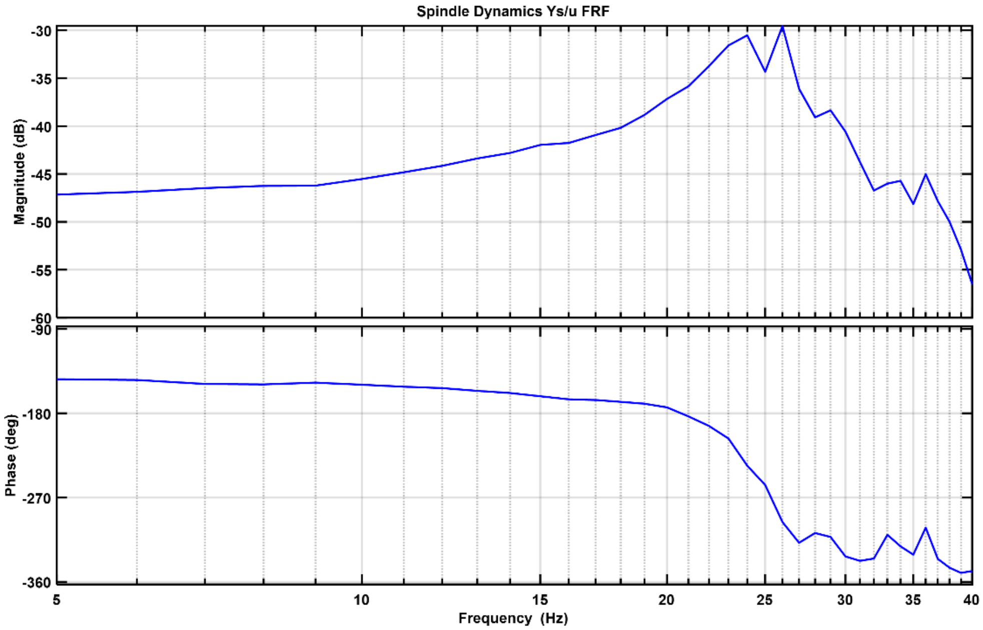

Figure 13. FRF of a CNC Milling Machine's Spindle Dynamics

In addition to obtaining the FRF of the machine, mathematical models of the system is generally required for controller tuning procedure and simulating any control system design algorithm. Therefore, in this section, a frequency domain transfer function tuning approach utilizing the least squares algorithm is explained.

First, the general form of a system's FRF can be written in the form of complex numbers with respect to its frequency such as in Eq(5).

$$G_m(j\omega_{i=m,\ldots,M}) = \begin{pmatrix} \mathrm{Re}\left[G_m(j\omega_m)\right] + j\,\mathrm{Im}\left[G_m(j\omega_m)\right] \\ \vdots \\ \mathrm{Re}\left[G_m(j\omega_M)\right] + j\,\mathrm{Im}\left[G_m(j\omega_M)\right] \end{pmatrix} \tag{5}$$

where ω_m and ω_M denote the first and last frequency in the identified frequency range. $Re(G)$ and $Im(G)$ denote the real and imaginary components of the complex number

G respectively. Then, similar to [5], consider the transfer function in Eq(6).

$$F(s)=\prod_{k=1}^{K}\frac{s^{2}+2\zeta_{a,k}\omega_{a,k}s+\omega_{a,k}^{2}}{s^{2}+2\zeta_{r,k}\omega_{r,k}s+\omega_{r,k}^{2}},F(j\omega_{i})=\prod_{k=1}^{K}\frac{\left(j\omega_{i}\right)^{2}+2\zeta_{a,k}\omega_{a,k}\left(j\omega_{i}\right)+\omega_{a,k}^{2}}{\left(j\omega_{i}\right)^{2}+2\zeta_{r,k}\omega_{r,k}\left(j\omega_{i}\right)+\omega_{r,k}^{2}} \qquad (6)$$

where $\omega_{a,k}$ and $\omega_{r,k}$ denote the antiresonance and resonance frequencies respectively while K is the number of resonances to be identified. $\zeta_{a,k}$ and $\zeta_{r,k}$ are antiresonance and resonance damping coefficients of the transfer function respectively. To simplify the least squares algorithm, the parameters in transfer function in Eq(6) are grouped and simplified as Eq(7).

$$F(j\omega_{i})=\prod_{k=1}^{K}\frac{b_{0,k}\left(j\omega_{i}\right)^{2}+b_{1,k}\left(j\omega_{i}\right)+b_{2,k}}{\left(j\omega_{i}\right)^{2}+a_{1,k}\left(j\omega_{i}\right)+a_{2,k}} \qquad (7)$$

From this point onwards, the system is assumed to have only 1 dominant resonance to simplify the explanation of the least squares algorithm and hence $K=1$. The main idea of the least squares algorithm is to match the frequency domain response of the transfer function to the FRF of the system for every frequency i. Thus, the least squares formulation is written as Eq(8).

$$\min_{b,a} J=\frac{1}{2}\left\|Ax-B\right\|_{2}^{2} \qquad (8)$$

Then, the matrix A and B has to be constructed so that the frequency response of the fitted transfer function is very close to that of the actual system as follows:

$$\underbrace{\frac{b_{0}\left(j\omega_{m,\ldots,M}\right)^{2}+b_{1}\left(j\omega_{m,\ldots,M}\right)+b_{2}}{\left(j\omega_{m,\ldots,M}\right)^{2}+a_{1}\left(j\omega_{m,\ldots,M}\right)+a_{2}}}_{\text{Frequency Response of Fitted Transfer Function}}=\underbrace{\operatorname{Re}\left[G_{m}(j\omega_{m,\ldots,M})\right]+j\operatorname{Im}\left[G_{m}(j\omega_{m,\ldots,M})\right]}_{\text{Frequency Response of Actual System}} \qquad (9)$$

Then, Eq(9) is re-arranged to 2 separate equations; 1 for real part of the equation and the other for the imaginary part of the equation as shown in Eq(10).

$$\underbrace{\begin{gathered}\left(-b_0\omega_{m,\dots,M}^2+b_2\right)-a_2\operatorname{Re}\left[G_m(j\omega_{m,\dots,M})\right]+a_1\omega_{m,\dots,M}\operatorname{Im}\left[G_m(j\omega_{m,\dots,M})\right]=\dots\\ -\omega_{m,\dots,M}^2\operatorname{Re}\left[G_m(j\omega_{m,\dots,M})\right]\end{gathered}}_{\text{Real}}$$

(10)

$$\underbrace{\begin{gathered}\left(b_1\omega_{m,\dots,M}\right)-a_2\operatorname{Im}\left[G_m(j\omega_{m,\dots,M})\right]-a_1\omega_{m,\dots,M}\operatorname{Re}\left[G_m(j\omega_{m,\dots,M})\right]=\dots\\ -\omega_{m,\dots,M}^2\operatorname{Im}\left[G_m(j\omega_{m,\dots,M})\right]\end{gathered}}_{\text{Imaginary}}$$

These equations are then written in a matrix form to satisfy the least squares objective function written in Eq(8). Thus, the resulting least squares objective function and analytical solution can be written as Eq(11).

$$\underbrace{\begin{pmatrix}-\omega_m^2 & 0 & 1 & \omega_m\operatorname{Im}\left[G_m(j\omega_m)\right] & -\operatorname{Re}\left[G_m(j\omega_m)\right]\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ -\omega_M^2 & 0 & 1 & \omega_M\operatorname{Im}\left[G_m(j\omega_M)\right] & -\operatorname{Re}\left[G_m(j\omega_M)\right]\\ 0 & \omega_m & 0 & -\omega_m\operatorname{Re}\left[G_m(j\omega_m)\right] & -\operatorname{Im}\left[G_m(j\omega_m)\right]\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ 0 & \omega_M & 0 & -\omega_M\operatorname{Re}\left[G_m(j\omega_M)\right] & -\operatorname{Im}\left[G_m(j\omega_M)\right]\end{pmatrix}}_{A}\underbrace{\begin{pmatrix}b_0\\ b_1\\ b_2\\ a_1\\ a_2\end{pmatrix}}_{x}=\underbrace{\begin{pmatrix}-\omega_m^2\operatorname{Re}\left[G_m(j\omega_m)\right]\\ \vdots\\ -\omega_M^2\operatorname{Re}\left[G_m(j\omega_M)\right]\\ -\omega_m^2\operatorname{Im}\left[G_m(j\omega_m)\right]\\ \vdots\\ -\omega_M^2\operatorname{Im}\left[G_m(j\omega_M)\right]\end{pmatrix}}_{B}$$

$$x=A^{-1}B$$

(11)

As the number of resonances increases, so does the order of the transfer function. The formulation shown in Eq(6) can be modified to fit a higher order transfer function by increasing the variable K, which increase the number of second order transfer functions and tuning parameters.

While this method is quite effective at modelling the transfer function, obtaining the FRF of the machine is not an easy and fast task. Therefore, another transfer function identification method that doesn't require the FRF of the system is developed. In the next section, time domain identification algorithm will be explained.

3.3.2. Time Domain Identification

In contrast to frequency domain identification, time domain identification doesn't require the system's FRF to be obtained, which makes it more practical than frequency domain identification [16]. There are two ways to do time domain identification; continuous domain and discrete domain. However, both require the same set of data. Figure 14 shows the procedure of obtaining the necessary data for the identification algorithm.

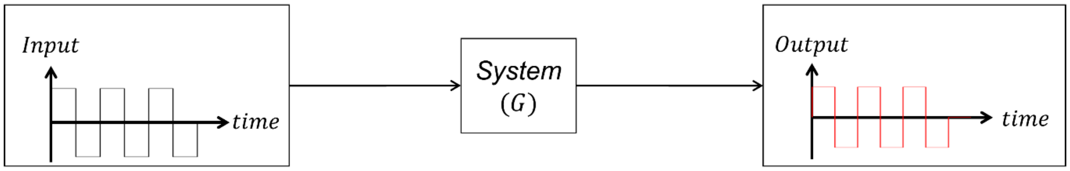

Figure 14. Experimental Procedure of obtaining the required signals for time domain identification

First, consider the machine dynamics to be identified where the input signal is motor position and the output signal is the table position as Eq(12).

$$G_t = \frac{y_{table}}{y_{motor}} \tag{12}$$

The model transfer function structure to be identified is the same as Eq(6) in continuous domain. To simplify the algorithm explanation, the system is assumed to have only one resonance, hence $K = 1$. Thus, the structure of the fitted transfer functions in continuous domain as shown in Eq(13).

$$F(s)=\frac{y_{table}}{y_{motor}}=\frac{b_0s^2+b_1s+b_2}{s^2+a_1s+a_2} \tag{13}$$

Then, the equation above is discretized with Euler's discretization method where $s=\frac{z-1}{T_s}$. All coefficients of z can be grouped and the resulting transfer function is shown below.

$$F(z)=\frac{y_{table}}{y_{motor}}=\frac{\bar{b}_0z^2+\bar{b}_1z+\bar{b}_2}{z^2+\bar{a}_1z+\bar{a}_2} \tag{14}$$

Similar to frequency domain identification, the first step is to write a least squares formulation that follows Eq(8). To achieve the same objective function equation, A and B are formulated in Eq(15) for continuous time domain identification.

$$\begin{gathered}F(s)=\frac{b_0s^2+b_1s+b_2}{s^2+a_1s+a_2}=\frac{y_{table}}{y_{motor}}\\ \left(b_0s^2+b_1s+b_2\right)y_{motor}-\left(a_1s+a_2\right)y_{table}=s^2y_{table}\end{gathered} \tag{15}$$

Then, the equation above is transformed from Laplace domain to time domain using Inverse Laplace Transform and constructed in a matrix form as below.

$$b_0\ddot{y}_{motor}+b_1\dot{y}_{motor}+b_2y_{motor}-a_1\dot{y}_{table}-a_2y_{table}=\ddot{y}_{table}$$

$$\underbrace{\begin{pmatrix}\ddot{y}_{motor}(1) & \dot{y}_{motor}(1) & y_{motor}(1) & -\dot{y}_{table}(1) & -y_{table}(1)\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ \ddot{y}_{motor}(N) & \dot{y}_{motor}(N) & y_{motor}(N) & -\dot{y}_{table}(N) & -y_{table}(N)\end{pmatrix}}_{A}\underbrace{\begin{pmatrix}b_0\\ b_1\\ b_2\\ a_1\\ a_2\end{pmatrix}}_{x}=\underbrace{\begin{pmatrix}\ddot{y}_{table}(1)\\ \vdots\\ \ddot{y}_{table}(N)\end{pmatrix}}_{B} \tag{16}$$

Discrete time domain identification has very similar procedure as continuous time domain identification. First, similar to Eq(15), the equation for discrete domain is constructed as below.

$$F(z)=\frac{\bar{b}_0 z^2+\bar{b}_1 z+\bar{b}_2}{z^2+\bar{a}_1 z+\bar{a}_2}=\frac{y_{table}}{y_{motor}} \qquad (17)$$

$$\left(\bar{b}_0+\bar{b}_1 z^{-1}+\bar{b}_2 z^{-2}\right) y_{motor}-\left(\bar{a}_1 z^{-1}+\bar{a}_2 z^{-2}\right) y_{table}=y_{table}$$

Since z^{-k} in discrete domain is a k samples delay in time domain, the equation can be constructed into a matrix form as

$$\underbrace{\begin{pmatrix} y_{motor}(n) & y_{motor}(n-1) & y_{motor}(n-2) & -y_{table}(n-1) & -y_{table}(n-2) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{motor}(N) & y_{motor}(N-1) & y_{motor}(N-2) & -y_{table}(N-1) & -y_{table}(N-2) \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ a_1 \\ a_2 \end{pmatrix}}_{x} = \underbrace{\begin{pmatrix} \ddot{y}_{table}(n) \\ \vdots \\ \ddot{y}_{table}(N) \end{pmatrix}}_{B}$$

$$(18)$$

where $n = 1, \dots, N$ and N is the length of the time domain data used for the identification algorithm. Finally, for both continuous and discrete time domain identification, the transfer function parameters can be calculated by using the equation $x = A^{-1}B$.

Note that these identification method can be used for various transfer function structure such as a second order transfer function shown in Eq(13), first order transfer function like $F(s)=\frac{1}{ms+b}$, or a higher order transfer function. However, depending on the order of the transfer function, discrete time domain identification is often preferred over continuous time domain identification. In continuous time domain identification, the higher the order of the transfer function, the more derivation and integration that needs to be done. More integration results in amplifying the drift of the measured data while derivation amplifies the noise of the measured data. Since drift and noise are amplified, the identified transfer function won't be optimal as it tries to fit those unwanted vibrations. Filtering the data before doing the identification algorithm also affect the magnitude and phase of the measured data, causing the identification to be slightly off. Therefore, discrete time domain identification is often preferred over continuous time domain identification especially on higher order transfer function fitting.

3.4. Experiment Results

This section presents the experiment results of the identification procedure, both frequency and time domain approaches. The dynamic that is fitted is the torque to motor dynamics. The structure that is used to fit the dynamics is a second order transfer function $G_m(s) = \frac{1}{m_m s^2 + b_m s}$.

Figure 15 shows the frequency response functions of X and Y axis' motor and table dynamics obtained through the identification procedure presented in section 3.3.1. Figure 16 shows the frequency domain least squares fitting performance on X and Y axis' torque to motor dynamics. The fitting is done on 5 Hz to 20 Hz of the FRF since the system model $G_m(s) = \frac{1}{m_m s^2 + b_m s}$ introduced in section 3.2. doesn't capture any resonances. Since the bandwidth frequency of the controllers used for further experiments is significantly lower than the system resonances, it is not required to have an accurate model that capture those resonances.

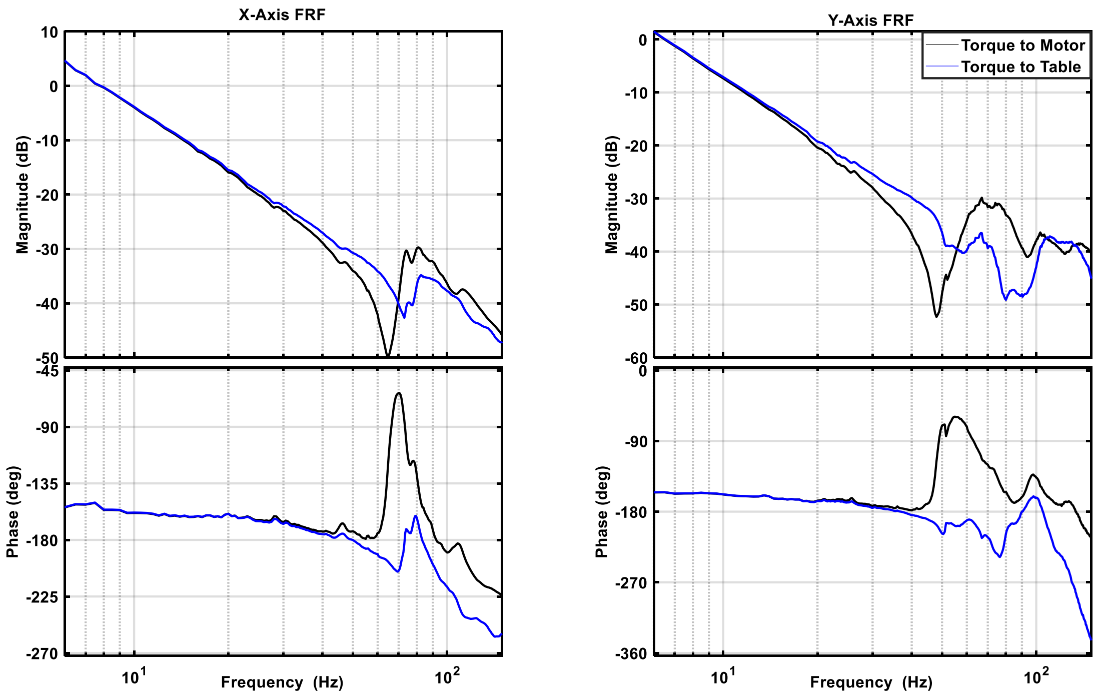

Figure 15. X and Y Axis Motor and Table Frequency Response Functions

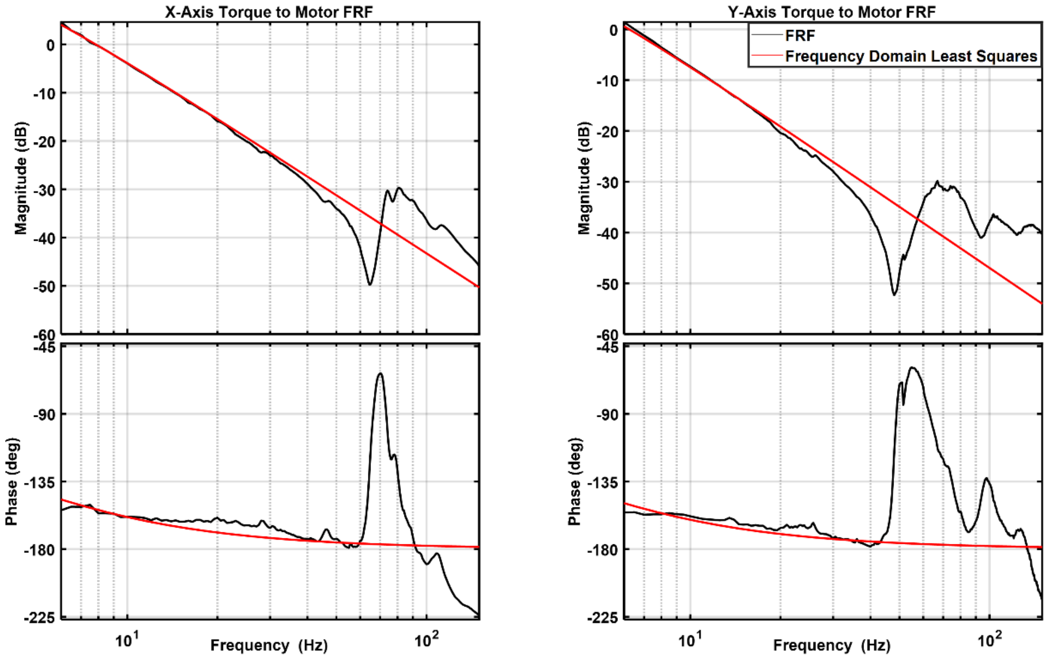

Figure 16. X and Y Axis Torque to Motor FRF and Frequency Domain Least Squares Fitting

For time domain least squares fitting, only Y-axis experiment and identification is conducted. Figure 17 shows the Y-axis time domain least squares fitting performance. Table 1 shows the identified transfer function parameters for the frequency domain identification method and Table 2 shows the identified transfer function parameters for the time domain identification method.

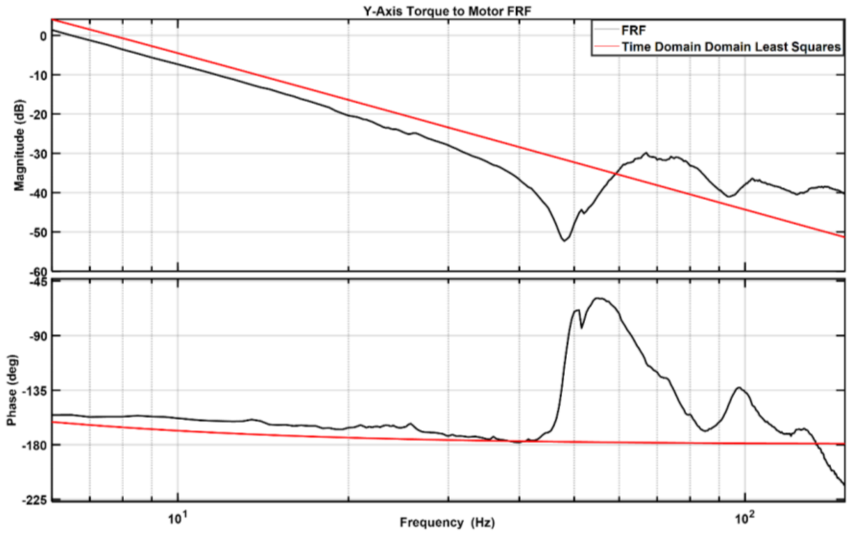

Figure 17. Y Axis Torque to Motor FRF and Time Domain Least Squares Fitting

Axis	$m_m\left(\frac{mm}{V}\right)$	$b_m\left(\frac{mm}{sV}\right)$
X	0.0004	0.009
Y	0.0006	0.0126

Table 1. Identified Parameters from Frequency Domain Least Squares Fitting

Axis	$m_m\left(\frac{mm}{V}\right)$	$b_m\left(\frac{mm}{sV}\right)$
Y	0.0004	0.0054

Table 2. Identified Parameters from Time Domain Least Squares Fitting

4. Acceleration Feedback Based Vibration Mitigation

4.1. Introduction

Improving the controller's dynamic positioning accuracy while improving its robustness to external disturbances and any changes to the machine are mostly done by actively modifying the virtual mass observed by the system at any given time. However, due to this approach is done by modifying the controller's feedback loop, a high bandwidth controller is usually required to ensure the stability of the machine when exposed to external disturbances. Furthermore, most of the proposed acceleration feedback technique in the literature are applied only on the motor side to reduce its tracking error [3], [4], [5], [6], [7]. In real manufacturing process, the vibration that is visible on the product surface is originated from the combination of spindle and table vibration. Thus, the feedback loop proposed in this thesis will take both the spindle and motor dynamics into consideration.

This section proposed a novel and systematic design and tuning procedure for acceleration feedback control that targets the vibration on the spindle and the motor while ensuring the stability of the machine.

4.2. Loop Shaping Approach

Structural dynamics of the machine tool system can be modelled considering multi-body motor/table and spindle/column dynamics. As the table accelerates back and forth during positioning, reaction forces excite the machine column and vibrate the spindle/column structure. Overall dynamics of table and spindle can be represented in Laplace domain as Eq(19).

$$G_m = \frac{y_m}{u} = \frac{1}{m_m s^2}, G_s = \frac{y_s}{u} = \frac{-1}{m_s s^2 + b_s s + k_s} \tag{19}$$

where u is the motor torque, y_m and y_s are motor and spindle position respectively. m_m, m_s, b_s and k_s are rigid body parameters of motor and spindle which can be identified using identification procedure in chapter 3.

To compensate for the relative vibration between the spindle and the table, it is required to incorporate the feedback loop from the spindle side in addition to the motor regular feedback loop. Figure 18 shows FRF of the Y-axis of a commercial cartesian CNC milling machine shown in Figure 8 to see the vibration frequency that needs to be compensated and Figure 19 shows the proposed acceleration feedback control scheme.

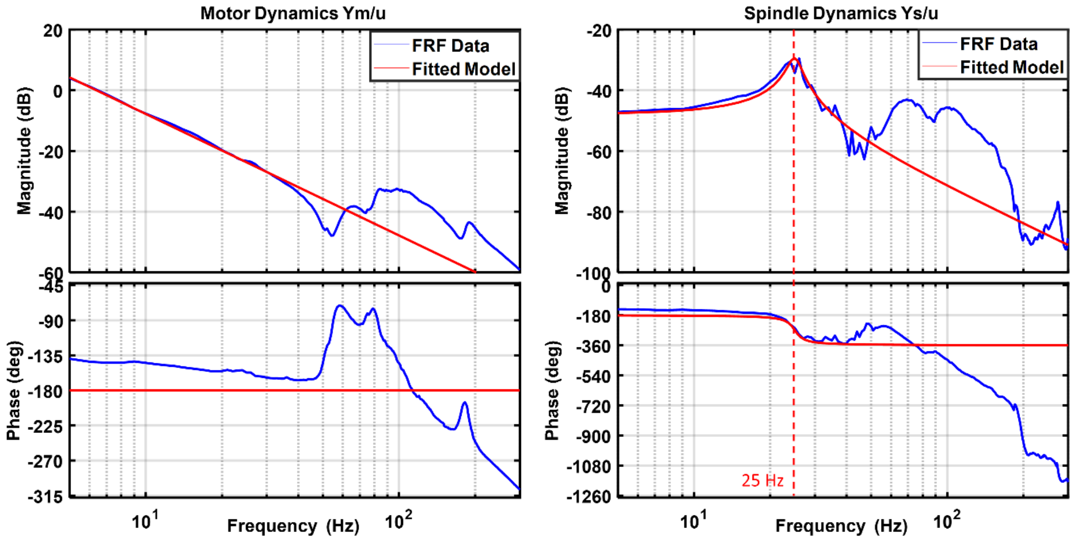

Figure 18. Frequency Response Function of Y-Axis

The position controller C in Figure 19 is an industry standard P-PI controller, which is the controller shown in Figure 20. The transfer function of the system can be postulated as Eq(20).

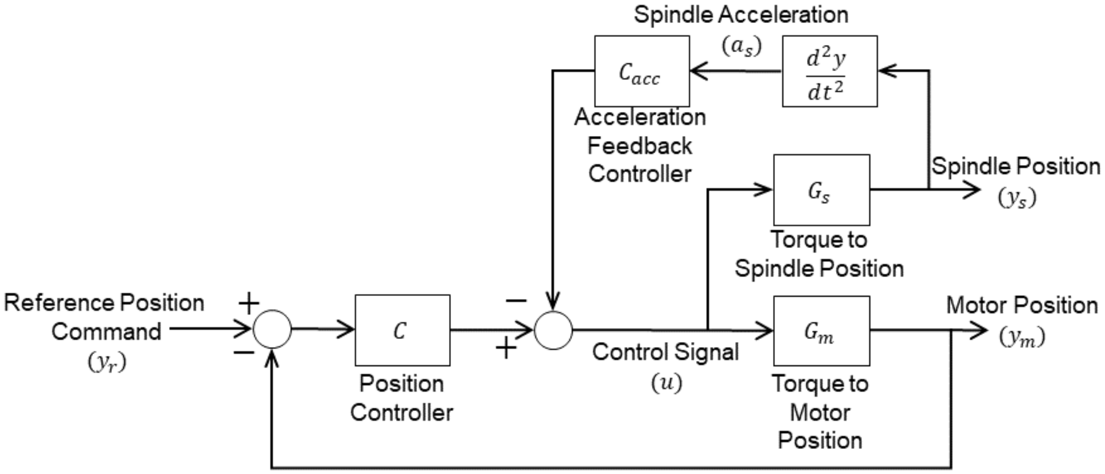

Figure 19. Proposed Acceleration Feedback Control Scheme

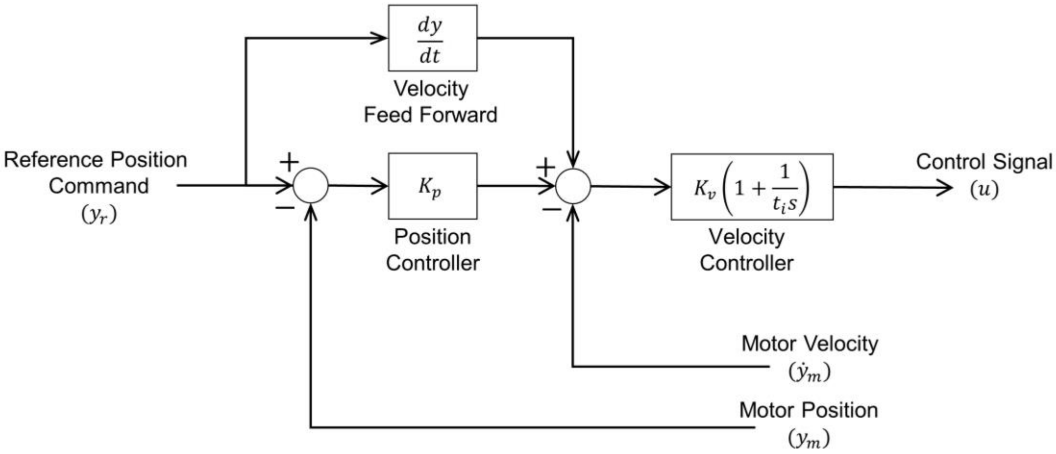

Figure 20. Industry Standard PPI Controller

where K_p, K_v and t_i are the position and velocity proportional control gain and velocity integral gain.

$$C = K_p + s\ K_v\left(1+\frac{1}{t_i s}\right)$$
$$G_m^{cls}(s) = \frac{y_m}{y_r} = \frac{G_m C}{1+G_m C - G_s C_{acc} s^2} \tag{20}$$
$$G_s^{cls}(s) = \frac{y_{spindle}}{y_r} = \frac{G_s C}{1+G_m C - G_s C_{acc} s^2}$$

In this case, the targeted low frequency vibration is 25Hz. To obtain the spindle acceleration, an accelerometer is attached to the spindle housing of the machine. Then, spindle's acceleration signal must be properly filtered by C_{acc} before being fed back to the closed loop controller to ensure the stability of the controller. Mainly, the acceleration signal must be converted back to voltage to ensure the compensation signal unit and amplitude are correct. Then, accelerometer measurement contains noise which might distort the compensation signal. These problems can be solved by a simple proportional gain on the acceleration feedback. Thus, the structure of the feedback controller is chosen to be a simple proportional gain as shown below.

$$C_{acc} = \underbrace{K_a}_{\substack{\text{Proportional}\\ \text{Gain}}} \tag{21}$$

Proportional gain K_a must be small enough to ensure that compensation signal of the acceleration feedback doesn't have a huge amplitude and also minimize the effect of noise to the desired compensation signal. This gain is tuned by trial and error while looking at the closed loop FRF of the system, which results in non-optimal control. Thus, the next section presents a more systematic approach to tune spindle dynamics feedback controller.

4.3. State Space Approach

As described before, the previous acceleration feedback method explained in Section 4.2 is not optimal due to the trial-and-error nature of the tuning procedure. This means, there exists a parameter value that will result in a better performance than that obtained in the previous method. In addition, the goal of the loop shaping acceleration feedback method in the previous method is to compensate for the spindle vibration. However, the goal of this section is to reduce the relative motion between the table and spindle, which is called true error. Thus, to be able to analytically obtain the optimal controller parameters that will reduce the relative motion between the table and spindle, a new systematic approach utilizing the state space representation is proposed in this section.

To formulate the overall system in a state space representation, a simplified system representation is constructed as shown in Figure 21 to obtain the correct equation of motion that represents the actual system behavior.

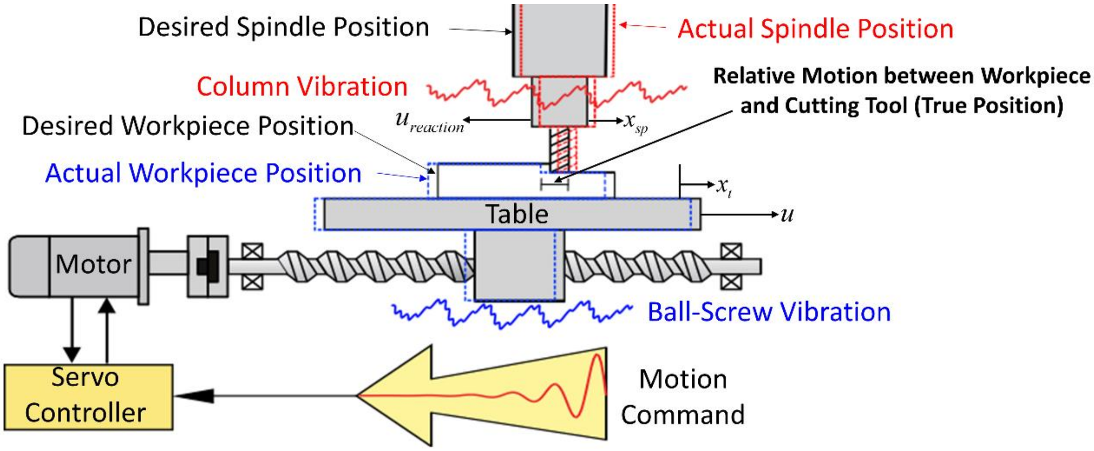

Figure 21. System Representation of a Ball-Screw Driven Machine

As a motion command is sent to the motor, it will rotate the ball screw which will rotate the table. However, this motion creates a reaction force to the machine base which is connected to the spindle. Thus, the spindle experienced a reaction force on the opposite direction of the force applied to the table. This representation can be further simplified as a simple mass spring and damper system as shown in Figure 22.

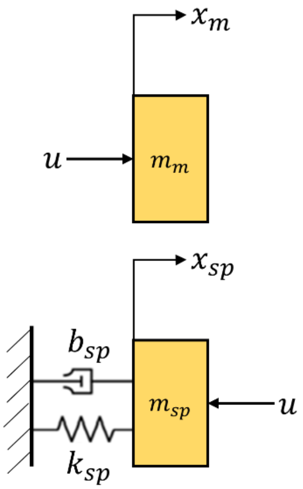

Figure 22. Simplified Mass Spring Damper Representation of the System

where m_m is the motor mass, x_m is the motor position, m_{sp}, b_{sp}, k_{sp} are the mass, damping and stiffness of the spindle, x_{sp} is the spindle position and u is the control signal.

Since the goal of this section is to minimize the relative vibration of the spindle and workpiece/table, two equations of motion with respect to the motor and spindle position are formulated as Eq(22). Note that motor and table position are assumed to be the same in this case for simplification of the algorithm and the fact that rotary encoder used for the experiment has a cleaner data than linear encoder.

$$\begin{aligned} m_m\ddot{x}_m &= u \\ m_{sp}\ddot{x}_{sp} &= -b_{sp}\dot{x}_{sp} - k_{sp}x_{sp} - u \\ m_{sp}\dddot{x}_{sp} &= -b_{sp}\ddot{x}_{sp} - k_{sp}\dot{x}_{sp} - \dot{u} \end{aligned} \tag{22}$$

where $\dot{x}_{sp}, \ddot{x}_{sp}$ and $\dddot{x}_{sp}$ are the velocity, acceleration and jerk of the spindle respectively, and $\dot{u}$ is the derivative of the control signal. It is important to ensure that double integration of accelerometer signal is avoided to reduce the effect of accelerometer drift. In this case, accelerometer is used to measure the spindle acceleration, which means the spindle position variable may not occur in the equations of motion [17]. Once the equations of motion are determined, state space representations for both the motor and spindle are constructed as shown in Eq(23) and Eq(24) respectively.

$$\begin{pmatrix} x_m \\ \dot{x}_m \\ \ddot{x}_m \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \int x_m \\ x_m \\ \dot{x}_m \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \dfrac{1}{m_m} \end{pmatrix} u \tag{23}$$

For spindle dynamics, consider the basic state space form:

$$\begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} = \begin{pmatrix} \alpha_1 & \alpha_2 \\ \alpha_3 & \alpha_4 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} + \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} u \tag{24}$$

Where in this case u is the input vector, which in this case is the control signal, q_1 is the output vector and q_2 is the state vector. Since spindle position variable is avoided, it is necessary to modify the state space formula to include $\dot{u}$ in addition of u, following the equation of motion shown in Eq(22). Generally, q_1 is assigned as the position x_{sp}. However, in this modified state space, q_1 is assigned as the spindle velocity $\dot{x}_{sp}$. The mathematical procedure is shown below in Eq(27).

$$\begin{aligned}\dot{q}_1 &= \alpha_1 q_1 + \alpha_2 q_2 + \beta_1 u\\ \dot{q}_2 &= \alpha_3 q_1 + \alpha_4 q_2 + \beta_2 u\\ q_1 &= \dot{x}_{sp}\end{aligned} \tag{25}$$

Then, $\dot{q}_2$ equation of the state space must follow the desired equation of motion in Eq(22). Eq(26) shows the calculation to obtain $\dot{q}_2$.

$$\begin{aligned}\dot{q}_2 = \alpha_3 q_1 + \alpha_4 q_2 + \beta_2 u \Rightarrow \dddot{x}_{sp} &= -\frac{b_{sp}}{m_{sp}}\ddot{x}_{sp} - \frac{k_{sp}}{m_{sp}}\dot{x}_{sp} - \frac{1}{m_{sp}}\dot{u}\\ \dot{q}_2 &= \dddot{x}_{sp} + \frac{b_{sp}}{m_{sp}}\ddot{x}_{sp} + \frac{1}{m_{sp}}\dot{u}\\ q_2 &= \ddot{x}_{sp} + \frac{b_{sp}}{m_{sp}}\dot{x}_{sp} + \frac{1}{m_{sp}}u\end{aligned} \tag{26}$$

Lastly, Eq(25)-(26) are substituted into Eq(24), and the overall state space representation is obtained as shown in Eq(27).

$$\begin{pmatrix} \ddot{x}_{sp} \\ \dddot{x}_{sp} + \frac{b_{sp}}{m_{sp}}\ddot{x}_{sp} + \frac{1}{m_{sp}}\dot{u} \end{pmatrix} = \begin{pmatrix} -\frac{b_{sp}}{m_{sp}} & 1 \\ -\frac{k_{sp}}{m_{sp}} & 0 \end{pmatrix} \begin{pmatrix} \dot{x}_{sp} \\ \ddot{x}_{sp} + \frac{b_{sp}}{m_{sp}}\dot{x}_{sp} + \frac{1}{m_{sp}}u \end{pmatrix} + \begin{pmatrix} -\frac{1}{m_{sp}} \\ 0 \end{pmatrix} u \tag{27}$$

Then, the combined state space representation consisting of motor and spindle dynamics is shown in Eq(28).

$$\underbrace{\begin{pmatrix} x_m \\ \dot{x}_m \\ \ddot{x}_m \\ \ddot{x}_{sp} \\ \dddot{x}_{sp} + \frac{b_{sp}}{m_{sp}}\ddot{x}_{sp} + \frac{1}{m_{sp}}\dot{u} \end{pmatrix}}_{\dot{X}} = \underbrace{\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{b_{sp}}{m_{sp}} & 1 \\ 0 & 0 & 0 & -\frac{k_{sp}}{m_{sp}} & 0 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} \int x_m \\ x_m \\ \dot{x}_m \\ \dot{x}_{sp} \\ \ddot{x}_{sp} + \frac{b_{sp}}{m_{sp}}\dot{x}_{sp} + \frac{1}{m_{sp}}u \end{pmatrix}}_{X} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ \frac{1}{m_m} \\ -\frac{1}{m_{sp}} \\ 0 \end{pmatrix}}_{B} u$$

(28)

One way to reduce the relative motion between the spindle and the motor/table is to reduce their overall error altogether. Since eq (28) is a state space representation that contains both the motor and spindle, the error formulation can be constructed as follows.

$$u = ke$$
$$k = \begin{pmatrix} k_1 & k_2 & k_3 & k_4 & k_5 \end{pmatrix}$$
$$e = X_r - X = \begin{pmatrix} \int x_{m,r} - \int x_m \\ x_{m,r} - x_m \\ \dot{x}_{m,r-}\dot{x}_m \\ \dot{x}_{sp,r} - \dot{x}_{sp} \\ \ddot{x}_{sp,r} - (\ddot{x}_{sp} + \frac{b_{sp}}{m_{sp}}\dot{x}_{sp} + \frac{1}{m_{sp}}u) \end{pmatrix} \tag{29}$$

Motor and spindle error can be extracted from Eq(29) by using the equation shown below:

$$e_m = x_{m,r} - x_m, e_{sp} = x_{sp,r} - x_{sp} \tag{30}$$

0

Then, by using Eq(29)-(30), the equation that governs the control signal can be obtained as shown below.

$$u = k_1 \int (e_m) + k_2 e_m + k_3 \dot{e}_m + k_4 \dot{e}_{sp} + k_5 \ddot{e}_{sp} + k_5 \left(-\frac{b_{sp}}{m_{sp}} \dot{x}_{sp} - \frac{1}{m_{sp}} u\right) \tag{31}$$

To implement Eq(31) in a control system form, Laplace transformed is performed, which results in the overall transfer function shown in Eq(32).

$$u(s) = \underbrace{\left(\frac{m_{sp}}{m_{sp}+k_5}\right)}_{k_u} \left[\underbrace{\left(\frac{k_1}{s} + k_2 + k_3 s\right)}_{\text{Motor PID Constant}} e_m(s) + \underbrace{\left(k_4 s + k_5 s^2\right) e_{sp}(s) - k_5 \frac{b_{sp} s}{m_{sp}} x_{sp}(s)}_{\text{Acceleration Feedback Controller}} \right] x_m \tag{32}$$

Since the overall transfer function is obtained, a new controller block diagram is constructed to follow the state space formulation as shown in Figure 23.

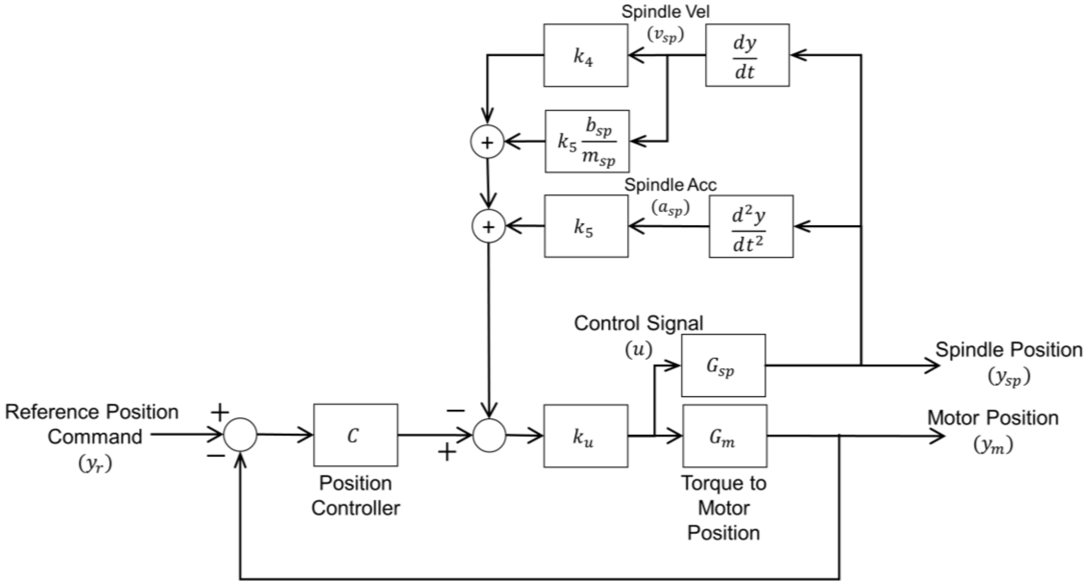

Figure 23. State Space Acceleration Feedback Controller

To obtain the parameters of the acceleration feedback controller, the pole of the overall controller must be determined. Then, by formulating the overall controller in a state space equation in Eq(33), MATLAB's pole placement function is utilized to obtain the best parameters to achieve the desired poles [18].

$$\dot{X} = \left(A - Bk\right)X + BkX_r \tag{33}$$

where $eig\left(A - Bk\right)$ is the desired pole locations. There are 5 poles to be placed; 3 motor feedback controller poles which in this case are the PPI controller poles, and 2 desired poles for the spindle dynamics. Therefore, it is possible to directly control the damping of the spindle dynamics. Consider a simple second order transfer function $G_{sp} = \dfrac{1}{s^2 + 2\zeta\omega_{n,sp}s + \omega_{n,sp}^2}$ representing the spindle dynamics where ζ is the damping coefficient and $\omega_{n,sp}$ is the natural frequency of the spindle. This transfer function contains poles that is dependent on ζ and $\omega_{n,sp}$. Hence, by selecting the desired ζ and ω_n then obtaining the poles, which is the root of the second order polynomial, the state space controller will be able to send a compensated signal to force the spindle to behave as desired. As for the PPI poles, since the PPI controller is originally tuned separately without the spindle dynamics as will be shown in section 4.4.1, it is possible to obtain the pole locations of the PPI controller.

This controller is able to attenuate the spindle resonance by adding vibration compensator signals to the motor. Therefore, the motor will move more, resulting in relative vibration between the motor and spindle remains the same if not worse. Thus, feedforward *H* is added to the system to ensure that the overall true tracking performance is the same as the motor tracking of original PPI controller without acceleration feedback. This is because the motor tracking of the original PPI controller is smooth since it is not affected by the spindle dynamics. Thus, if the true tracking of the system with acceleration feedback can be as smooth as the motor tracking without acceleration feedback, the vibration is completely compensated. True tracking is how well the combination of motor and spindle motions follow the reference trajectory. True error is the error between the reference position and combination of motor and spindle position. Thus, it can be defined as $e^{true} = y_r - (y_m + y_{sp})$. The updated control system is shown Figure 24.

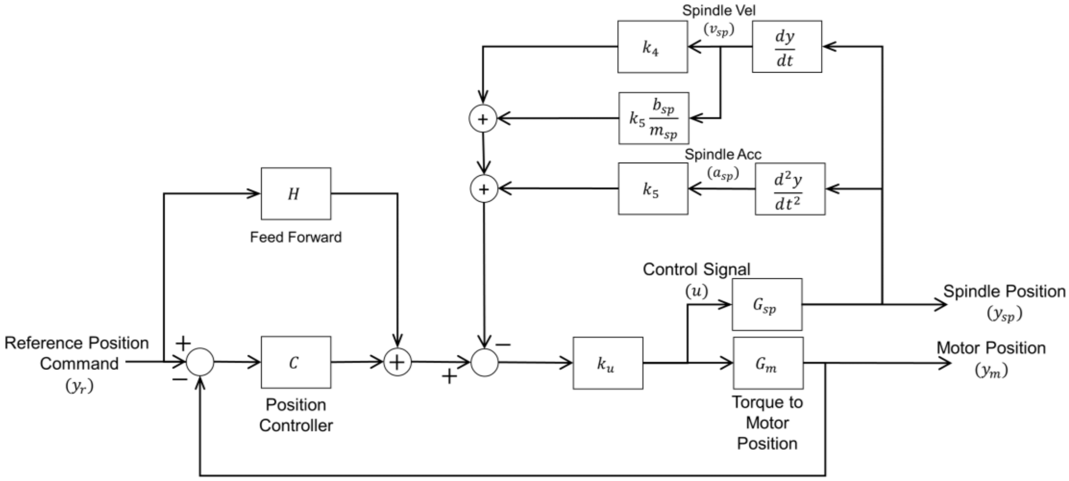

Figure 24. State Space Acceleration Feedback Controller with Feed Forward

The purpose of feedforward *H* is to make the true error of the state space controller to be the same as the motor error of a PPI controller. Thus, the objective can be written mathematically as $e_{SS}^{true} = e_{PPI}^{motor}$ where e_{SS}^{true} and e_{PPI}^{motor} are true error of state space controller and motor tracking error of PPI controller respectively. To obtain the proper transfer function for *H* that will satisfy that aforementioned objective, closed loop transfer function for both state space controller and PPI controller has to be obtained. Eq(34) shows the closed loop true tracking transfer function for state space controller.

$$G_{SS}^{true} = \frac{y_{t,SS}}{y_r} = \frac{K_u C(G_m + G_s) + K_u H(G_m + G_s)}{1 + NLT_{ss}} \tag{34}$$

Then, the closed loop motor tracking transfer function for a PPI controller is also constructed as follows:

$$G_{PPI}^{motor} = \frac{y_{m,PPI}}{y_r} = \frac{C_v G_m (K_p + s)}{1 + C_v G_m (K_p + s)} \tag{35}$$

Where C_v is the velocity controller and K_p is the proportional gain for position controller of a regular PPI controller. These closed loop transfer function shown in Eq(34)-(35) are derived from the block diagram shown in Figure 24 for state space controller and Figure 19 and Figure 20 for PPI controller.

Then, since the true error of the state space controller has to be equal to the motor tracking error of the common PPI controller, the error transfer function of both state space and PPI controller are constructed as follows:

$$\begin{aligned} e_{SS}^{true} &= 1 - G_{SS}^{true} \\ e_{PPI}^{motor} &= 1 - G_{PPI}^{motor} \end{aligned} \tag{36}$$

Finally, the transfer function for *H* can be obtained in Eq(37).

$$\begin{gathered} e_{SS+FF}^{true} = e_{PPI}^{motor} \\ H = \left(\frac{G_{PPI}^{track}\left(1 + NLT_{ss}\right) - K_u C(G_m + G_{sp})}{K_u (G_m + G_{sp})} \right) \end{gathered} \tag{37}$$

4.4. Simulation and Experiment Results

4.4.1. PPI Controller Tuning

This section focuses on the tuning procedure of a PPI controller without acceleration feedback which will serve as a base controller to be modified. The controller block diagram is shown in Figure 25.

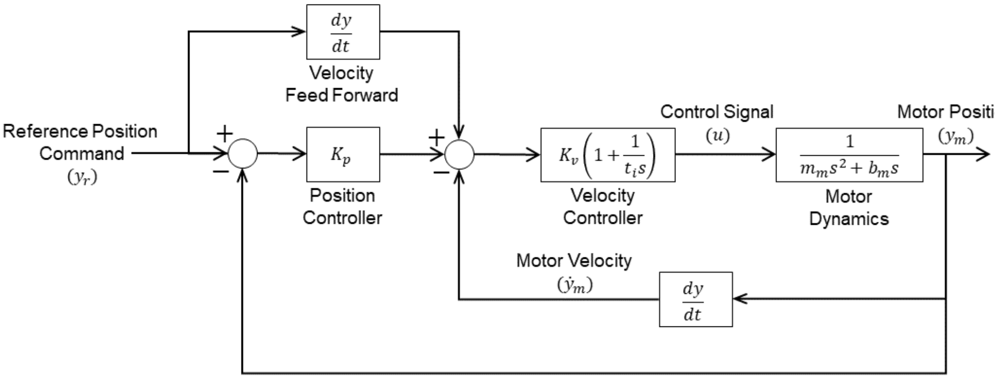

Figure 25. PPI Controller without Spindle Feedback

There are 3 parameters to tune, K_p, K_v and t_i. From the block diagram above, a closed loop transfer function is constructed as:

$$\frac{y_m}{y_r} = \frac{\left(K_p + s\right)K_v\left(t_i s + 1\right)}{m_m t_i s^3 + b_m t_i s^2 + \left(K_p + s\right)K_v\left(t_i s + 1\right)} \tag{38}$$

The equivalent controller transfer function can be written as:

$$C(s) = \left(K_p + s\right)K_v\left(t_i s + 1\right) = K_p K_v K_i \frac{\left(t_i s + 1\right)\left(t_p s + 1\right)}{s}$$
$$K_i = \frac{1}{t_i} \tag{39}$$
$$t_p = \frac{1}{K_p}$$

There are 3 frequency domain parameters that has to be determined to tune the controller; cross-over frequency $\left(\omega_c\right)$, phase margin *(PM)* and integrator phase delay

(ϕ_i). The integrator phase delay is used to calculate the integral time constant t_i as shown in Eq.(40).

$$\begin{gathered}\phi_i = \angle \frac{t_i s+1}{s}\bigg|_{s=j\omega_c} \\ t_i = \frac{\tan\left(\phi_i+90\right)}{\omega_c} = \frac{1}{K_i}\end{gathered} \tag{40}$$

Next is to calculate the position gain parameter K_p. The position gain K_p is responsible for ensuring that the phase margin around the cross-over frequency is as close to the desired phase margin as possible. Phase margin by definition can be found by calculating the phase of the system's NLT when the gain of the NLT crosses 0dB, which in this case is at the cross-over frequency. Thus, consider a mathematical equation that calculate the phase of the NLT at the cross-over frequency:

$$\phi_p = -180 - \phi_i - \angle G_m + PM \tag{41}$$

Where ϕ_p is the phase of the NLT and $\angle G_m$ is the phase of the actual system at the cross-over frequency. In this case, since the transfer function for the system representation is a simple rigid body model, the phase is considered to be -180 degree. Then, from Eq(39), the only parameter left that contributes to the phase at the crossover frequency is $t_p s+1$. Thus, by combining it with Eq(41), the position gain K_p can be calculated as:

$$\begin{gathered}\phi_p = \angle\left(t_p s+1\right)\Big|_{s=j\omega_c} \\ t_p = \frac{\tan\left(\phi_p\right)}{\omega_c} = \frac{1}{K_p}\end{gathered} \tag{42}$$

Since the phase margin and integrator phase delay is taken care by K_p and t_i, the only desired parameters to be considered is the cross-over frequency ω_c. Also, the only parameters to be identified is the velocity proportional gain K_v. Thus, K_v must be calculated so that the NLT of the system has the desired cross-over frequency. This can be achieved by calculating the gain of the NLT at the cross-over frequency and set it to 0dB or 1 in absolute scale. The mathematical equation is shown in Eq(43).

$$\begin{gathered}\left|\frac{K_v(s+K_i)(s+K_p)}{s}G_m(s)\right|_{s=j\omega_c}=1\\K_v=\frac{1}{\left|\frac{(j\omega_c+K_i)(j\omega_c+K_p)}{j\omega_c}\right|\left|G_m(j\omega_c)\right|}\end{gathered}\tag{43}$$

The desired parameters used for this controller tuning is as follows:

$$\begin{aligned}\omega_c&=60[Hz]\\PM&=80\deg\\\phi_i&=-10\deg\end{aligned}\tag{44}$$

Figure 26 shows the frequency domain performance of the explained PPI tuning method only on the Y-axis. X-axis is tuned using the same method.

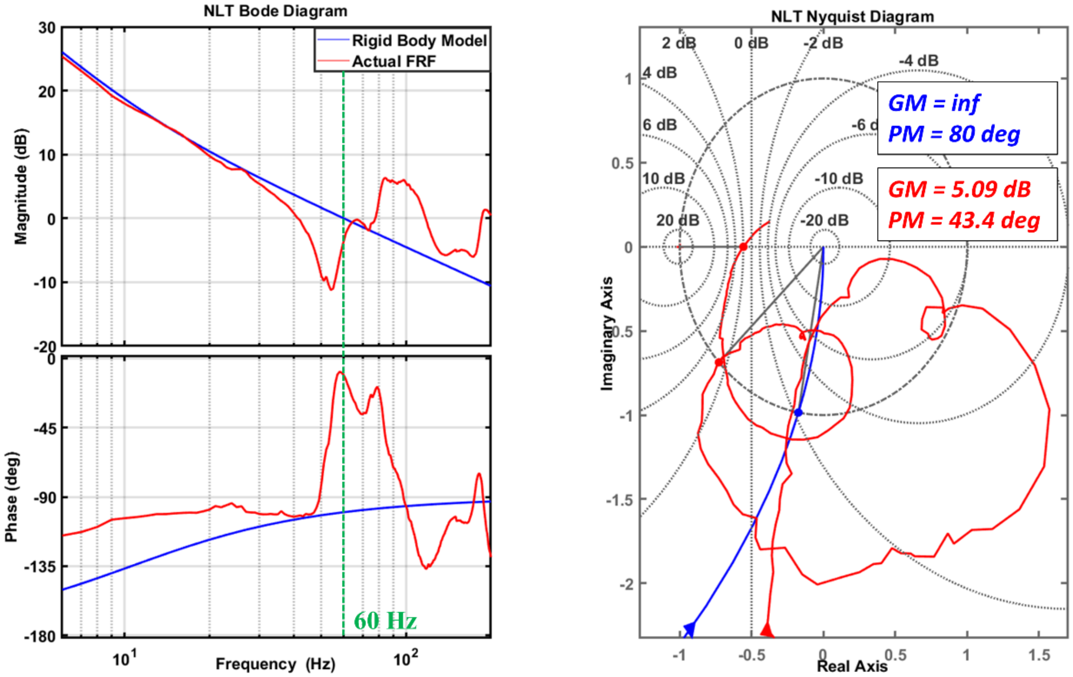

Figure 26. Frequency Domain Performance of PPI Tuning

There are significant differences between the frequency domain performance of modelled system compared to that of the actual system. When the controller is implemented in the actual system, it has significantly reduced phase and gain margin. This is caused by the modelling inaccuracy shown in Figure 16. Since K_p is the parameter that dictates the phase margin, it can be manually adjusted to ensure that the phase margin of the actual system is still stable.

4.4.2 Loop Shaping Acceleration Feedback Experiment Results

The motors of the machine are controlled by P-PI position controllers with velocity feed forward. Since the feedback controller is a simple proportional gain, it is possible to tuned them by trial and error, even though the result isn't optimal. Figure 27 shows the closed loop true tracking FRF with respect to K_a .

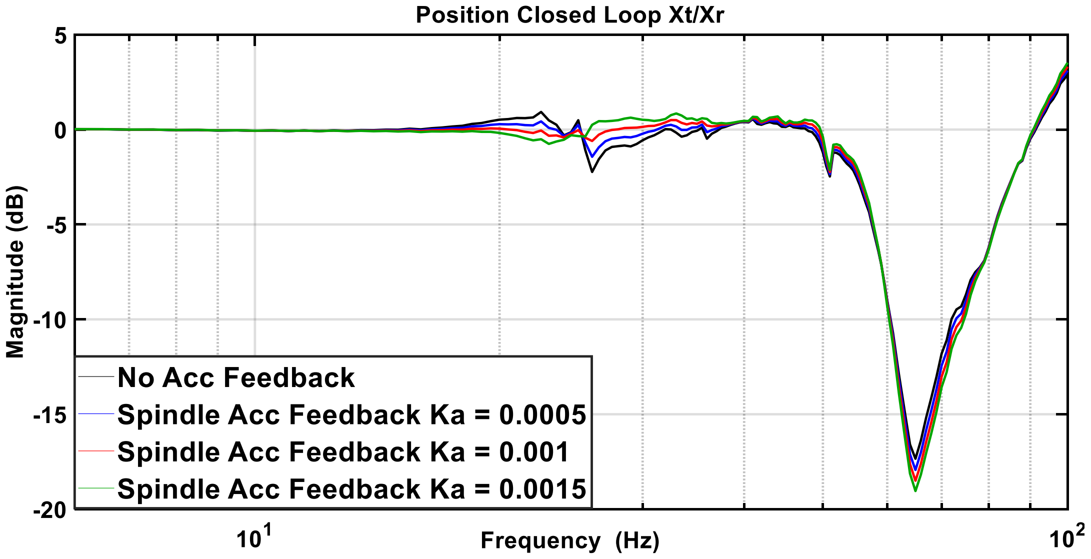

Figure 27. Closed Loop True Tracking FRF With Respect to Ka

As shown above, $K_a = 0.001$ yield the best closed loop true tracking performance as it reduces the amplitude of resonance and antiresonance around 25 Hz the most, which is approximately the resonance frequency of the spindle. Hence, the experiment is done with $K_a = 0.001$. Time domain experiment result is shown in Figure 28 and frequency domain experiment result is shown in Figure 29.

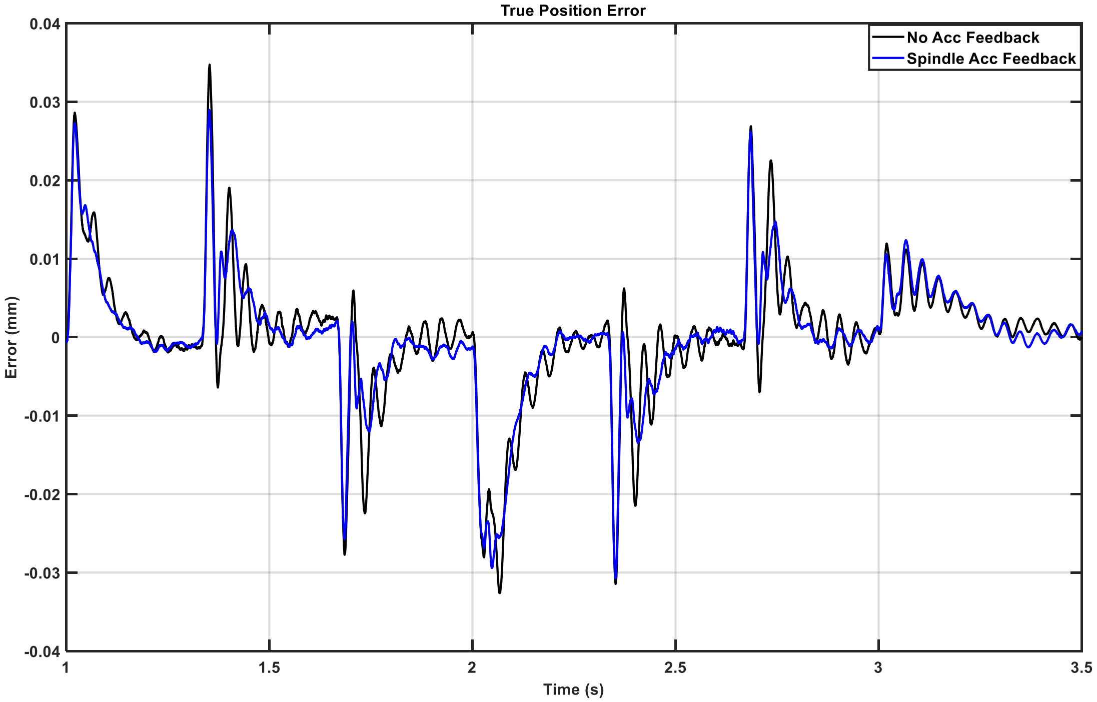

Figure 28. Acceleration Feedback Time Domain Experiment Result with Ka = 0.001

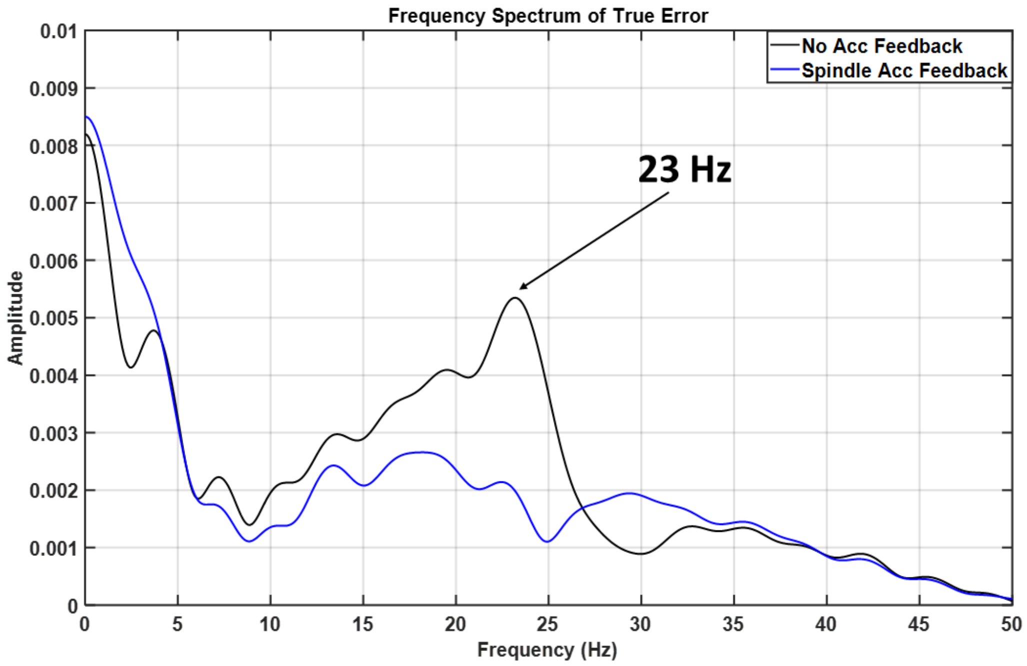

Figure 29. True Error Frequency Spectrum of Experiment with Ka=0.001

Figure 29 shows the true error frequency spectrum of the acceleration feedback experiment with $K_a = 0.001$ and it shows that the resonance at 23 Hz is greatly attenuated when the acceleration feedback is implemented. This proves that the acceleration feedback loop shaping method is able to reduce the true error of the machine, although the feedback parameters are suboptimal.

To circumvent the suboptimal acceleration feedback controller of the loop shaping method, state space method is developed to ensure it is possible to move the poles of spindle dynamics to any desired pole locations.

4.4.3. State Space Spindle Feedback Simulation and Experiment Results

The PPI crossover frequency for this section is reduced from 60 Hz on the previous section to 30 Hz. This is done to ensure that the system is more stable and robust than before. Thus, all PPI and state space approach simulations in this section used 30 Hz crossover frequency.

Then, as mentioned in section 4.3, the double integration of spindle acceleration needs to be avoided. Figure 30 shows the drift that occurred when accelerometer signals are integrated. Although the drift can be compensated to some extent by implementing high pass filters, a more aggressive filter is needed if the signals are integrated multiple times. A more aggressive filter has more gain and phase which will distort the data especially if the vibrations measured is at a low frequency (around 25 Hz). Due to this limitation, only 1 integration of the spindle acceleration signal is allowed and it is filtered by a simple first order high pass filter $H_{HPF}(s) = \frac{s}{s + \omega_{HPF}}$ where ω_{HPF} is the cut off frequency for the high pass filter and is set to be 2 Hz.

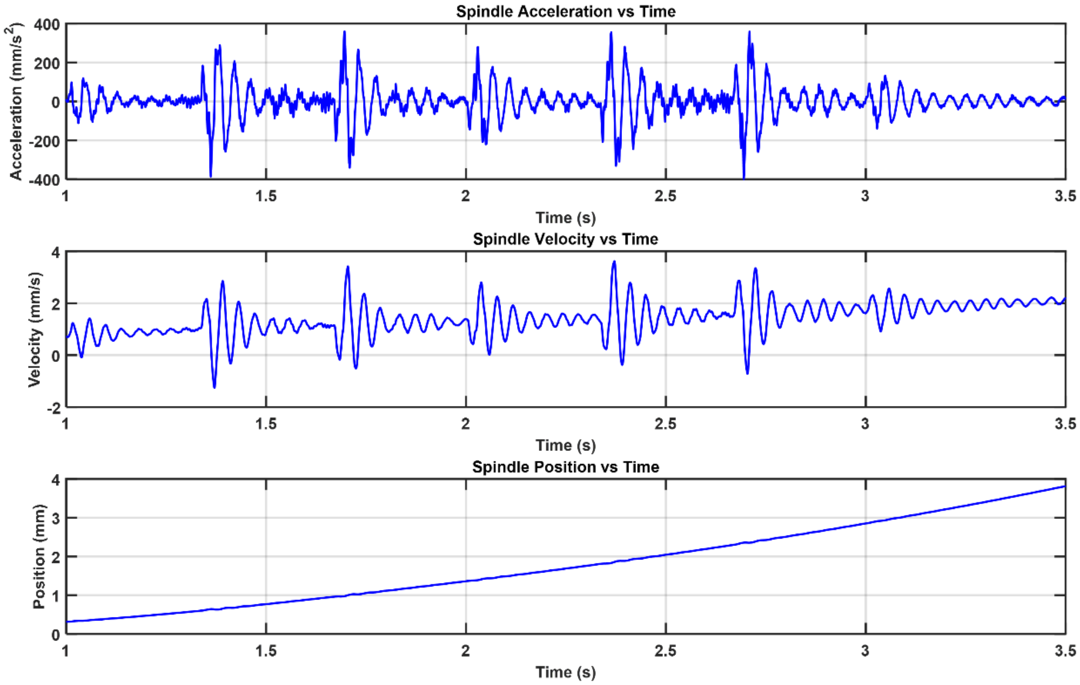

Figure 30. Accelerometer Drift Occurs when Integrating Accelerometer Signal (Experiment)

The algorithm is first tested on simulation and its performance and stability is confirmed by looking at the FRF of the overall system. Figure 31 shows the difference in motor, spindle and true tracking performance between PPI controller without acceleration feedback and state space controller with spindle feedback. As can be seen from those graphs, there is an attenuation at 25 Hz on both motor and spindle dynamics, which results in overall attenuation on the true tracking performance at 25 Hz compared to a regular PPI controller which has resonance that frequency.

Then, a simulation model is made in MATLAB Simulink to simulate the time domain response of the controller. Figure 32 shows the reference trajectory used for the simulation and Figure 33 shows the time domain simulation results for the motor position error and spindle position.

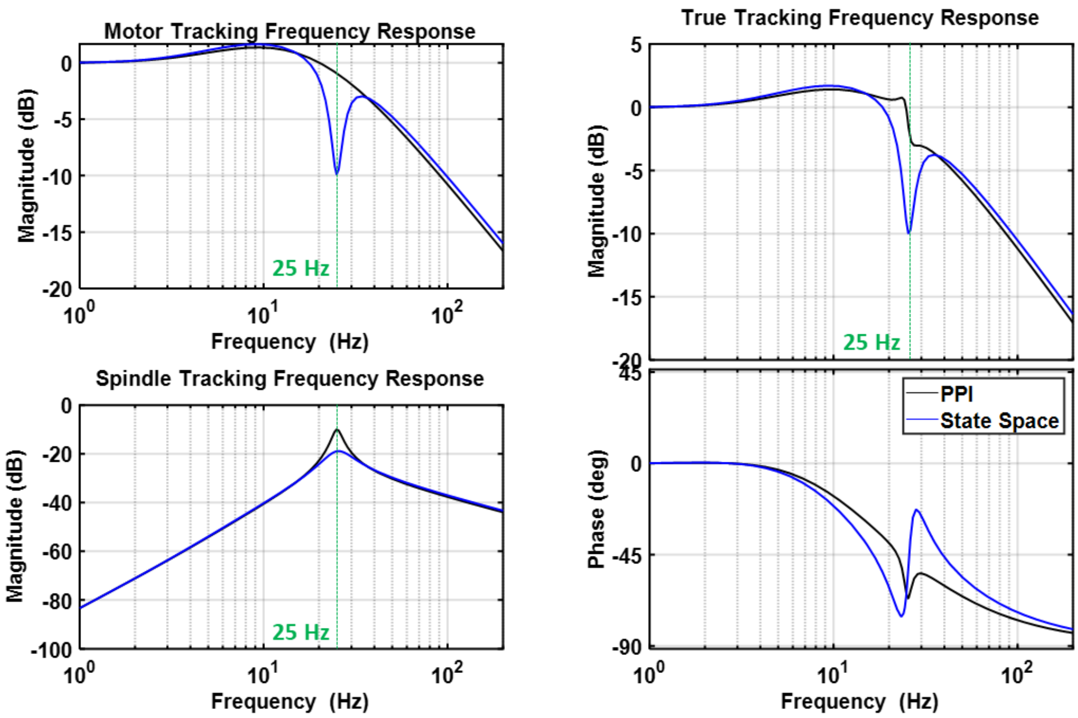

Figure 31. Frequency Response Comparison of PPI Controller without Acc Feedback and State Space Controller (Simulation)

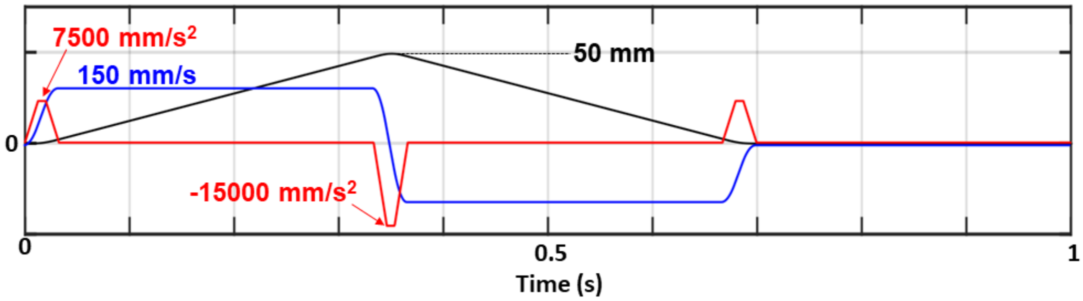

Figure 32. Reference Trajectory

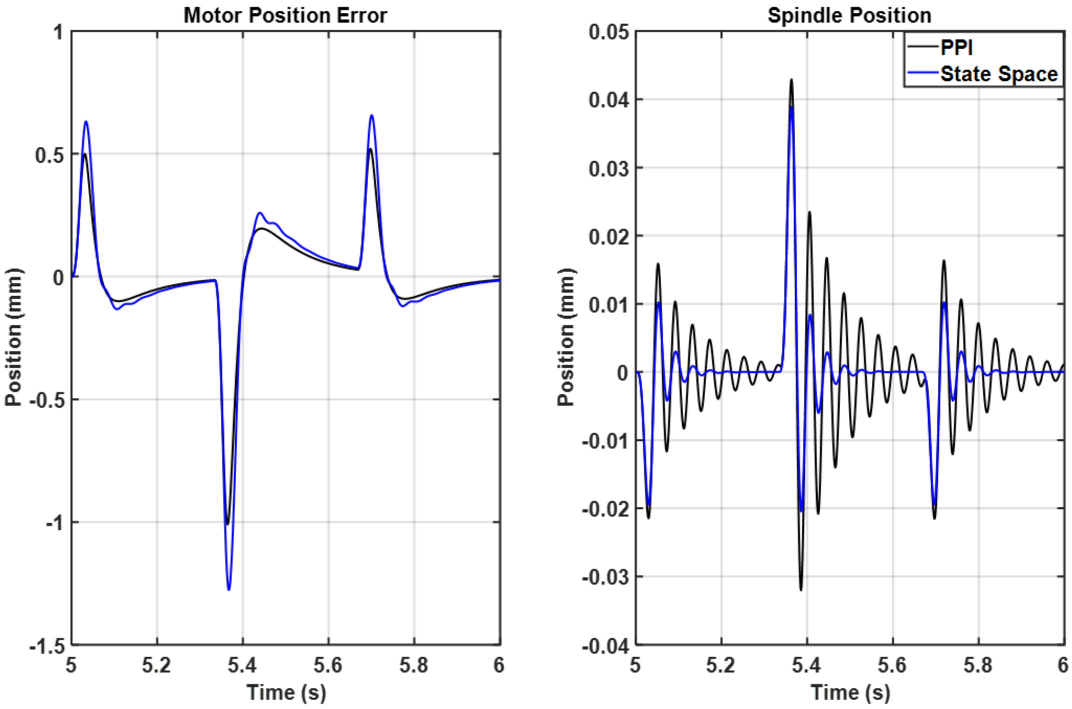

Figure 33. Motor Position Error and Spindle Position Time Domain Simulation Results (Simulation)

As can be seen on Figure 33, the spindle vibrations are significantly damped. However, the compensation signal introduces a vibration to the motor motion as can be seen by the motor error of state space controller has more vibration than that of the PPI. The true error data of this simulation is shown in the figure below.

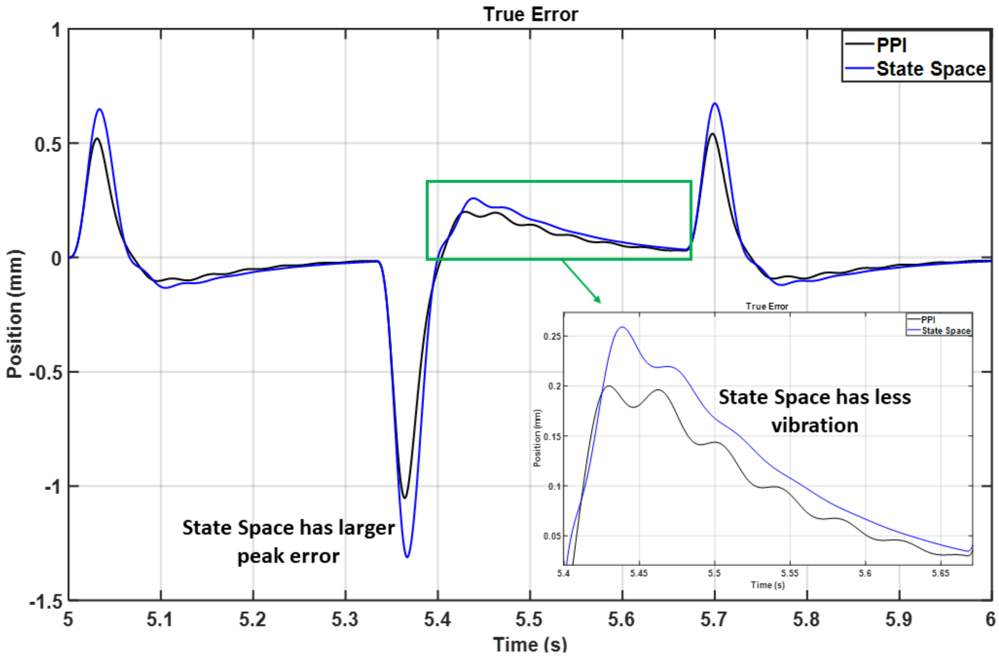

Figure 34. True Error Comparison Between State Space and PPI (Simulation)

Comparison between true error of state space controller and that of the PPI controller can be seen in Figure 34. State space controller has less vibration in its true error. However, from Figure 31, it shows that the low frequency tracking is not as good as that of the PPI controller. Hence, state space controller has a larger peak error then the PPI controller.

Before the experiment is conducted, gain and phase margin are analyzed to ensure the system's stability. However, due to the high frequency model inaccuracy shown in Figure 16, the system is unstable as it has a negative gain margin as shown in Figure 35.

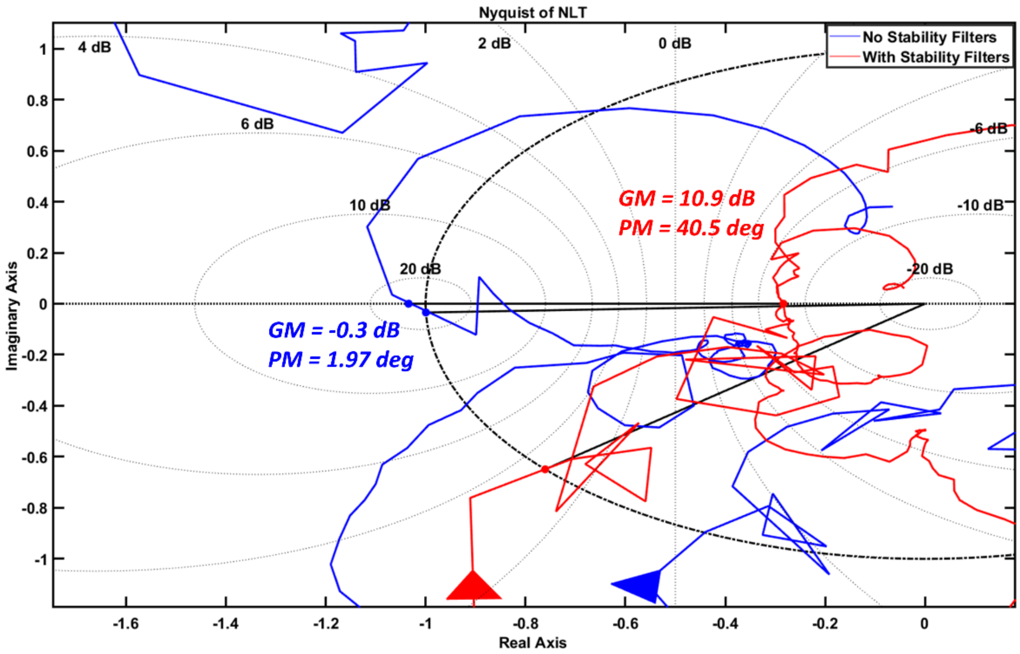

Figure 35. Stability Margins Comparison between State Space Spindle Feedback Controller with and without Stability Filters

Therefore, additional filters are introduced to the controller to fix the stability of the controller as shown in Figure 36.

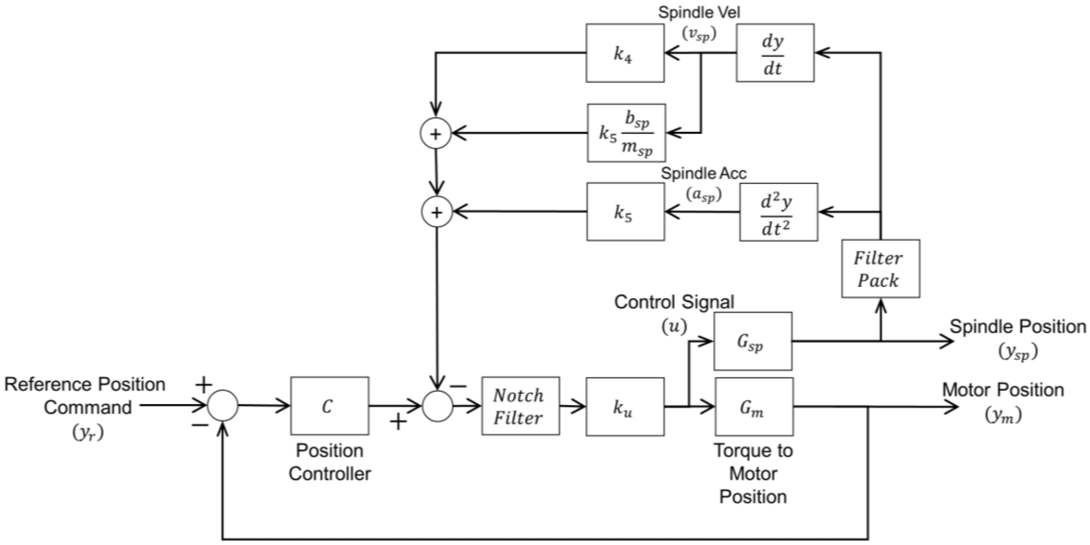

Figure 36. State Space Spindle Feedback Controller with Stability Filters

The transfer function structure of the additional stability filters are shown in Eq(45).

$$NotchFilter = \frac{s^2 + 2\zeta_{N,r}\omega_N s + \omega_N{}^2}{s^2 + 2\zeta_{N,n}\omega_N s + \omega_N{}^2}$$

$$FilterPack = \left(\frac{1}{\dfrac{s}{\omega_{LPF}} + 1}\right)\left(\frac{s}{s + \omega_{HPF}}\right)\left(\frac{\omega_{notch}{}^2}{\omega_{res}{}^2}\,\frac{s^2 + 2\zeta_r\omega_{res}s + \omega_{res}{}^2}{s^2 + 2\zeta_n\omega_{notch}s + \omega_{notch}{}^2}\right) \tag{45}$$

Notch filter is used to attenuate high frequency resonance of the motor while the filter pack is used to eliminate unmodelled spindle dynamics from the feedback loop. Unmodelled spindle dynamics is shown in Figure 37.

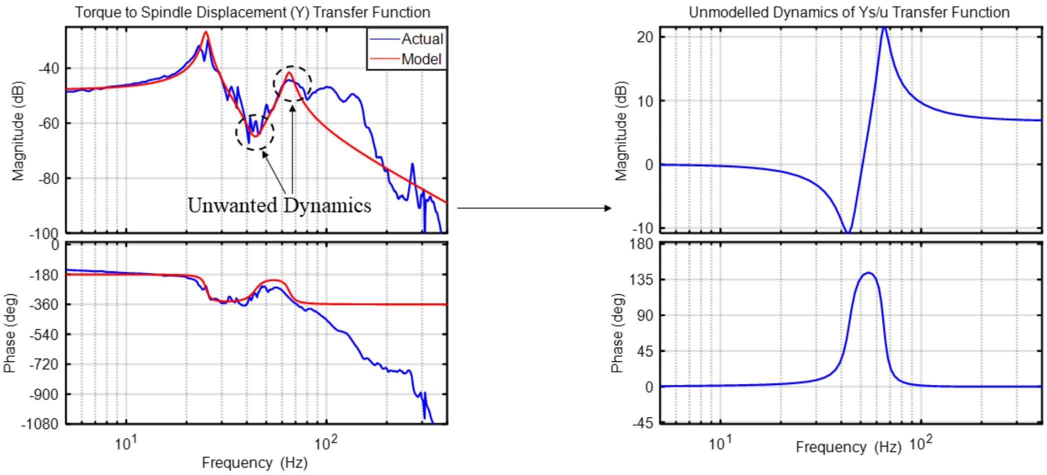

Figure 37. Unmodelled Spindle Dynamics for Filter Pack

State space feedback tuning method without feed forward was experimentally tested on a CNC milling machine. Figure 38 and Figure 39 shows the experimental results of state space spindle feedback method.

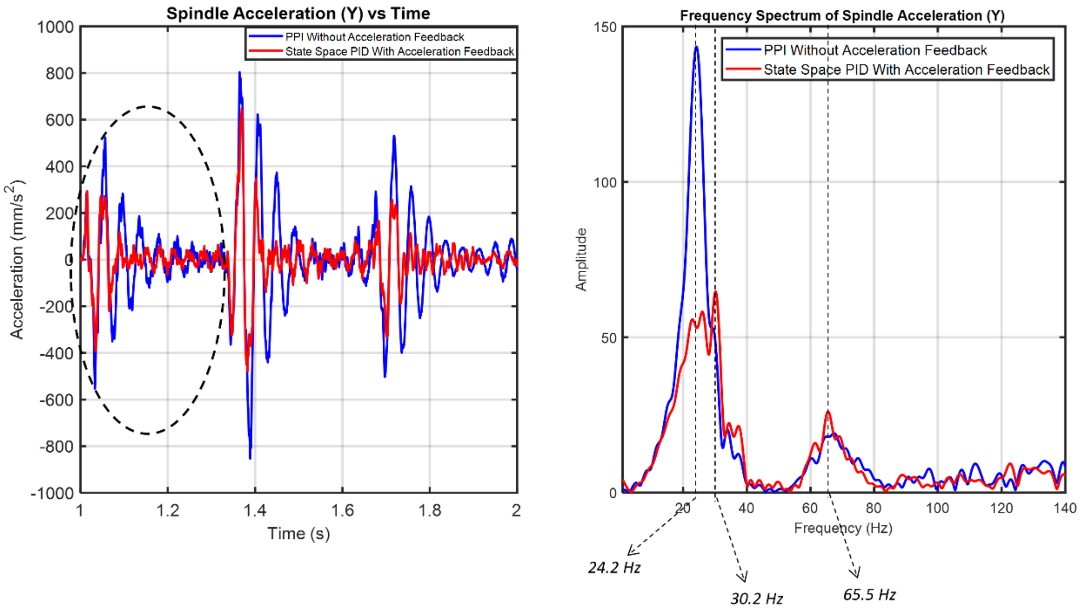

Figure 38. Spindle Acceleration Comparison Between PPI and State Space Spindle Feedback (Experiment)

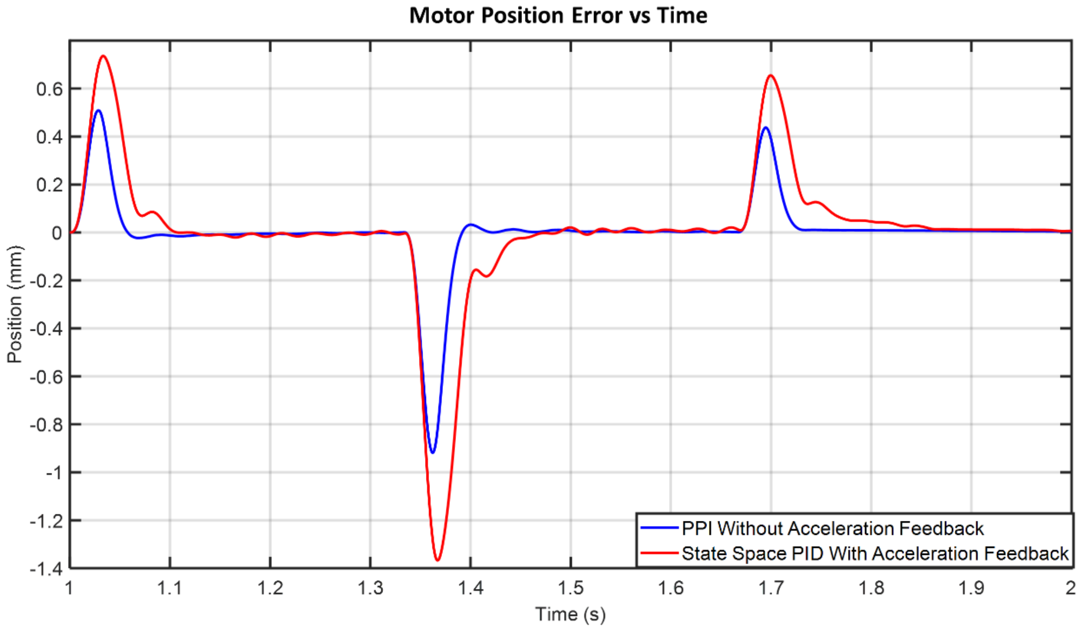

Figure 39. Motor Position Error Comparison Between PPI and State Space Spindle Feedback (Experiment)

Similar to simulation results, the vibration amplitude of the spindle is significantly reduced. On the other hand, the motor position error has a larger error to compensate for spindle vibration.

To compensate for the deterioration of low frequency tracking performance, the feedforward *H* is added. The goal of the feedforward is to make the true tracking performance of the state space controller similar to the motor tracking performance of the PPI controller. Figure 40 shows the motor error and spindle position simulation comparison

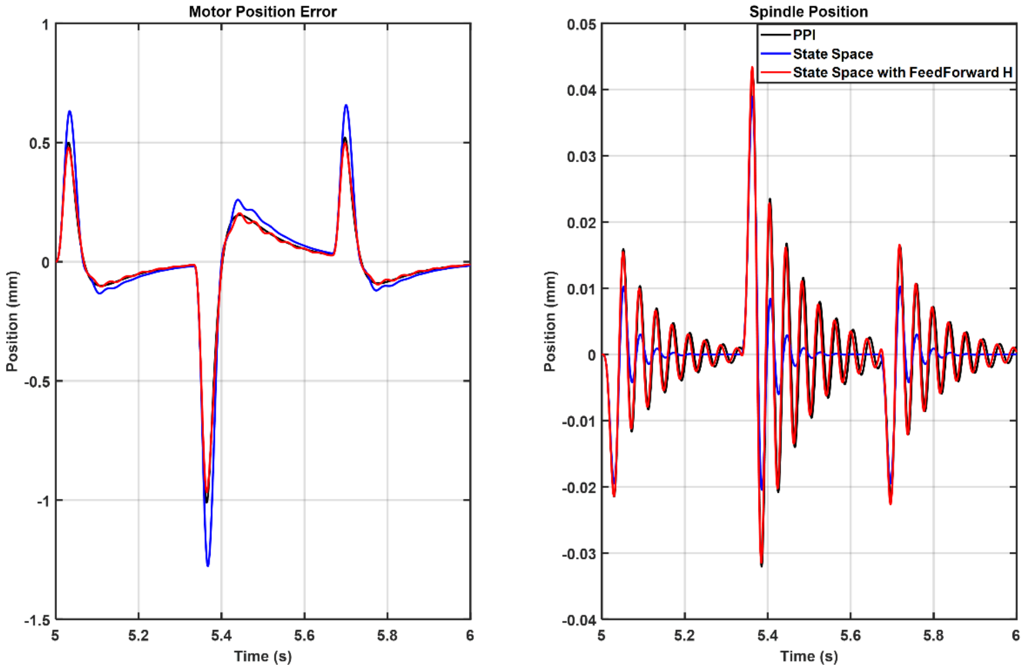

Figure 40. Motor Error and Spindle Position Simulation Comparison (Simulation)

As can be seen from the figure above, the feedforward helps to move the motor to compensate the spindle movement instead of trying to dampen the spindle vibration. Figure 41 shows the true error comparison between PPI and state space approach with and without the feedforward H. The addition of feedforward completely eliminates the true error vibration. Thus, the trajectory pre-compensation method on the next section is developed with the idea that a feedforward or pre-filter is adequate to improve the tracking performance at the cost of the robustness granted by the additional feedback.

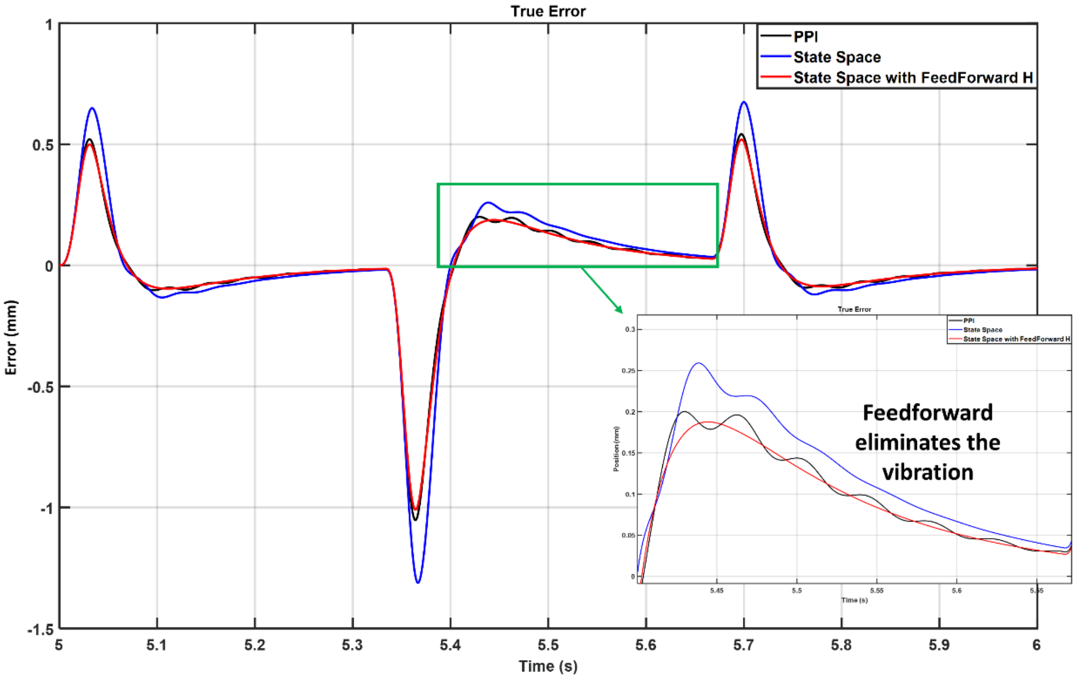

Figure 41. True Error Comparison Between State Space with and without feedforward H and PPI (Simulation)

Then the state space method with feed forward performance is compared to loop shaping method explain on Section 4.2. The simulation result is shown on Figure 42.

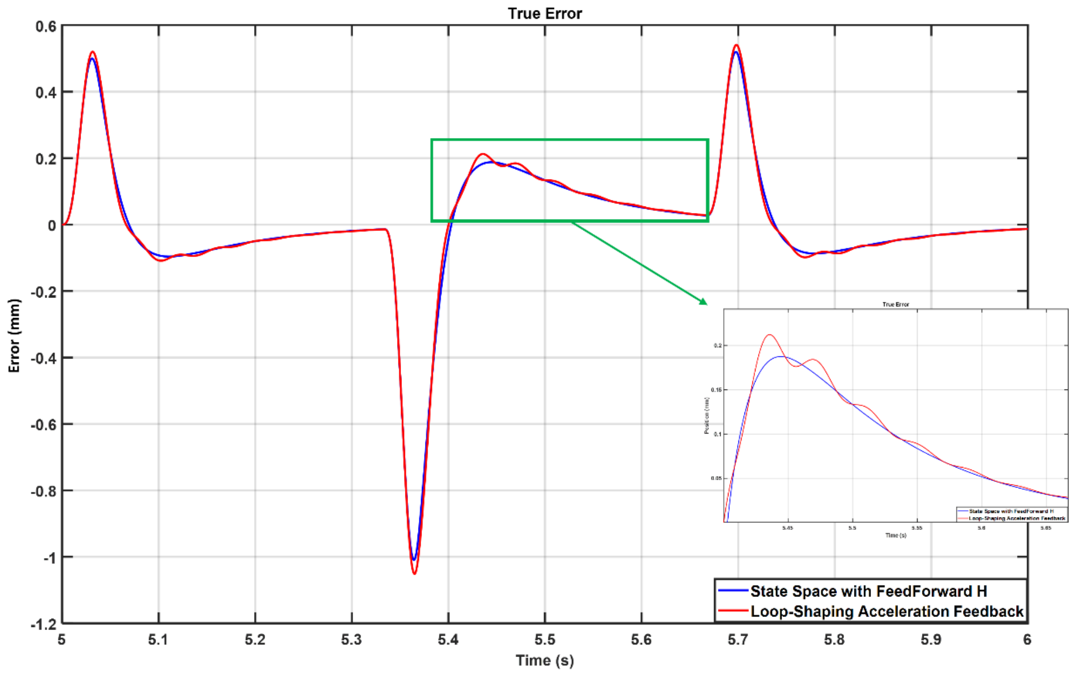

Figure 42. True Error Comparison Between State Space with Feedforward H and Loop-Shaping Acceleration Feedback

In conclusion, state space with feedforward results in the best true error vibration mitigation compared to state space without feed forward, loop shaping acceleration feedback and a normal PPI controller. State space performance also has the ability to move the spindle poles to a desired location analytically, which means spindle vibration mitigation of state space controller is better than loop-shaping acceleration feedback method. Therefore, the proposed state space approach is better at both true error and spindle vibration mitigation compared to a standard industrial PPI controller and loop-shaping acceleration feedback method.

5. Trajectory Pre-Compensation Based Vibration Mitigation

5.1. Introduction

Current industrial machine tool servo loops rely heavily on acceleration and jerk feedforward (FF) to widen their tracking bandwidth. However, for ball screw drive systems exhibiting lightly damped structural resonances, acceleration FF may induce unwanted vibrations and jerk FF are only most effective if resonance are rather low frequency. There are other attempts on feedforward tuning such as input shapers (IS) and Finite Impulse Response (FIR) based trajectory generation [19],[20]. However, they introduce a delay to motion, which elongates overall motion duration and deteriorates productivity. Other model based approaches exists in literature which are able to increase the tracking bandwidth while avoiding unwanted vibrations [21]. One of the most well known technique zero phase error tracking controller (ZPETC), which has been widely used in high speed precision positioning system such as in hard disk drives, metrology and lithography equipment [22]. However, this technique requires accurate modelling of the system.

This section presents a novel IIR trajectory prefilter design to widen closed loop tracking bandwidth of machine tool feed drives while avoiding unwanted vibration at the same time but doesn’t require accurate system modelling. In addition, the proposed method also includes the IIR trajectory prefilter design to compensate for motor to table ball screw transmission dynamics.

5.2. Motor Tracking Dynamics Compensator ($\boldsymbol{F_{MTDC}}$) Design

Typical ball-screw feed drive systems are controller based on motor position feedback. This section presents design of a motor tracking dynamics compensator to realize perfect motor side tracking. Figure 43 shows the proposed control system schematic.

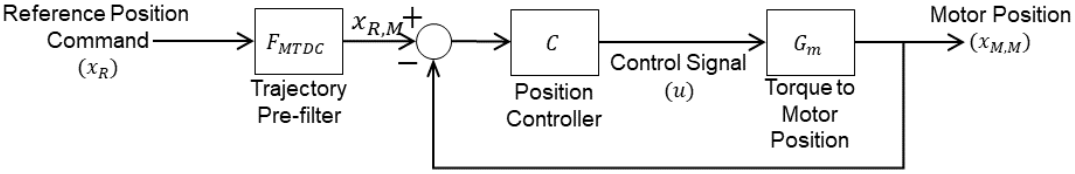

*Figure 43***.** *Proposed Control Scheme for Motor Tracking Dynamics Compensator*

One way to alleviate the model dependency of the compensation method is to introduce iterative learning control (ILC) to determine the parameters of F_{MTDC} through machine in the loop iteration. Note that the ideal transfer function for F_{MTDC} is the inverse of the closed loop tracking dynamics as shown in Eq (46).

$$\frac{y_{M,C}}{y_r} = F_{MTDC}\underbrace{\frac{G_m C}{1+G_m C}}_{\frac{y_M}{y_r}} \cong 1 \rightarrow F_{MTDC} = \left(\frac{y_M}{y_r}\right)^{-1} \qquad (46)$$

where y_M is the motor position, $y_{M,C}$ is the motor position once F_{MTDC} is applied and y_r is the reference position. Ideally, the form and order of F_{MTDC} requires a detailed knowledge of the feedback controller C as well as the motor dynamics G_m. Acquiring such information in a practical setting requires rigorous identification procedure. In order to circumvent these limitations, the form of F_{MTDC} is pre-determined based on a set of $N = 3$ 2nd order basis functions as:

$$F_{MTDC} = \sum_{k=1}^{N=3} \underbrace{\frac{b_{0,k}s^2 + b_{1,k}s + b_{2,k}}{a_{0,k}s^2 + a_{1,k}s + a_{2,k}}}_{\phi_k} \qquad (47)$$

where ϕ_k is the so-called 2nd order basis function. The denominator parameters; $\mathbf{a}_{\mathrm{k}} = [a_{2,k}\ a_{1,k}\ a_{0,k}]^T$ are assigned based on a selected damping ratio ζ_k and resonance frequency $\omega_{n,k}$ of each basis function is pre-assigned via prior knowledge of the system; for example, by simply observing the dominant motor side closed-loop dynamics (resonances) through time-domain tracking experiments.

The numerator coefficients $\mathbf{b}_{\mathrm{k}} = [b_{2,k}\ b_{1,k}\ b_{0,k}]^T$ are identified through closed-loop tracking experiments and making use of the ILC framework as follows. A motion trajectory is commanded to the servo system, motor side tracking errors (e_M) are recorded and numerator coefficients $\mathbf{b}_{\mathrm{k}}$ is determined to minimize the following cost function iteratively:

$$\min_{\mathbf{b}=[\mathbf{b}_1,\mathbf{b}_2,\mathbf{b}_3]^T} \left(J_{MTDC} = \frac{1}{2}\mathbf{e}_M^T \mathbf{e}_M \right) \tag{48}$$

Identical reference trajectory is commanded to the system several times and the parameters are updated in each iteration. To ensure monotonic convergence, gradient ∇J_{MTDC} and hessian $\nabla^2 J_{MTDC}$ of the objective function are generated for $N=3$ basis function case from:

$$\left.\begin{aligned}
\mathbf{e}_M &= x_R - F_{MTDC}CG_M\big/\left(1+CG_M\right)x_R \\
&= x_R - F_{MTDC}x_M \\
\nabla\mathbf{e}_M &= \frac{\partial\mathbf{e}}{\partial\mathbf{b}} = \begin{bmatrix} \frac{\partial\mathbf{e}}{\partial\mathbf{b}_0} & \frac{\partial\mathbf{e}}{\partial\mathbf{b}_1} & \frac{\partial\mathbf{e}}{\partial\mathbf{b}_2} \end{bmatrix}^T = -\nabla F_{MTDC}x_M \\
\nabla F_{MTDC} &= \begin{bmatrix} \dfrac{\begin{bmatrix} s^2 & s & 1\end{bmatrix}^T}{a_{1,1}s^2+a_{2,1}s+a_{3,1}} \\ \dfrac{\begin{bmatrix} s^2 & s & 1\end{bmatrix}^T}{a_{1,2}s^2+a_{2,2}s+a_{3,2}} \\ \dfrac{\begin{bmatrix} s^2 & s & 1\end{bmatrix}^T}{a_{1,3}s^2+a_{2,3}s+a_{3,3}} \end{bmatrix}, \quad \begin{aligned} \nabla J_{MTDC} &= \mathbf{e}_M^T\nabla\mathbf{e}_M \\ \nabla^2 J_{MTDC} &= \nabla\mathbf{e}_M^T\nabla\mathbf{e}_M \end{aligned}
\end{aligned}\right\} \tag{49}$$

Notice that the motor position x_M before the F_{MTDC} is implemented appears in Eq(49). This means that the gradient (∇) and Hessian (∇^2) of J_{MTDC} can be obtained in a fully data-based fashion without requiring any system model by updating based on the error profile e_M in each iteration. Numerator coefficients $\mathbf{b}=[\mathbf{b}_1\mathbf{b}_2\mathbf{b}_3]^T$ can be optimized iteratively through the following update law:

$$\begin{bmatrix}\mathbf{b}_0\\ \mathbf{b}_1\\ \mathbf{b}_2\end{bmatrix}^{n+1} = \begin{bmatrix}\mathbf{b}_0\\ \mathbf{b}_1\\ \mathbf{b}_2\end{bmatrix}^{n} - \alpha_{MTDC}\left(\nabla^2 J_{MTDC}\right)^{-1}\left(\nabla J_{MTDC}\right) \tag{50}$$

where n is the iteration counter, and α_{MTDC} is the learning gain. Bounds on α_{MTDC} can be determined as $\alpha_{MTDC} \le 2$ using discrete-time linear analysis [1], and $\alpha_{MTDC} = 1$ is used to mitigate the effect of measurement noise.

Basis function damping and natural frequencies $(\zeta_k, \omega_{n,k})$ used in defining denominator of F_{MTDC} are tuning parameters and selected approximately by analyzing motor side tracking response. However, they can also be optimized to further improve the performance of F_{MTDC}.

Denominator parameters $\mathbf{a}=[\mathbf{a}_1\ \mathbf{a}_2\ \mathbf{a}_3]^T$ in Eq(47) are also tuned through convex optimization by re-writing the basis functions from Eq(47) in gain normalized form, and re-postulating the objective function from Eq(49) with added stability $\mathbf{a}\geq\mathbf{0}$ and DC gain constraints as:

$$ {}_{MTDC}=\sum_{k=1}^{N=3}\underbrace{K_k\frac{b_{0,k}s^2+b_{1,k}s+b_{2,k}}{a_{0,k}s^2+a_{1,k}s+a_{2,k}}\frac{a_{2,k}}{b_{2,k}}}_{\Phi_k} $$

$$ \min_{\mathbf{a}=[\mathbf{a}_1\ \mathbf{a}_2\ \mathbf{a}_3]^T}\left(J_{MTDC}=\frac{1}{2}\mathbf{e}_M^T\mathbf{e}_M\right)\text{ subject to: }\begin{array}{c}\mathbf{a}\geq\mathbf{0}\\ \sum_{k=1}^{3}K_k=1\end{array} \qquad (51) $$

Note that inequality constraints in Eq(51) are introduced to ensure stability of F_{MTDC}. However, inequality constraints hinder direct use of 2nd order newton's iteration for the hardware-in-the-loop parameter update scheme given in Eq(50). Therefore, denominator coefficients are tuned through *1st* order iterations, and constraints are imposed as follows.

Firstly, *1st* order parameter update law for the denominator is written as:

$$ \begin{bmatrix}\mathbf{a}_0\\ \mathbf{a}_1\\ \mathbf{a}_2\end{bmatrix}^{n+1}=\begin{bmatrix}\mathbf{a}_0\\ \mathbf{a}_1\\ \mathbf{a}_2\end{bmatrix}^{n}+\gamma\nabla J_{MTDC} \qquad (52) $$

where $\boldsymbol{\gamma}$ is the step-size, and n is the iteration number. $\nabla J_{MTDC}= dJ_{MTDC}/d\mathbf{a}$ is the objective function gradient, and it must to be computed considering the constraints imposed in Eq(51). Consider the truncated Taylor series expansion for motor side tracking error w.r.t to the filter parameters,

$$e_M^{n+1} = e_M^n + \underbrace{\frac{\partial e_M^n}{\partial \mathbf{a}}}_{\nabla e} \Delta\mathbf{a} + HOT \tag{53}$$

and notice that $\Delta\mathbf{a} = \mathbf{a}^{n+1} - \mathbf{a}^n = \gamma\nabla J_{MTDC}$ in Eq.8. Hence, combining Eq(52) and (53), and forcing motor side tracking error at the next iteration to converge to origin yields

$$e_M^{n+1} = 0 \rightarrow e_M^n + \underbrace{\frac{\partial e_M^n}{\partial \mathbf{a}}}_{\nabla e} \Delta\mathbf{a} = 0 \tag{54}$$

where $\Delta\mathbf{a}$ can be calculated from Eq(54) through the following constrained linear least squares problem

$$\min_{\Delta a} ||\nabla e \Delta a + e_M^n||_2^2 \text{ subject to: } a \geq 0 \text{ and } \sum K_k = 1 \tag{55}$$

which can be solved easily either through use of KKT conditions or quadratic programming [24].

To summarize, denominator tuning starts with an initial guess, $\mathbf{a}^n$. F_{MTDC} is implemented, and the tracking error profile e_M is recorded. $\Delta\mathbf{a}$ is then computed from Eq(55), and denominator parameters are updated via $\mathbf{a}^{n+1} = \mathbf{a}^n + \Delta\mathbf{a}$. 1st order hardware-in-the-loop iterations are continued until parameter convergence is observed.

5.3. Table Tracking Dynamics Compensator ($\boldsymbol{F_{TTDC}}$) Design

Previous section designed motor side tracking dynamics compensator filter F_{MTDC} to ensure that motor follows reference commands precisely. However, F_{MTDC} filtered (compensated) reference motor command trajectory may excite the resonances in the flexible ball-screw drivetrain and induce unwanted vibrations on the table side. To ensure that the table follows the motor accurately, a table transmission dynamics compensator F_{TTDC} is added as shown in Figure 44.

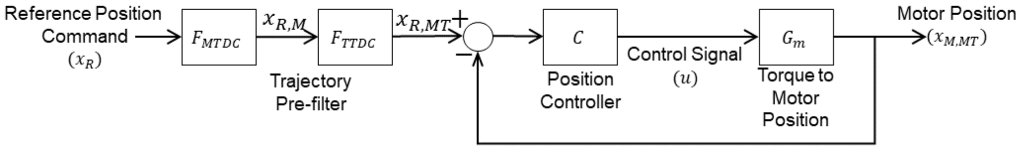

Figure 44. Proposed Control Scheme for Table Tracking Dynamics Compensator

The table transmission dynamics compensator F_{TTDC} is implemented after the motor tracking dynamics compensator is F_{MTDC} is tuned, and Figure 44 depicts overall prefilter structure. F_{TTDC} takes compensated motor command trajectory and filters it again to compensate for the motor-to-table transmission dynamics. F_{TTDC} is designed in a similar fashion to F_{MTDC} by making use of the basis function structure given in Eq(47). Denominator coefficients, i.e. resonance and damping parameters are pre-assigned, and the numerator coefficients are tuned through machine-in-the-loop iterations by minimizing the following cost function:

$$\min_{\mathbf{b}=[\mathbf{b}_1,\mathbf{b}_2,\mathbf{b}_3]^T}\left(J_{TTDC}=\frac{1}{2}\left(\mathbf{x}_{T,MT}-\mathbf{x}_{M,M}\right)^T\left(\mathbf{x}_{T,MT}-\mathbf{x}_{M,M}\right)\right) \quad (56)$$

where $\mathbf{x}_{T,MT}$ is the table position compensated by both the F_{MTDC} and F_{TTDC} compensators. Whereas, $\mathbf{x}_{M,M}$ is the F_{MTDC} compensated motor position. Notice that

Eq(56) poses a convex optimization problem as well, and its iterative solution can be guided to global optimality with the use of gradient (∇) and Hessian (∇^2) of its cost function F_{TTDC}, which can be obtained as:

$$\begin{aligned}
x_{T,MT} - x_{M,M} &= F_{TTDC} F_{MTDC} T_M G_{TM} x_R - F_{MTDC} T_M x_R \\
\nabla\left(x_{T,MT} - x_{M,M}\right) &= \nabla F_{TTDC} \underbrace{F_{MTDC} T_M G_{TM} x_R}_{x_{T,M}} \\
\nabla F_{TTDC} &= \frac{\partial F_{TTDC}}{\partial \mathbf{b}} \rightarrow \begin{cases} \nabla J_{TTDC} = \left(x_{T,MT} - x_{M,M}\right)^T \nabla\left(x_{T,MT} - x_{M,M}\right) \\ \nabla^2 J_{TTDC} = \nabla\left(x_{T,MT} - x_{M,M}\right)^T \nabla\left(x_{T,MT} - x_{M,M}\right) \end{cases}
\end{aligned}$$

(57)

In the above Eq(57), $x_{T,M}$ is the table position compensated by the F_{MTDC}, which appears in the calculation of $\nabla(x_{T,MT} - x_{M,M})$ and measurable. Therefore, ∇J_{TTDC} and $\nabla^2 J_{TTDC}$ can be obtained by using only the measurement data without relying on any system model. Once the gradient and Hessian of the objective function J_{TTDC} are obtained, Eq(57) is solved iteratively by moving each axis back-and-forth at each iteration and updating filter parameters $\mathbf{b}_k = [b_{0,k}\ b_{1,k}\ b_{2,k}]^T$ with 2nd order parameter update law:

$$\begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}^{n+1} = \begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}^{n} - \alpha_{TTDC} \left(\nabla^2 J_{TTDC}\right)^{-1} \left(\nabla J_{TTDC}\right) \qquad (58)$$

where n is the iteration counter and $\alpha_{TTDC} = 1$ is the learning rate.

Finally, tuning of the denominator coefficients of F_{TTDC} follows a similar strategy presented through Eqs(52)-(55).

5.4. Trajectory Pre-compensation Experiment Results

Parameters of the trajectory pre-compensation filter are tuned iteratively using the machine in the loop approach. Figure 45 shows the reference trajectory sent to the system as well as a marked region where the objective function of the optimization algorithm is computed. A linear acceleration region is chosen for the computation of the pre-filter parameters to avoid incorporating the effect of non-linear friction to the algorithm. Friction can be compensated separately by a friction compensator similar to [25]. At each iteration, the same trajectory is sent to both axes and the same region is used for calculating the objective function of the optimization algorithm.

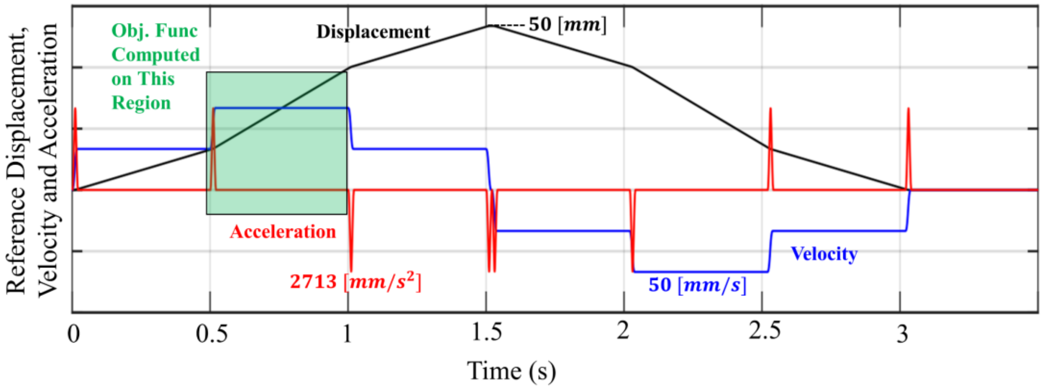

Figure 45. Reference Trajectory

Figure 46 shows the tracking dynamics of the system.

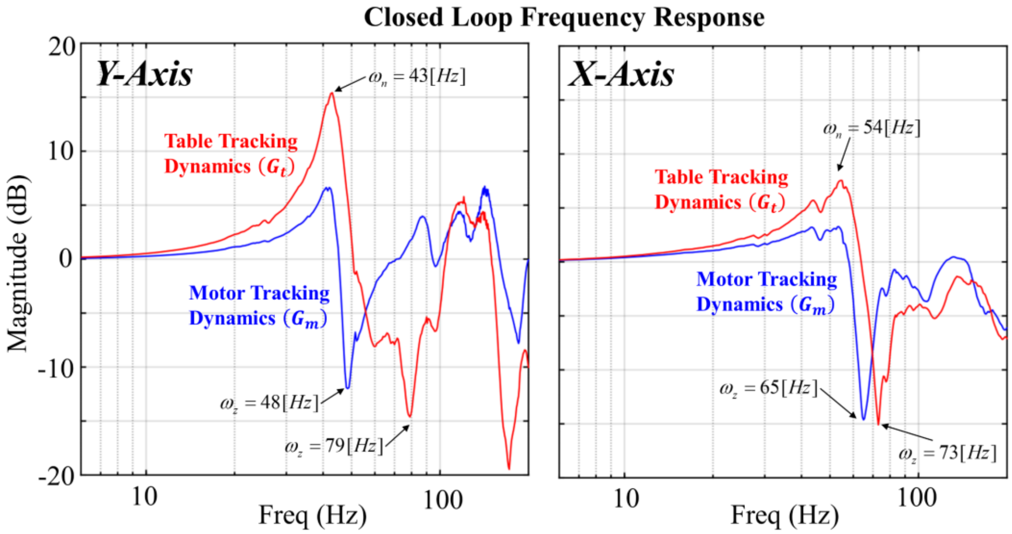

Figure 46. Tracking Dynamics

F_{MTDC} is used to compensate motor tracking dynamics. Hence, the initial denominator parameters are $\omega_{n,x} = [60, 65, 70]Hz$, $\omega_{n,y} = [45, 50, 55]Hz$, and $\zeta = 0.05$. On the other hand, F_{TTDC} is used to compensate the motor to table dynamics. Thus, the initial denominator parameters are $\omega_{n,x} = [70, 75, 80]Hz$, $\omega_{n,y} = [75, 80, 85]Hz$, and $\zeta = 0.05$. These initial parameter guesses for F_{MTDC} and F_{TTDC} are based on the approximate dominant antiresonance frequency of the dynamics that needs to be compensated. This is due to the ideal prefilter transfer function is the inverse of the dynamics that needs to be compensated, so the poles of the prefilter should be as close as possible to the zeros of the targeted dynamics.

Figure 47 shows the numerator tuning performance of F_{MTDC}. As shown, F_{MTDC} numerator tuning converges is 4 iterations and significantly reduces the motor tracking error. Once the numerator tuning is done, denominator portion of F_{MTDC} is tuned to further improve the performance of the filter. Figure 48 shows the improvement of the pre-filter performance once the denominator is tuned. The peak motor tracking error are reduced from 30 micron to 5 micron for Y-axis and from 31 micron to 3 micron for X-axis.

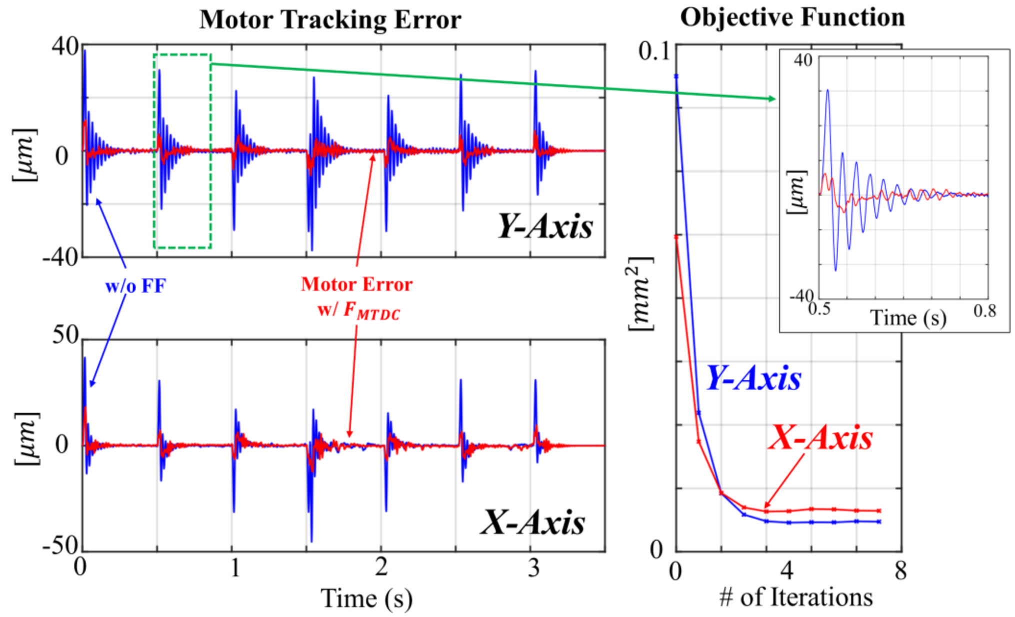

Figure 47. F_{MTDC} Numerator Tuning

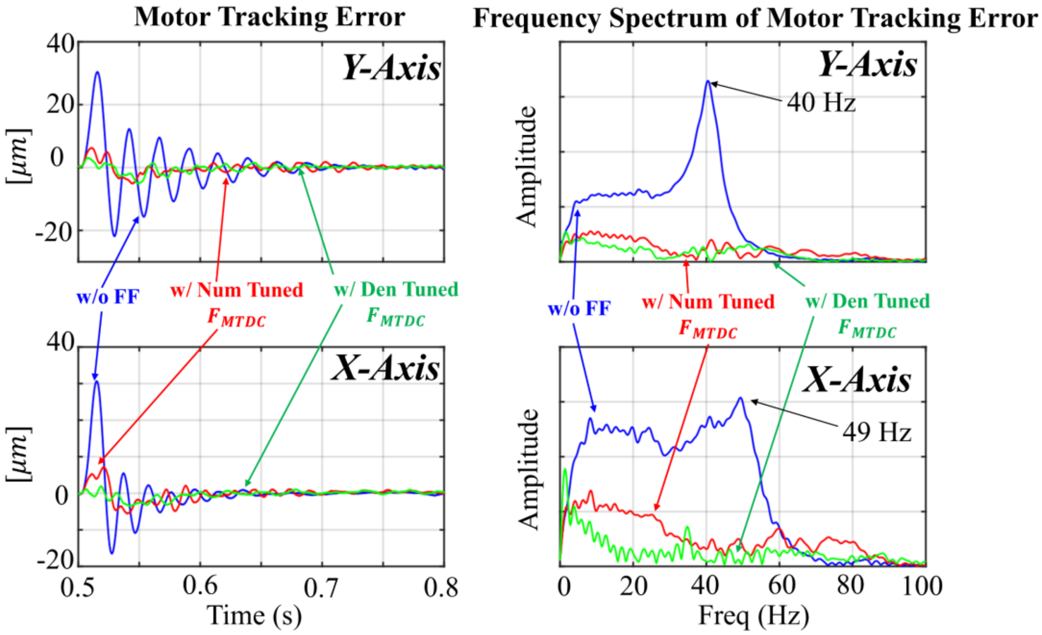

Figure 48. Impact of Denominator Tuning of F_{MTDC}

Figure 49 shows the frequency response comparison between the closed loop motor tracking and the inverse of F_{MTDC}.

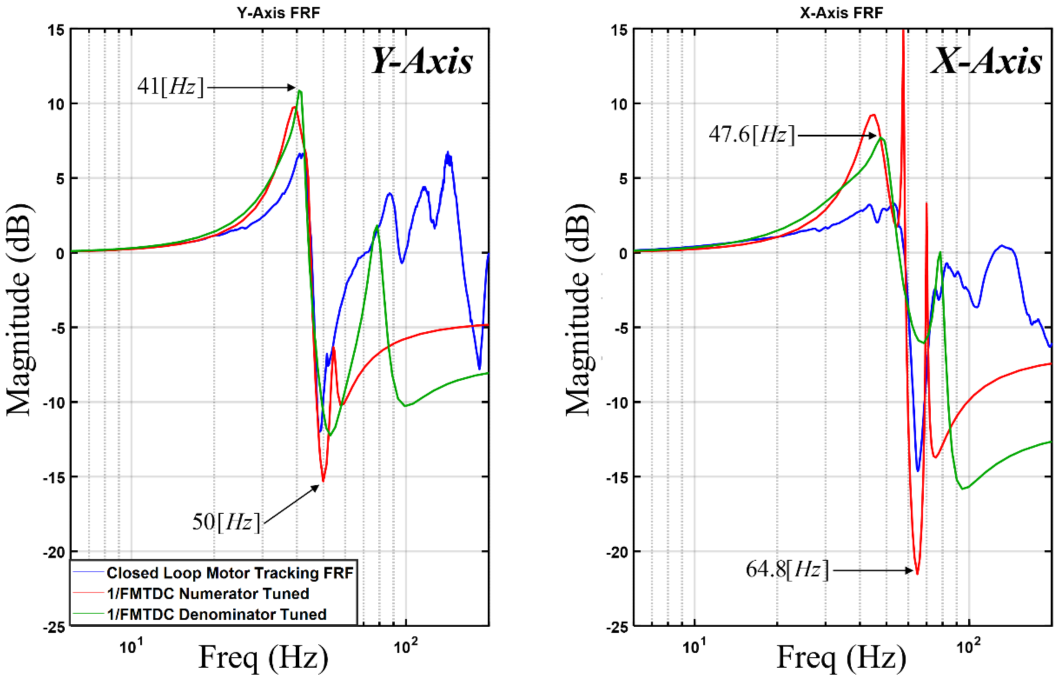

Figure 49. Frequency Response Comparison of Closed Loop Tracking and F_{MTDC}

Finally, F_{TTDC} is added to improve the table tracking performance. Figure 50 shows that F_{TTDC} also converges within a few iterations. Once converged, compensated table position $x_{T,MT}$ becomes similar to the motor position $x_{M,M}$. The peak vibration level on the table reduces from 60 micron to 8 micron for Y-axis and from 40 micron to 6 micron for X-axis.

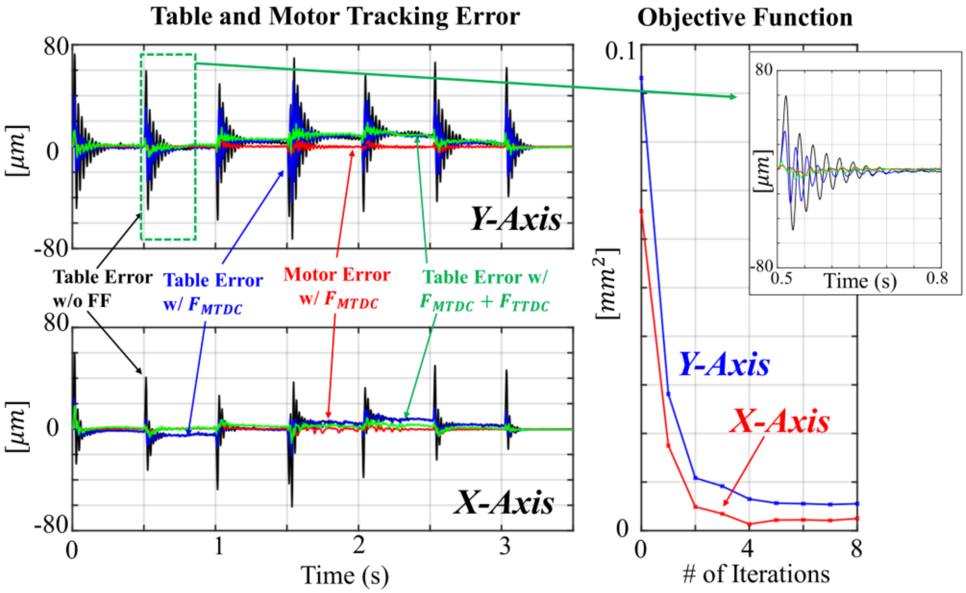

Figure 50. Performance of F_{TTDC} on Table Tracking Performance

Figure 51 shows the frequency response comparison between the motor to table dynamics and the inverse of F_{TTDC}. F_{TTDC} didn't undergo a denominator tuning because the numerator tuning performance already produces satisfactory results as shown in Figure 50. The improvement provided by the denominator tuning is negligible due to the numerator tuning already performed well.

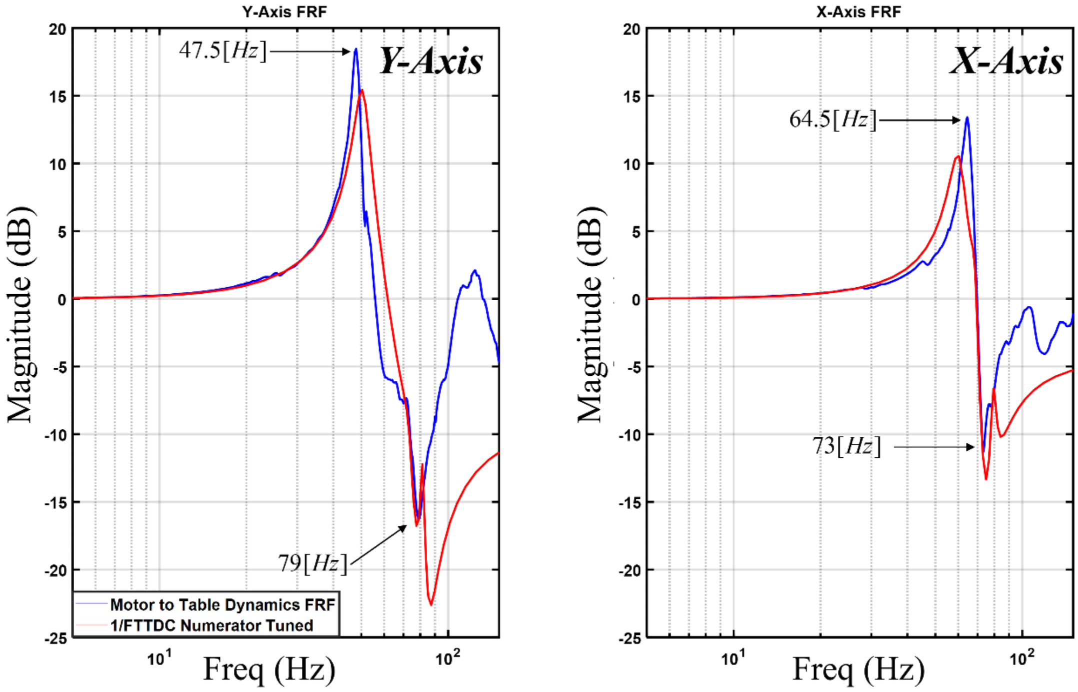

Figure 51. Frequency Response Comparison of Motor to Table Dynamics and F_{TTDC}

6. Stability Requirement of Iterative Tuning

To ensure the optimization algorithm for the pre-filter parameters converges to an optimal value, it is required to operate within a stable learning gain γ range. This section presents a mathematical procedure to obtain the range of convergence for γ .

The presented analysis is applied on F_{TTDC} tuning algorithm. However, it can also be applied to F_{MTDC} tuning since both prefilters utilize similar tuning algorithms. The method to find the convergence range of the learning gain focuses on the propagation equation of the newton's method $a^{(k+1)} = a^{(k)} - \gamma\left[\nabla^2 J^{(k)}\right]^{-1}\left[\nabla J^{(k)}\right]$ [1],[26].

In this method, the system representation needs to be converted from a transfer function form to a lifted form which utilized toeplitz matrix [27],[28]. A toeplitz matrix is a diagonal constant matrix with each descending diagonal from left to right is a constant. To convert from transfer function form to lifted form, a state space representation has to be constructed from the transfer function. Consider a transfer function:

$$F(s) = \frac{x_{R,MT}}{x_{R,M}} = \frac{a_0 s^2 + a_1 s + a_2}{s^2 + b_1 s + b_2} \tag{59}$$

To convert the equation above to a state space domain, a placeholder parameter is required. In this case, a placeholder z is introduced to Eq(59) as shown below.

$$\begin{aligned} x_{R,MT} &= \left(a_0 s^2 + a_1 s + a_2\right) z \\ x_{R,M} &= \left(s^2 + b_1 s + b_2\right) z \end{aligned} \tag{60}$$

Then, Eq(60) is turned in a state space representation as follows.

$$\text{Let} \begin{array}{ll} q_1 = z & \dot{q}_1 = q_2 \\ q_2 = \dot{z} & \ddot{q}_2 = x_{R,M} - b_1 q_2 - b_2 q_1 \end{array} \tag{61}$$

Substituting Eq(61) to Eq(60) results in Eq(62).

$$\begin{aligned} x_{R,MT} &= a_0 \left(x_{R,M} - b_1 q_2 - b_2 q_1 \right) + a_1 q_2 + a_2 q_1 \\ x_{R,MT} &= a_0 \left(x_{R,M} - b_1 \dot{z} - b_2 z \right) + a_1 \dot{z} + a_2 z \\ x_{R,MT} &= a_0 x_{R,M} + \left(a_1 - a_0 b_1 \right) \dot{z} + \left(a_2 - a_0 b_2 \right) z \end{aligned} \tag{62}$$

Then, the state space representation becomes:

$$\begin{gathered} \dot{q} = Aq + Bx_{R,M} \quad x_{R,MT} = Cq + Dx_{R,M} \\ \text{where} \\ A = \begin{pmatrix} 0 & 1 \\ -b_2 & -b_1 \end{pmatrix} \qquad B = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ C = \begin{pmatrix} a_2 - a_0 b_2 & a_1 - a_0 b_1 \end{pmatrix} \quad D = a_0 \end{gathered} \tag{63}$$

Where $[a_0\ a_1\ a_2\ b_1\ b_2]$ are the numerator and denominator coefficients of the transfer function. Then, the state space is converted from continuous domain to discrete domain and the lifted form is derived from the discrete state space form as Eq(64).

$$\begin{gathered} q(n+1) = A_d q(n) + B_d q(n) \quad x_{R,MT} = C_d q(n) + D_d x_{R,M} \\ \text{where} \\ A_d = e^{AT_s} \quad B_d = A^{-1}(A_d - I)B \\ C_d = C \qquad D_d = D \\ x_{R,MT} = \underbrace{\begin{pmatrix} D_d & 0 & 0 & \vdots & 0 \\ C_d B_d & D_d & \ddots & \ddots & \vdots \\ C_d A_d B_d & C_d B_d & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ C_d A_d{}^{N-2} B_d & C_d A_d{}^{N-2} B_d & \cdots & C_d B_d & D_d \end{pmatrix}}_{T_{x_{R,M} \to x_{R,MT}}} x_{R,M} \end{gathered} \tag{64}$$

where $T_{x_{R,M}\rightarrow x_{R,MT}}$ is the lifted form of the transfer function from $x_{R,M}$ to $x_{R,MT}$ which in this case is FTTDC.

The objective function for FTTDC parameter tuning is $j^{(k)} = x_{T,MT} - x_{M,M}$ and it can be rewritten in the Toeplitz domain as Eq(65).

$$j^{(k)} = \underbrace{T^{(k)}_{x_{R,M}\rightarrow x_{T,MT}} x_{R,M}}_{x_{T,MT}} - \underbrace{T^{(k)}_{x_{R,M}\rightarrow x_{M,M}} x_{R,M}}_{x_{M,M}} \tag{65}$$

Then, $T^{(k)}_{x_{R,M}\rightarrow x_{T,MT}} x_{R,M}$ can be expanded to extract the prefilter numerator a as shown below.

$$T^{(k)}_{x_{R,M}\rightarrow x_{T,MT}} x_{R,M} = T^{(k)}_{x_{R,MT}\rightarrow x_{T,MT}} T^{(k)}_{b} \xi a^{(k)}$$

where

$$T^{(k)}_{b} \text{ is the lifted form of } \frac{1}{s^2 + b_1 s + b_2} \tag{66}$$

$$\xi = \begin{pmatrix} \dfrac{d^2 x_{R,M}}{dt^2} & \dfrac{dx_{R,M}}{dt} & x_{R,M} \end{pmatrix}$$

$$a^{(k)} = \begin{pmatrix} a^{(k)}_0 \\ a^{(k)}_1 \\ a^{(k)}_2 \end{pmatrix}$$

The propagation equation of the optimization can be written similar to [1] as:

$$\begin{aligned} a^{(k+1)} &= a^{(k)} - \gamma \left[\nabla^2 J^{(k)}\right]^{-1} \left[\nabla J^{(k)}\right] \\ a^{(k+1)} &= a^{(k)} - \gamma \left[\nabla j^{(k)T} \nabla j^{(k)}\right]^{-1} \left[\nabla j^{(k)} j^{(k)}\right] \end{aligned} \tag{67}$$

Then, the hessian and gradient term of the equation above can be written in Toeplitz form as shown in Eq(68).

$$\begin{gathered} a^{(k+1)} = a^{(k)} - \gamma \left[\left(T^{(k)}_{x_{R,MT}\rightarrow x_{T,MT}} T^{(k)}_{b} \xi \right)^{T} \left(T^{(k)}_{x_{R,MT}\rightarrow x_{T,MT}} T^{(k)}_{b} \xi \right) \right]^{-1} \cdots \\ \left[\left(T^{(k)}_{x_{R,MT}\rightarrow x_{T,MT}} T^{(k)}_{b} \xi \right)^{T} \left(T^{(k)}_{x_{R,MT}\rightarrow x_{T,MT}} T^{(k)}_{b} \xi a^{(k)} - T^{(k)}_{x_{R,M}\rightarrow x_{M,M}} x_{R,M} \right) \right] \end{gathered} \tag{68}$$

The equation above is rearranged to obtain the propagation matrix η from $a^{(k+1)} = \eta a^{(k)} + c$ where c is a constant. From a linear system theory, the linear discrete time system is stable if the eigenvalues of the propagation matrix η are within the unit circle, $\left|\lambda\left[\eta\right]\right| < 1$[29]. Thus, the propagation matrix η can be calculated from Eq(68) as follows:

$$a^{(k+1)} = \underbrace{\left\{ I - \gamma \left[\left(T^{(k)}_{x_{R,MT} \rightarrow x_{T,MT}} T_b^{(k)} \xi \right)^T \left(T^{(k)}_{x_{R,MT} \rightarrow x_{T,MT}} T_b^{(k)} \xi \right) \right]^{-1} \left[\left(T^{(k)}_{x_{R,MT} \rightarrow x_{T,MT}} T_b^{(k)} \xi \right)^T \left(T^{(k)}_{x_{R,MT} \rightarrow x_{T,MT}} T_b^{(k)} \xi \right) \right] \right\}}_{\eta} a^{(k)} + \ldots$$

$$\gamma \left[\left(T^{(k)}_{x_{R,MT} \rightarrow x_{T,MT}} T_b^{(k)} \xi \right)^T \left(T^{(k)}_{x_{R,MT} \rightarrow x_{T,MT}} T_b^{(k)} \xi \right) \right]^{-1} \left[\left(T^{(k)}_{x_{R,MT} \rightarrow x_{T,MT}} T_b^{(k)} \xi \right)^T \left(T^{(k)}_{x_{R,M} \rightarrow x_{M,M}} \right) \right]$$

$$\eta = 1 - \gamma$$

(69)

Since $\eta = 1 - \gamma$, convergence is guaranteed for $0 < \gamma < 2$.

7. Conclusion

This thesis presented various novel and systemic control system design method ranging from active to passive structural vibration mitigation.This thesis presented various novel and systemic control system design method ranging from active to passive structural vibration mitigation. These vibrations include motor, table, spindle and also the relative motion between the table and the spindle. The proposed active vibration mitigation method include loop shaping acceleration feedback and state space spindle feedback method. On the other hand, the proposed passive vibration mitigation method include trajectory pre-compensation filter design method. The improvement in dynamics positioning accuracy when passive vibration mitigation method is implemented is significant, around 85% reduced peak error on the table side. However, the improvement is dynamics positioning accuracy when active vibration feedback is implemented is smaller than that of the passive vibration mitigation. This is mainly caused by inaccurate modelling. Overall, the developed method is able to improve the accuracy of high-speed machine tools without modifying the hardware.

Bibliography

[1] S. H. Van Der Meulen, R. L. Tousain, and O. H. Bosgra, “Fixed structure feedforward controller design exploiting iterative trials: Application to a wafer stage and a desktop printer,” *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 130, no. 5, pp. 0510061–05100616, 2008, doi: 10.1115/1.2957626.

[2] D. Bruijnen and N. Van Dijk, “Combined input shaping and feedforward control for flexible motion systems,” *Proceedings of the American Control Conference*, pp. 2473–2478, 2012, doi: 10.1109/acc.2012.6315055.

[3] H. Butler, “Acceleration feedback in a lithographic tool,” *Control Engineering Practice*, vol. 20, no. 4, pp. 453–464, 2012, doi: 10.1016/j.conengprac.2011.12.008.

[4] J. Munoa, X. Beudaert, K. Erkorkmaz, A. Iglesias, A. Barrios, and M. Zatarain, “Active suppression of structural chatter vibrations using machine drives and accelerometers,” *CIRP Annals - Manufacturing Technology*, vol. 64, no. 1, pp. 385–388, 2015, doi: 10.1016/j.cirp.2015.04.106.

[5] A. H. H. Hosseinabadi and Y. Altintas, “Modeling and active damping of structural vibrations in machine tools,” *CIRP Journal of Manufacturing Science and Technology*, vol. 7, no. 3, pp. 246–257, 2014, doi: 10.1016/j.cirpj.2014.05.001.

[6] T. Kai, H. Sekiguchi, and H. Ikeda, “Relative vibration suppression in a positioning machine using acceleration feedback control,” *IEEJ Journal of Industry Applications*, vol. 7, no. 1, pp. 15–21, 2018, doi: 10.1541/ieejjia.7.15.

[7] Y. Marushita, H. Ikeda, and H. Sugie, “Vibration suppression control using the load-side acceleration feedback,” *IECON Proceedings (Industrial Electronics Conference)*, pp. 810–815, 2007, doi: 10.1109/IECON.2007.4460028.

[8] Y. Marushita, H. Ikeda, and H. Sugie, “Vibration suppression control using the load-side acceleration feedback,” *IECON Proceedings (Industrial Electronics Conference)*, pp. 810–815, 2007, doi: 10.1109/IECON.2007.4460028.

[9] T. Kai, H. Sekiguchi, and H. Ikeda, “Relative vibration suppression in a positioning machine using acceleration feedback control,” *IEEJ Journal of Industry Applications*, vol. 7, no. 1, pp. 15–21, 2018, doi: 10.1541/ieejjia.7.15.

[10] A. Dumanli and B. Sencer, “CIRP Annals - Manufacturing Technology Pre-compensation of servo tracking errors through data-based reference trajectory modification,” vol. 68, pp. 397–400, 2019.

[11] D. J. Gordon and K. Erkorkmaz, “Accurate control of ball screw drives using pole-placement vibration damping and a novel trajectory prefilter,” *Precision Engineering*, vol. 37, no. 2, pp. 308–322, 2013, doi: 10.1016/j.precisioneng.2012.09.009.

[12] Y. Hosseinkhani and K. Erkorkmaz, “High Frequency Harmonic Cancellation in Ball-screw Drives,” *Procedia CIRP*, vol. 1, pp. 615–620, 2012, doi: 10.1016/j.procir.2012.05.009.

[13] L. Blanken, F. Boeren, D. Bruijnen, and T. Oomen, “Batch-To-Batch Rational Feedforward Control: From Iterative Learning to Identification Approaches, with Application to a Wafer Stage,” *IEEE/ASME Transactions on*

Mechatronics, vol. 22, no. 2, pp. 826–837, 2017, doi: 10.1109/TMECH.2016.2625309.

[14] J. Boot, “Frequency response measurement in closed loop: brushing up our knowledge,” p. 34, 2003.

[15] D. Alper, “Optimal High-bandwidth Non-collocated Control of Ball-screw Drives,” Oregon State University, Corvallis, 2018.

[16] K. Erkorkmaz and Y. Altintas, “High speed CNC system design. Part II: modeling and identification of feed drives,” 2001.

[17] J. Abir, S. Longo, P. Morantz, and P. Shore, “Optimized estimator for real-time dynamic displacement measurement using accelerometers,” *Mechatronics*, vol. 39, pp. 1–11, 2016, doi: 10.1016/j.mechatronics.2016.07.003.

[18] J. Kautsky, N. K. Nichols, and P. Van Dooren, “Robust pole assignment in linear state feedback,” *International Journal of Control*, vol. 41, no. 5, pp. 1129–1155, 1985, doi: 10.1080/0020718508961188.

[19] S. Tajima, B. Sencer, and E. Shamoto, “Accurate interpolation of machining tool-paths based on FIR filtering,” *Precision Engineering*, vol. 52, no. December 2017, pp. 332–344, 2018, doi: 10.1016/j.precisioneng.2018.01.016.

[20] J. Vaughan, A. Yano, and W. Singhose, “Comparison of Robust Input Shapers,” *Journal of Sound and Vibration*, vol. 315, no. 4–5, pp. 797–815, 2008, doi: 10.1016/j.jsv.2008.02.032.

[21] H. Hjalmarsson, “Iterative feedback tuning - An overview,” *International Journal of Adaptive Control and Signal Processing*, vol. 16, no. 5, pp. 373–395, 2002, doi: 10.1002/acs.714.

[22] M. Tomizuka, “Zero Phase Error Tracking Algorithm for Digital Control,” vol. 109, no. March 1987, pp. 65–68, 1987.

[23] S. H. van der Meulen, R. L. Tousain, and O. H. Bosgra, “Fixed structure feedforward controller design exploiting iterative trials: Application to a wafer stage and a desktop printer,” *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 130, no. 5, pp. 0510061–05100616, Sep. 2008, doi: 10.1115/1.2957626.

[24] S. Boyd and L. Vandenberghe, *Convex Optimization*. 2004. doi: 10.1017/cbo9780511804441.

[25] A. Dumanli and B. Sencer, “Data-Driven Iterative Trajectory Shaping for Precision Control of Flexible Feed Drives,” *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 5, pp. 2735–2746, 2021, doi: 10.1109/TMECH.2020.3045444.

[26] M. Q. Phan and J. A. Frueh, “Learning control for trajectory tracking using basis functions,” *Proceedings of the IEEE Conference on Decision and Control*, vol. 3, no. December, pp. 2355–3592, 1996, doi: 10.1109/cdc.1996.573465.

[27] S. Van Der Meulen, R. Tousain, and O. Bosgra, “Fixed structure feedforward controller tuning exploiting iterative trials, applied to a high-precision electromechanical servo system,” *Proceedings of the American Control Conference*, no. 1, pp. 4033–4039, 2007, doi: 10.1109/ACC.2007.4282530.

[28] A. Kazemi Amiri and C. Bucher, “Derivation of a new parametric impulse response matrix utilized for nodal wind load identification by response

measurement," *Journal of Sound and Vibration*, vol. 344, pp. 101–113, May 2015, doi: 10.1016/j.jsv.2014.12.027.

[29] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice-Hall Information and System Series, 1980.