Digital Twin Feed Drive Identification for Virtual Process Planning

by

Ginette Wei Get Tseng

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Masters of Applied Science
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2018

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Computer numerical controlled (CNC) machines have become an integral part of the manufacturing industry, allowing companies to increase the accuracy and productivity of their manufacturing lines. The next step to improving and accelerating the development process of a part is to involve virtual prototyping during the design phases. Virtual manufacturing has become an invaluable tool to process planners and engineers in recent years to model the manufacturing environment in a virtual setting to determine the final geometry and tolerances of new parts and processes. For a virtual twin of a CNC machine to be built, the dynamics of the drive and CNC controller must be identified. Traditionally, these identification techniques require several intrusive tests to be run on the machine tool, causing valuable time lost on production machines. In this thesis, three new techniques of developing virtual models of machine tools are discussed.

The first model presented is a quasi-static model which is suitable for trajectory tracking error prediction. This technique is used to determine the contributions of the commanded velocity, acceleration, and jerk to the tracking errors of each axis of the machine tool. After determining these contributions, process planners can modify the axis feedrates in a virtual environment during trajectory optimization to find the best parameters for the shortest cycle time. This method was validated using a laser drilling machine tool from Pratt and Whitney Canada (P&WC) and was able to predict the root mean square (RMS) of the tracking error within 2.62 to 11.91 μm. A simple graphical user interface (GUI) was developed so that process planners and engineers can import data collected from the FANUC and Siemens CNC controllers to identify quasi-static models.

The second model presented is a single input – single output (SISO) rigid body rapid identification model. In previous literature, a rapid identification method was proposed where a short G-code was run on machine tools, the input and output signals were collected from the controller and the dynamics were reverse engineered from the gathered data. However there were some shortfalls with this older method, the new proposed rapid identification model addresses these by improving parameter convergence and using commanded signal derivatives for identification. Tests were conducted on a five-axis machine tool located at the University of Waterloo (UW) to verify and compare the new rapid identification model to the previous model. It was determined that the model is able to predict the RMS of the tracking errors with 50-76% improvement and maximum contour error discrepancy with 22-35% improvement. Another GUI was developed for the SISO rigid body rapid identification model that allows users to import data collected from different machine tools and identify a model.

The third model that is discussed in this thesis is a multi input – multi output (MIMO) model. This model builds upon the SISO rigid body model and is able to capture vibratory and elastic dynamics. Relations between inputs, such as reference and disturbance signals, can be related to a variety of measurable outputs. The model is used to predict the relationship between the inputs of commanded position and disturbance to the outputs of tracking error and velocity of the x- and y- axes of a P&WC five axis milling machine tool. Three different models were identified using this algorithm, two 1-axis *3*rd order decoupled models and two 2-axis *6*th order coupled model are compared in this thesis. The two 6^{th} order models have different search spaces, the first has a search space defined from the *3*rd order decoupled identified parameters while the second has a more general search space. Overall, the *6*th order model with a larger search space was able to predict the RMS and maximum tracking error more closely, with a maximum improvement of 19% for both metrics. However it should be noted that *6*th order model with a smaller search space was still able to predict the RMS and maximum tracking error similarly to the *6*th order model with the larger search space. The smaller search space configuration can save on computational time which can be advantageous in real world applications.

In order to verify that the MIMO rapid identification technique would be able to identify a vibration mode, an experimental setup was designed and machined. A flexure mount with known vibration modes was designed, built and tested to validate Solidworks frequency simulation results. It was concluded that the simulation results were able to estimate the frequencies of the flexure with 95-98% accuracy and with a maximum absolute difference of 2.87 Hz.

The flexure was mounted onto the five-axis machine tool at UW to introduce vibratory dynamics. Since there is a flexible mode being introduced at the tool-workpiece interface, the motor encoders would not be able to capture these dynamics, therefore a two-dimensional grid encoder (KGM) and two 3-axis accelerometers (one located on the tool head and the other on the workpiece table) were also placed on the machine tool to record the true tool-workpiece response. The data collected from the accelerometers were corrected for possible roll, pitch and yaw misalignments before synchronizing the accelerometer and KGM data to the motor encoder data. This data was then used to build MIMO rapid identification models with the commanded position (recorded from the motor encoders) and normalized Coulomb disturbance as the inputs to the system and the true tool-workpiece position or acceleration and machine tool feed drive velocity as the outputs to the model. The model estimated from the position measurements from the KGM yielded better results 19-1496% improvement in RMS tracking error prediction over the acceleration based models. The contouring error estimated using the KGM position model also has an improvement of 233-370% over the acceleration models. Using the transfer functions estimated from the accelerometer data,

there was a 16-33% improvement in the RMS tracking error prediction and an 11-51% improvement in the maximum tracking error prediction over the KGM acceleration based model. The RMS contour prediction error also improved 4-5% and the maximum contour error prediction improved by 1-6% between the two models.

Further development into the MIMO LTI algorithm is currently being done in the laboratory, including research into more complex friction models. It is also recommended to machine an actual part on the five-axis machine tool and to measure the contouring error of the part on the coordinate measuring machine to verify the predictions presented in this thesis.

Acknowledgements

I would like to thank everyone at Pratt & Whitney Canada, especially Dr. Serafettin Engin, Dr. William Ferry, Dr. Jochem Roukema, Mr. Frederic Alexandre and Mr. Samuel Jacques, for assistance in validating my research by providing machine time, materials, and tools at their facility as well as their knowledge in the various manufacturing processes. I would also like to thank Michael Ellis from FANUC Canada for offering continued technical support.

I would like to thank Mr. Robert Wagner, Mr. Jason Benninger and Mr. Neil Griffett for their assistance with my experiments and valuable knowledge. I would also like to thank my colleagues in the Precision Controls Laboratory, my friends and my family for their support. Portions of this research were also conducted with Mr. Mustafa Hakan Turhan, which are outlined in the thesis. Finally, I would like to thank my advisor Dr. Kaan Erkorkmaz for his guidance, support and encouragement throughout my studies, the knowledge and skills he has shared have been key to the success of my research.

This research was sponsored by NSERC and Pratt & Whitney Canada through grant number CRDPJ 462114-13.

Dedication

To my parents and my brother.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

In the manufacturing industry, many companies use computer numerical controlled (CNC) machines to mass produce parts that have complex geometries. These machines are able to achieve very high productivity rates and are capable of creating components with high precision and repeatability. However, before the CNC machines can be used to produce these parts, it is essential for the manufacturing process and the performance of the machine tool to be optimized through prototyping and production trials. A program must be defined, tested and verified that it can create a part with the desired quality. Only then can the program be run with confidence that the components will be manufactured within the given specifications, usually with periodic human intervention for verification. Unfortunately, this prototyping process can be very time consuming, resulting in downtime on the manufacturing line and in wasted materials.

The manufacturing sector is currently moving towards its fourth revolution, often called “Industry 4.0”, and one of the main goals is to provide companies with complete digitalization of their facilities and processes. There are many benefits to the digitalization of businesses such as shortening products time to market, improving the quality of the parts, and increasing the flexibility and efficiency of the plant [1]. To do this, digital twins of the machine tools and processes must be created in a cyber-physical environment. By creating these digital twins, virtual manufacturing/prototyping can be used to minimize cost and time losses when developing new processes or improving older ones, reducing overall turnaround time. This prototyping method can be used to determine the shortest cycle time for these processes while keeping the part within the specified tolerances, which in turn results in the lowest manufacturing cost. Powerful computers that are capable of performing these simulations have become more readily available, making this method more appealing to companies.

Virtual CNC (VCNC) [2] software, as seen in Figure 1-1, has been developed to predict the tracking and contouring performance of machine tools. In order for virtual manufacturing to be effective in its predictions, it is critical for an accurate model of the actual machine tool to be created. These models must account for several different factors, such as the actual process (trajectory, velocity, acceleration, and jerk), the dynamics of the machine tool, the controller, and the relationships between the three. Traditionally, these models are very time consuming and costly to develop, requiring a qualified dynamics and controls engineer to run several tests on the machine tool. Some of these tests require certain controller settings, such as the servo loop or the trajectory interpolator, to be turned off while some identification signals are

injected into the servo system. These tests are not practical for manufacturing companies, as these tests often cause significant downtime on the machines.

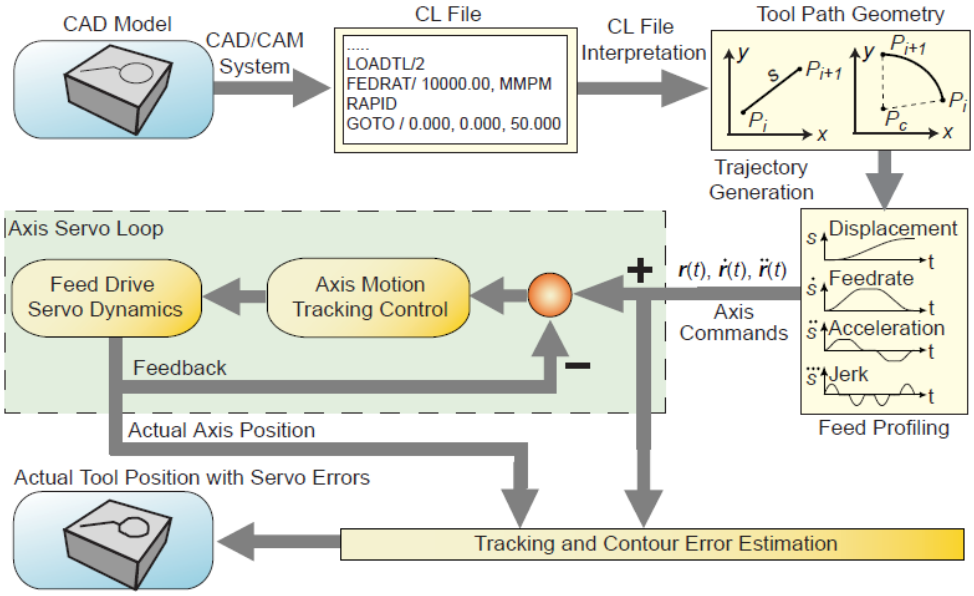


Figure 1-1 Overview of VCNC system [2]

An alternative technique called "rapid identification" was proposed in [3], where a model was identified by executing a short G-code, and collecting the input and output data using the oscilloscope function of the machines (further discussed in Chapter 2). In this thesis, three new rapid identification techniques, which use in-process gathered data to identify models, will be discussed. The first method is a quasi-static model, which can be identified very quickly and can predict the instantaneous tracking error of rigid machine tools. The second is a single input – single output (SISO) rigid body model, which takes into account the poles and friction of the machine tool and is able to predict the tracking error. The final model is a multi input – multi output (MIMO) model that is able to predict vibration modes as well as several outputs (current, tracking error, position, etc.) using several inputs (commanded position, friction, velocity, acceleration, etc.).

Chapter 2
Literature Review

2.1 Introduction

In the manufacturing industry, computer aided design and computer aided manufacturing software, or more commonly known as CAD/CAM software, have become an irreplaceable tool that assists designers, process planners, and engineers to expedite the design and production of new parts. Ever since powerful computers capable of performing the complex calculations and simulations required for these tools have become more accessible to the general public, several software companies have been moving towards an end-to-end, fully digital manufacturing environment. These newer software updates often include modules that allow designing, simulation, optimization and quality assurance to occur all within a virtual environment.

Some CAD/CAM packages, such as MasterCAM [4] and GibbsCAM [5], contain general features for collision detection, trajectory and feedrate optimization, and toolpath visualization. Other programs, such as Autodesk Fusion [6] and Vericut [7] [8], also include an option to create CNC probing subroutines, which can maximize the productivity of the machine tools, while minimize the risk of breaking an expensive probe tool on the machine floor. However, these software solutions only provide a simulation of what would occur in an ideal manufacturing environment. The effects of the machine tool dynamics and the controller must also be included in the computations for more accurate simulation of the actual manufacturing process. Some companies have started to include these important factors into their software packages, allowing users to create a digital twin of their CNC in a virtual environment.

2.2 Virtual CNCs

The Virtual CNC (VCNC) platform developed by the Manufacturing Automation Laboratory at the University of British Columbia [9] [10] and Siemen's Virtual NC Kernel [11], include the machine tool dynamics and CNC control systems, and the interaction between the two to provide accurate simulations and predictions. These software packages allow for a digital twin of a machine tool to be created, which allows engineers and process planners to design and fully optimize a process in a virtual environment. The main goal of this virtual prototyping process is to have the first part fabricated on the machine tool to be within the specified tolerances, with very little or no alterations to the manufacturing procedures on the production floor.

In the University of British Columbia's VCNC, there are three main modules: 1) toolpath interpolation, 2) drive response simulation, and 3) evaluation of performance. This thesis focuses on the model that will be used for the simulation of the drive response. The software includes several libraries where the user can configure the drive model, for both ball screw and direct feed drives. For the servo controller configuration, several control laws are included in the libraries, such as simple P, PID, P-PI cascade controllers to more complex structures such as pole-placement [12], generalized predictive [13], adaptive slide mode [14] [15] [16], and feedforward controller. Users can also supply a transfer function to the software, without any physical parameters/meanings, in either the s- or z-domain.

2.2.1 Tracking and Contouring Error

In the VCNC, the contouring accuracy of the part program is estimated and can be used for further optimization by doing more prototyping within the virtual environment. In Figure 2-1 below, the tracking error and contour error of a trajectory is illustrated.

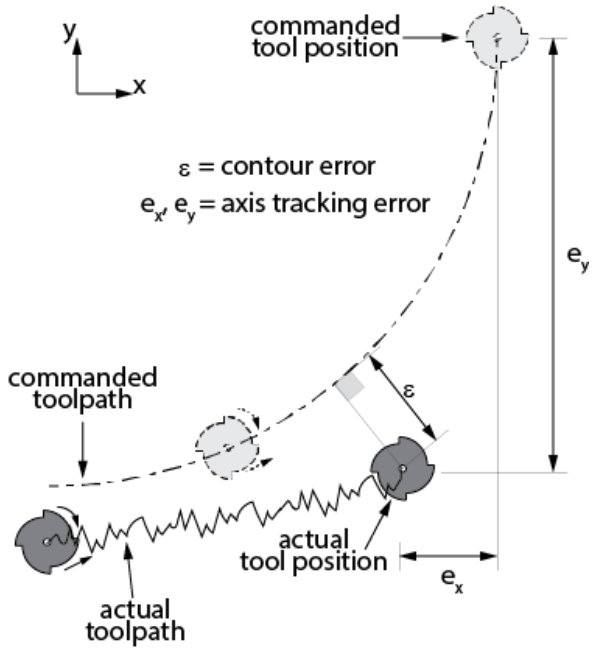


Figure 2-1 Tracking vs. contouring error

The tracking error is defined as the difference in distance between the commanded tool position and the actual tool position at a specific time step. The contour error is the minimum normal distance between the commanded toolpath and the actual toolpath at any point in time. There are two main methods to improve the control the performance of drive systems: 1) tracking control, and 2) contouring control [17]. The tracking control approach only focuses on reducing the tracking error in each individual axis, which

indirectly also reduces the overall contouring error. The contouring control method instead focuses on estimating the real-time contouring error and injects this signal into the feedback controller. In [18], it was shown that when different axes have matched dynamics, the contouring error of the machine tool is minimized. However, when the dynamics of the different axes are not well matched, due to a gain mismatch that can be caused by several factors, the system accuracy is affected and the contour errors become unacceptable in a manufacturing environment. Taking into account both the tracking and contouring errors of a process is very important in the quality of the final manufactured product and should be monitored during the processes planning and design phases.

2.3 Modelling and Identification of Feed Drives

In order for virtual prototyping to create accurate simulations of the manufacturing process, an accurate model of the machine tool must be constructed. The most widely used basic rigid body model is found in [19] [20], and a simplified diagram can be seen in Figure 2-2.

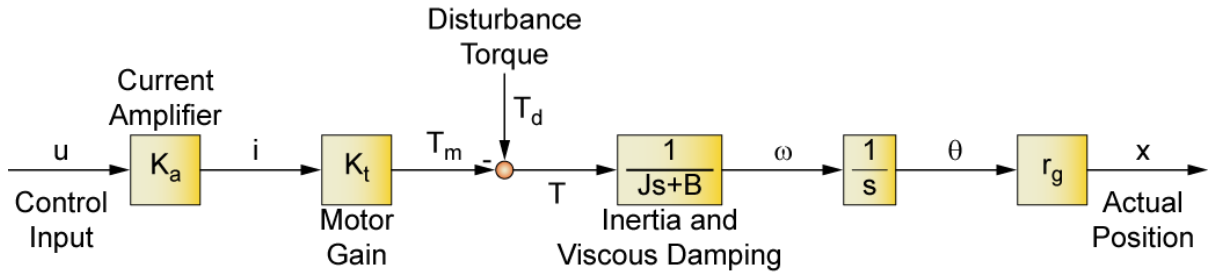


Figure 2-2 Rigid body model

The commanded current input, u (V), is applied to the current amplifier gain, K_a (A/V). This produces the current in the motor armature, i (A), which is then multiplied by the motor gain, K_t (N·m/A), to produce the motor torque, T_m (N·m). The model also includes a disturbance torque, T_d (N·m), which can originate from many sources including the guideway, bearing, ball screw friction, and cutting forces. The actual torque, T (N·m), is then applied to the mechanical system, which is represented by the equivalent inertia, J (kg·m^2), and the viscous damping, B (kg·m^2/s). This results in an angular velocity, ω (rad/s), of the motor shaft, which is integrated to obtain the angular position, θ (rad). The gear ratio, r_g (mm/rad), is then applied to calculate the linear axis movement, x (mm).

One major downfall of rigid body models are that they are not able to capture the effect of structural vibrations that occur at higher bandwidths. In literature, several researchers have proposed finite element

models of ball screw drives, and have identified torsional and axial vibrations from frequency response experiments [21] [22] [23] [24].

In literature, there are three main groups of system identification techniques: subspace, nonparametric correlation and spectral analysis, and prediction error based techniques [25]. System identification experiments are usually carried out in closed-loop or partially closed-loop conditions for safety, economic, and quality reasons [25] [26]. Although the subspace and nonparametric methods work well with open-loop data, one major drawback is that they do not work well with closed-loop data due to the difficulties of correlating measurement noise with the applied input [25] [26]. The prediction error based methods can be separated into three categories: direct, indirect, and joint input-output [27]. All the prediction error based techniques require a control input of sufficient excitation to be applied to the closed-loop system, such as step or pseudo random binary sequence (PRBS) input. However, this type of input is difficult to execute on a CNC machine tool since motion commands are usually interpolated to be continuous at the acceleration and/or jerk level. This can result in a lack of excitation in the machine tool which makes the identification of multiple transfer function parameters very difficult as they may not converge [28].

Aside from cutting and process forces, one of the main disturbances in machine tools that causes positioning errors is friction. Therefore, modeling and identifying the friction of a feed drive is essential to the accuracy of a VCNC. In literature, friction models are grouped into two categories: static (classic) and dynamic (memory based) [29]. In the classic static model, the applied (input) force and velocity is used to calculate the instantaneous friction force [9] [30]. For the memory based models, the sticking and sliding regimes must be characterized, for which several models have been proposed in literature [31] [32] [33] [34]. Friction models that require experimentally gathered time domain data have been studied extensively in [20] [35] [36], while frequency domain friction models have also been studied in [37] [38] [39]. In previous VCNC studies, it has been shown that simple Coulomb friction models provide sufficient accuracy in predicting quadrant glitch servo errors caused by motion reversals [3] [28].

2.4 Rapid Identification

The modeling and identification methods mentioned in the previous section can provide accurate models of machine tools. However, one major shortfall of these models are that they require several time and frequency domain tests using several different control loops to be conducted by an experienced controls engineer [40] [41] [42]. These time consuming tests can cause downtime on production machine tools, which results in lost profits for companies. In this thesis, three rapid identification models that do not require invasive testing on the machines will be presented:

1) Quasi-Static Model

2) SISO and Rigid Body Model

3) MIMO and Flexible Dynamics Model

Each model has its pros and cons. In [43], a motion control trajectory pre-filter theory was proposed which is extended in this thesis to estimate machine tool's quasi-static dynamics in a practical way using in process gathered data. This method will be discussed in Chapter 3. The quasi-static model has the fastest computation time, however it does not take into account the poles (natural response) or the friction of the

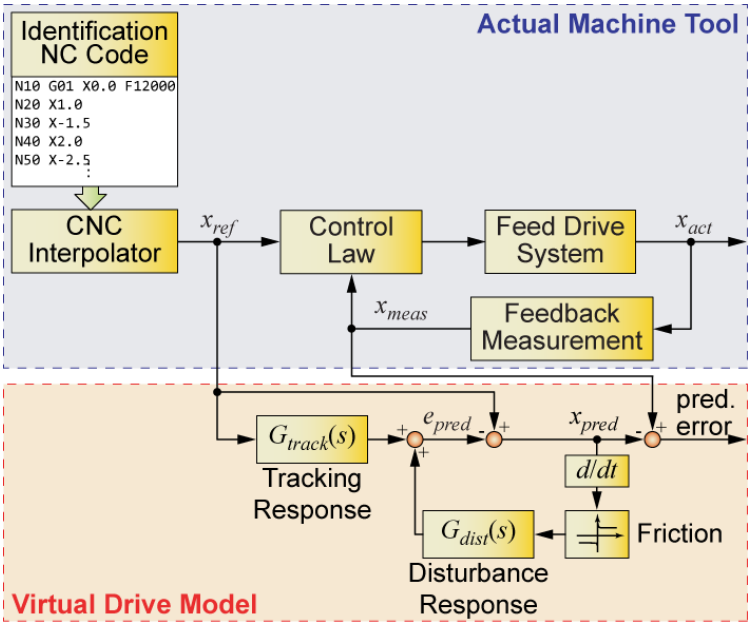


Figure 2-3 Overview of virtual drive model

system. The SISO model takes into account the machine's natural response, however it only considers rigid body dynamics. In previous literature, a rapid identification strategy was proposed, where a short G-code is executed on a machine tool, and the commanded position (input) and actual position (output) are collected using the motion capture feature that is available on most controllers. This data was then taken offline and used to reverse engineer the equivalent tracking and disturbance transfer functions. Several techniques have been used to identify the transfer functions, such as constrained least squares (LS) [3], genetic algorithm [44], and particle swarm optimization [28]. The constrained least squares method will be explained further in Section 4.2, and will be used to benchmark against the proposed model. There were two main shortcomings for this constrained LS method: 1) inaccurate or unstable models could be identified due to lack of persistent excitation of the machine tool, and 2) numerical derivatives of the encoder readings are

used in the regressor matrix, which can lead to noise issues on low resolution encoders. These issues are mitigated using the newer proposed SISO model in this thesis as seen in Figure 2-3 and discussed further in Chapter 4. The MIMO model can be used to identify the transfer functions between several inputs and several outputs and can account for vibratory dynamics, however it is more computationally intensive, and this method will be discussed in Chapter 5.

2.5 Conclusions

In this chapter, the importance of virtual CNCs to the manufacturing industry and to academia have been discussed. Several different modeling and identification techniques were discussed, including a variety of friction models. It is clear that an accurate model of the machine tool is required in order to accurately simulate processes in a virtual environment, however the more traditional ways of building these models are too time consuming and causes significant downtime on machines. Additionally, the previous commonly used constrained LS rapid identification method has shortcomings that must be solved. In the following chapters of the thesis, three new rapid identification techniques will be proposed, discussed and validated using experimental results.

Chapter 3
Quasi-Static Model for Rapid Identification

3.1 Introduction

In some situations, such as in trajectory tracking error prediction, virtual models do not need to take into account the drive's natural response (i.e. poles). For certain feed drives that have rigid body dominant dynamics, a simple and quick model can be used in trajectory tracking prediction. In this chapter, a practical model that predicts the instantaneous tracking error of a machine tool related to the time derivatives of the position command will be discussed.

3.2 Quasi-Static Model

The error transfer function, G_{er}, can be defined by Eqn. (3.1), where G_{track} is the tracking transfer function between the commanded and actual position of the machine tool.

$$G_{er} = 1 - G_{track} \tag{3.1}$$

The instantaneous positioning error, $G_{er}(s)$, can be approximated by a Taylor series expansion around zero frequency ($s = 0$) as seen in Eqn. (3.2) where $x_r(s)$ is the commanded position, $x(s)$ is the actual position, $e(s) = x_r(s) - x(s)$ is the tracking error of the machine tool, and $G'_{er}, G''_{er}, G'''_{er}$ are the first three derivatives of G_{er} with respect to s. These derivative terms will be finite at $s = 0$ because there are no integrators in G_{er}. In this equation, $G_{er}(0) = 0$ because G_{track} is designed to have zero steady-state error in response to position commands. A Butterworth filter is used to filter the derivative profiles to reduce the noise in the signals.

$$G_{er}(s) = \frac{e(s)}{x_r(s)} = \underbrace{G_{er}(0)}_{=0} + \underbrace{G'_{er}(0)}_{K_{vel}} s + \underbrace{\frac{1}{2!}G''_{er}(0)}_{K_{acc}} s^2 + \underbrace{\frac{1}{3!}G'''_{er}(0)}_{K_{jerk}} s^3 + \cdots \tag{3.2}$$

For this model, the low frequency dynamics of the machine tool will be estimated by $\hat{G}_{er}$, which only contains the first three derivative terms as seen in Eqn. (3.3).

$$\hat{G}_{er} = \frac{\hat{e}(s)}{x_r(s)} = K_{vel}s + K_{acc}s^2 + K_{jerk}s^3 \tag{3.3}$$

By rearranging Eqn. (3.3) to isolate the term $\hat{e}(s)$ and taking the inverse Laplace transform, the tracking error of the machine tool can be estimated using Eqn. (3.4).

$$L^{-1}\{\hat{e}(s)\} \Rightarrow \hat{e}(t) \cong K_{vel}\dot{x}_r(t) + K_{acc}\ddot{x}_r(t) + K_{jerk}\dddot{x}_r(t) \tag{3.4}$$

To build this simplified model for machine tool drives, the commanded and actual positions are collected from the CNC controller. The data is then taken offline, where the commanded position is differentiated and filtered to reduce the noise of the signal and the tracking error is calculated. Using the least squares (LS) formulation in Eqn. (3.5), K_{vel}, K_{acc}, and K_{jerk} can be estimated, where Y is the output vector containing tracking error measurements, θ is the model parameter vector, Φ is the regressor matrix containing experimentally gathered command signals, and N is the total number of samples collected.

$$\left.\begin{aligned} &\text{Minimize } f = \frac{1}{2}(Y - \Phi\theta)^T(Y - \Phi\theta) \\ &Y = [e_1\ e_2 \dots e_N]^T \\ &\theta = \left[K_{vel}\ K_{acc}\ K_{jerk}\right]^T \\ &\Phi = \begin{bmatrix} \dot{x}_{r,1} & \ddot{x}_{r,1} & \dddot{x}_{r,1} \\ \dot{x}_{r,2} & \ddot{x}_{r,2} & \dddot{x}_{r,2} \\ \vdots & \vdots & \vdots \\ \dot{x}_{r,N} & \ddot{x}_{r,N} & \dddot{x}_{r,N} \end{bmatrix} \end{aligned}\right\} \tag{3.5}$$
$$\theta = (\Phi^T\Phi)^{-1}\Phi^T Y$$

3.2.1 Feedrate Optimization

This quasi-static model can be used for practical feedrate optimization by process planners. As seen in Eqn. (3.4), the servo errors due to the velocity, acceleration, and jerk commands can be predicted for each axis. If a maximum servo error, e_{max}, is specified for the part or process, the kinematic limits imposed on the feedrate optimization for the axis velocity, acceleration and jerk can be modulated, as seen in Eqn. (3.6). By implementing different combinations of limits on the axes, the planner is able to determine what cycle time optimization can be achieved in a virtual environment.

$$\hat{e}(t) \leq |K_{vel}\dot{x}_{r,max}| + |K_{acc}\ddot{x}_{r,max}| + |K_{jerk}\dddot{x}_{r,max}| \leq e_{max} \tag{3.6}$$

3.3 Experimental Validation

Experimental validation of the quasi-static model has been carried out on a laser drilling machine from the Pratt and Whitney Canada (P&WC) manufacturing line in Longueil, Quebec. The servo data was collected using the software Sinucom from the Siemens 840D SL controller. The concept of the experimental identification process can be seen in Figure 3-1.

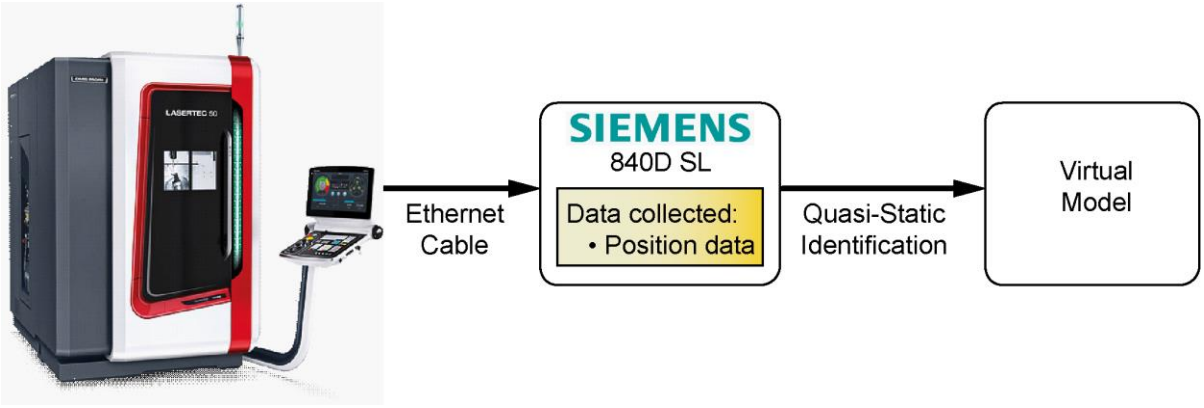


Figure 3-1 Experiment identification process for quasi-static model (representative picture of machine tool obtained from [45])

Data was collected for a gas turbine component at a sampling rate of 2 ms, as shown in Figure 3-2. Due to confidentiality reasons, the real machine tool, part dimensions, and process cycles times of P&WC related tests cannot be disclosed in this thesis.

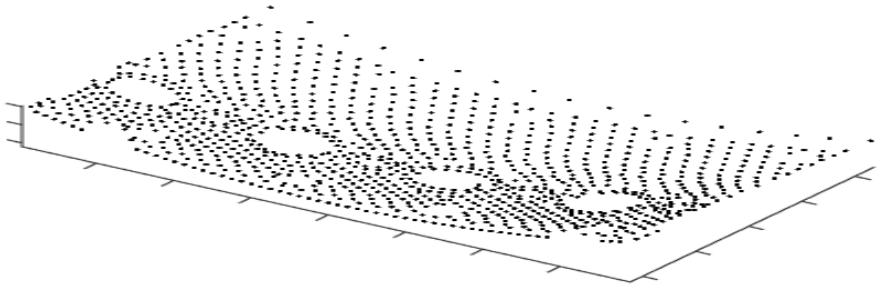


Figure 3-2 Hole locations for gas turbine engine combustion chamber panel

3.3.1 Results

The results of the identification and verification can be seen in Figure 3-3, where “Err. of Pred.” (or error of prediction) is the difference between the actual and predicted tracking error. The black lines show the commanded trajectory, the blue lines show the measured (actual) servo response from the encoders, the red lines show the data used for identifying the quasi-static model and the green lines show the data used for verification of the model. 10000 points of the collected data were used for identifying the model and 45000 points were used for model verification. The quasi-static identification method works well with commanded acceleration data that contains non-zero velocity, acceleration and jerk components, which is common with most CNC trajectories.

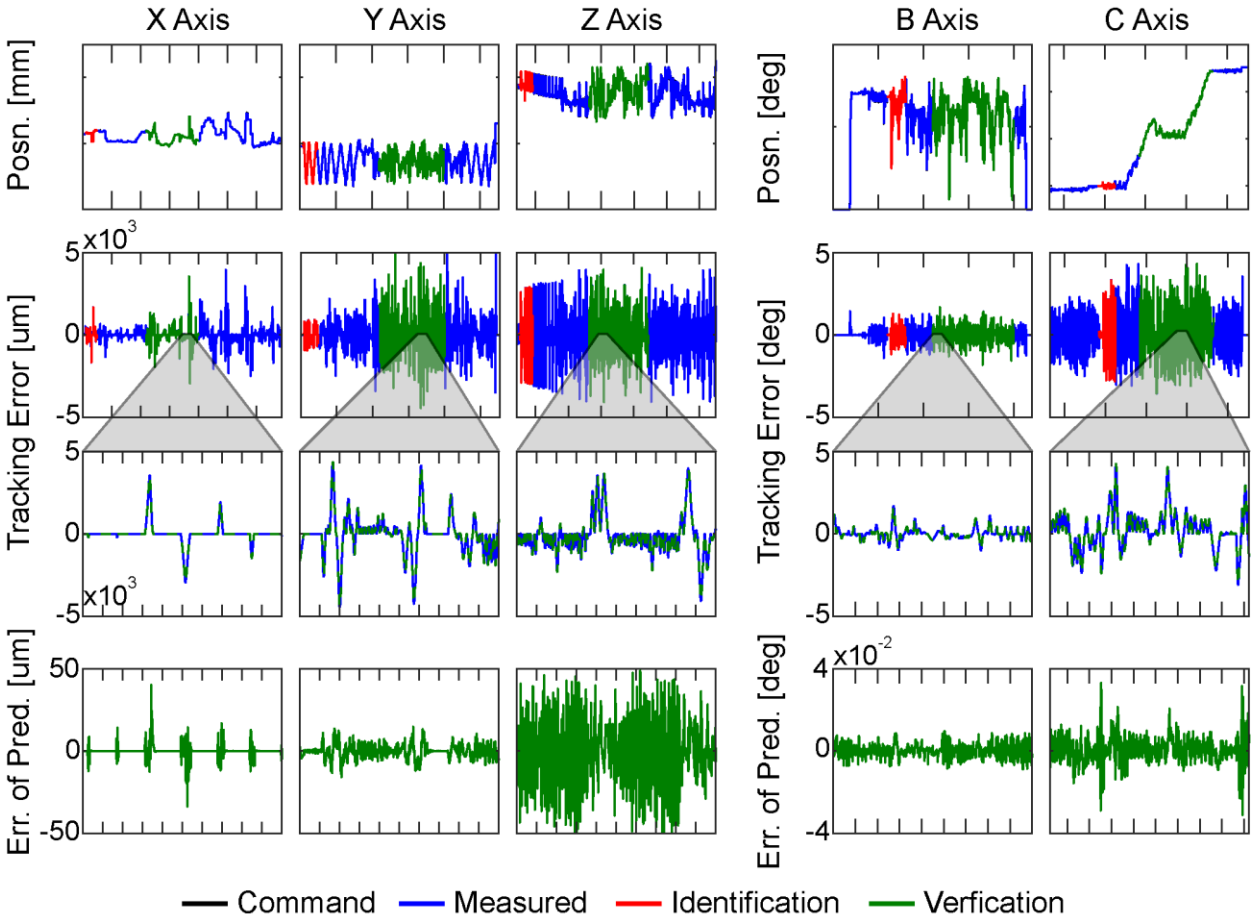


Figure 3-3 Results of quasi-static model for the laser drilling machine

The tracking error prediction results can be seen in Table 3-1. It can be seen from the figure and table that the quasi-static model can predict the machine tool's RMS tracking error between 2.62 to 11.91 µm of the actual RMS tracking error. Also the model is able to predict the maximum tracking error within 11.40 to 69.05 µm.

Table 3-1 Tracking error prediction for the laser drilling machine

Discrepancy in Servo Error Prediction	X Axis [µm]	Y Axis [µm]	Z Axis [µm]	B Axis [µdeg]	C Axis [µdeg]
Actual tracking error: RMS (MAX)	259.53 (3555.00)	861.77 (4915.50)	907.13 (4060.80)	320.68 (1689.40)	1039.10 (4302.40)
Predicted tracking error: RMS (MAX)	259.45 (3552.40)	861.81 (4921.30)	907.65 (4057.60)	320.51 (1689.20)	1038.50 (4308.10)
Prediction error: RMS (MAX)	2.62 (40.27)	3.44 (15.05)	11.91 (69.05)	2.83 (11.40)	6.41 (63.15)

3.4 GUI Development

A graphical user interface (GUI), as seen in Figure 3-4, was developed for the process planners and engineers at P&WC to be able to identify quasi-static machine models for both Siemens controlled and FANUC controlled machine tools.

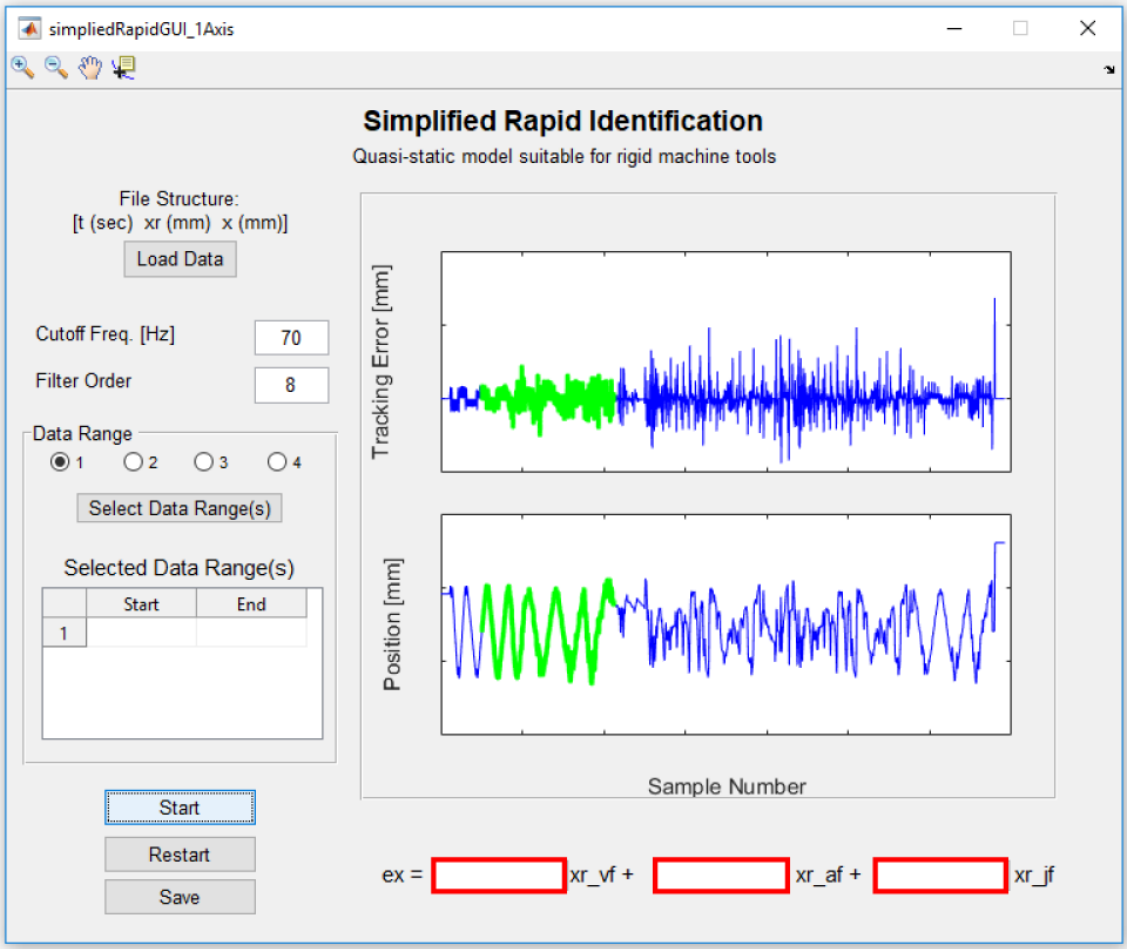


Figure 3-4 Graphical user interface for quasi-static model

This GUI allows users to import data collected directly from the different controller software, Sinucom (Siemens) and Servo Guide/Servo Viewer (FANUC), without any need for post-processing. The users are able to select a data range for identification, as well as the order and cutoff frequency of the Butterworth filter to be applied to the collected data. A model is then identified from the user specified parameters and used to simulate the response of the machine tool for the data set. Users can also vary the identified K_{vel},

K_{acc}, and K_{jerk} parameters by changing the red highlighted fields, as seen in Figure 3-4, to visualize the effect of the gains on the model.

3.5 Conclusions

In this chapter, a practical quasi-static model was presented. The model is able to determine the tracking errors contributed from the velocity, acceleration, and jerk commands of each axis of the machine tool from in-process gathered data. This method can be used by process planners to modify feedrate optimization axis limits, to determine which combination of limits results in the shortest cycle time during the virtual prototyping stage. The quasi-static model has been experimentally verified using data measured from a laser drilling machine. A GUI has also been created to aid engineers and technicians in the identification of a quasi-static model.

Chapter 4
SISO and Rigid Body Model for Rapid Identification

4.1 Introduction

In this chapter, a single input – single output (SISO) rigid body model will be discussed. This model takes into account the machine tool's natural response (i.e. poles) and includes a simplified friction model. In Chapter 3, the quasi-static model only considers the kinematics of the machine tool (i.e. how the velocity, acceleration and jerk affects the tracking error of a process). While the quasi-static model is useful for trajectory tracking prediction, a higher fidelity model is needed to build a virtual twin of machine tools for virtual CNCs and process planning. In this chapter, the proposed model is able to identify and capture key dynamics of the feed drives using in-process gathered data. This model improves on the rapid identification technique proposed in [3], which was able to simulate key dynamics of a large group of CNC drives, however there were some deficiencies in this method. The model that is discussed in this chapter only considers the low frequency response of the tools that originate from rigid body dynamics.

In Section 4.2, the previous constrained rapid identification model from [3] is described. The proposed pole search with least squares projection is explained in Section 4.3. Experimental results and discussion are presented in Section 4.4, followed by conclusions in Section 4.6. The work presented in this chapter was done in collaboration with Mr. Mustafa Hakan Turhan and Dr. Kaan Erkorkmaz, testing of the algorithm and experimental results were done by the author, while Mr. Turhan developed further theory and more advanced friction modelling [46] [47].

4.2 Full Least Squares Based Model

In this section, the rapid identification model proposed in [3] will be discussed in detail as it will be used for benchmarking at the end of this chapter. A large selection of CNC drives, which usually have a ball screw or direct feed drive, can be modelled using a third order transfer function. These drives can have a variety of feedback controllers such as P, PI, PD, PID and/or P-PI cascade control, and can also incorporate feedforward dynamics or friction compensation, as seen in Figure 4-1. This model makes three main assumptions:

1. The machine is modelled as a rigid body since the machine's structural resonances are avoided during the usage of the machine (Machine tools are designed to have structural resonances beyond the servo bandwidth).

2. The motor and amplifier are operating in their linear range (being run under the saturation limit).

3. The machine tool's nonlinearities have little influence on the motion (e.g. backlash).

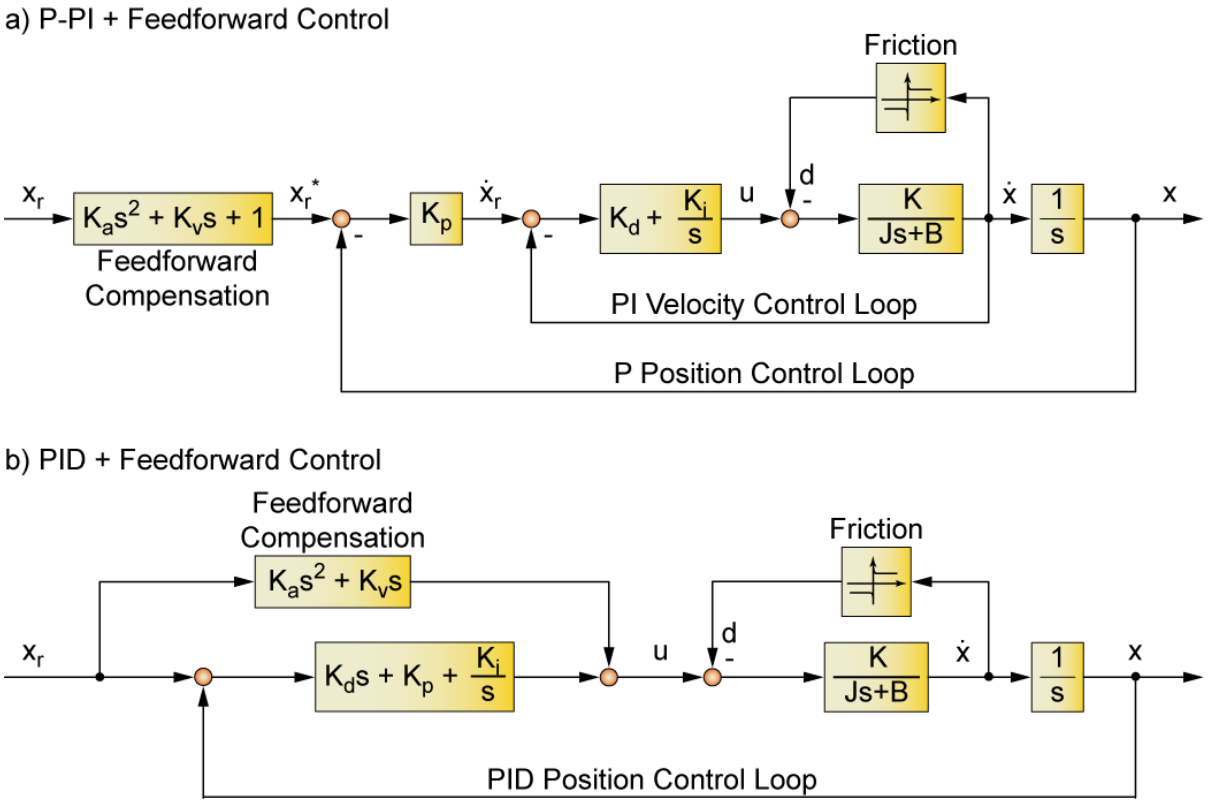


Figure 4-1 Frequently used control structures in CNC drives

From these assumptions, a closed loop transfer function can be modelled using Eqn. (4.1), where $x_r(s)$ is the commanded position, $x(s)$ is the actual position, $d(s)$ is the disturbance of the system, J is the open-loop inertia of the feed drive, and K is the combined motor torque, current amplifier, and ball screw gear transmission gain. G_{track} and G_{dist} are the equivalent command tracking transfer function and disturbance transfer function, respectively.

$$x(s) = \underbrace{\frac{b_0 s^2 + b_1 s + b_2 + a_3 \frac{1}{s}}{s^2 + a_1 s + a_2 + a_3 \frac{1}{s}}}_{G_{track}(s)} \cdot x_r(s) - \underbrace{\frac{K/J}{s^2 + a_1 s + a_2 + a_3 \frac{1}{s}}}_{G_{dist}(s)} \cdot d(s) \tag{4.1}$$

A Coulomb friction model is used to approximate the overall friction effect and is expressed using Eqn. (4.2). $PV()$ is a binary function which has a value of 1 when the axis velocity is positive and is otherwise 0. $NV()$ is a similar binary function which has a value of 1 when the axis velocity is negative and

is 0 otherwise. d^+ and d^- are the control signal equivalent values for the positive and negative directions of motion respectively for the Coulomb friction.

$$d(s) = d^+ \cdot PV(\dot{x}) + d^- \cdot NV(\dot{x}) \tag{4.2}$$

By combining Eqns. (4.1) and (4.2), the closed-loop dynamics can be written as Eqn. (4.3), where d_n^+ and d_n^- are the positive and negative normalized friction values respectively and can be calculated using Eqn. (4.4).

$$\left[s^2 + a_1 s + a_2 + a_3 \frac{1}{s}\right] \cdot x = \left[b_0 + b_1 s + b_2 + a_3 \frac{1}{s}\right] \cdot x_r - [PV(\dot{x}) \cdot d^+ + NV(\dot{x}) \cdot d^-] \tag{4.3}$$

$$d_n^{+/-} = \left(\frac{K}{J}\right) d^{+/-} \tag{4.4}$$

In discrete-time form, after normalizing the model parameters in Eqn. (4.3) with respect to a_2, the actual position of the machine tool at sample k can be predicted using Eqn. (4.5). In this equation, $\hat{x}_k = \hat{x}(kT_s)$, where T_s is the sample period of the collected data.

$$\hat{x}_k = \alpha_i e_{i,k} - \alpha_1 \dot{x}_k - \alpha_2 \ddot{x}_k + \beta_0 x_{r,k} + \beta_1 \dot{x}_{r,k} + \beta_2 \ddot{x}_{r,k} - PV(\dot{x}_k)\delta^+ - NV(\dot{x}_k)\delta^-$$

$$\left.\begin{aligned} \alpha_2 &= \frac{1}{a_2}, \quad \alpha_1 = \frac{a_1}{a_2}, \quad \alpha_i = \frac{a_3}{a_2}, \\ \beta_2 &= \frac{b_0}{a_2}, \quad \beta_1 = \frac{b_1}{a_2}, \quad \beta_3 = \frac{b_2}{a_2}, \\ \delta^+ &= \frac{d_n^+}{a_2}, \quad \delta^- = \frac{d_n^-}{a_2} \end{aligned}\right\} \tag{4.5}$$

The integrated tracking error, e_i, can be approximated using Eqn. (4.6):

$$e_{i,k} = T_s \sum_{m=1}^{k} (x_{r,m} - x_m) \tag{4.6}$$

The eight model parameters can be estimated using classical least squares (LS) by minimizing the problem in Eqn. (4.7), where N is the total number of samples, Y is the vector containing actual position measurements, θ is the parameter vector containing the unknown model parameters, and Φ is the regressor matrix containing the signals from the experimental measurements.

$$\left.\begin{aligned}
&\text{Minimize } f = \frac{1}{2}(Y-\Phi\theta)^T(Y-\Phi\theta)\\
&Y = [x_1 \quad x_2 \quad \dots \quad x_N]^T\\
&\theta = [\alpha_i \quad \alpha_1 \quad \alpha_2 \quad \beta_0 \quad \beta_1 \quad \beta_2 \quad \delta^+ \quad \delta^-]^T\\
&\Phi = \begin{bmatrix} e_{i,1} & -\dot{x}_1 & -\ddot{x}_1 & x_{r,1} & \dot{x}_{r,1} & \ddot{x}_{r,1} & -PV(\dot{x}_1) & -NV(\dot{x}_1)\\ e_{i,2} & -\dot{x}_2 & -\ddot{x}_2 & x_{r,2} & \dot{x}_{r,2} & \ddot{x}_{r,2} & -PV(\dot{x}_2) & -NV(\dot{x}_2)\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ e_{i,N} & -\dot{x}_N & -\ddot{x}_N & x_{r,N} & \dot{x}_{r,N} & \ddot{x}_{r,N} & -PV(\dot{x}_N) & -NV(\dot{x}_N) \end{bmatrix}
\end{aligned}\right\} \tag{4.7}$$

$$\theta = (\Phi^T\Phi)^{-1}\Phi^T Y$$

Using this method, a short numerical control (NC) code with varying distance linear interpolation movements must be executed on the machine tool to observe the performance over a range of different feedrates. Executing these back and forth trajectories with varying accelerations and velocities, the model is able to capture the Coulomb friction dynamics caused by the motor torque/force amplitude dependency fairly well. There were, however, two main short comings to this model:

1. The prediction accuracy of the identified model can decline or even lead to unstable models due to the lack of persistent excitation of the machine tool while collecting data. This occurs because CNC systems usually have commanded signals that are acceleration-continuous and can cause the identified parameters to not converge.

2. The regressor matrix in Eqn. (4.7) contains numerical derivatives of the measured encoder signals. If the machine tool has low resolution encoders or the significant digits is limited in the collected data, the differentiation could cause noise issues.

To avoid unstable models due to the lack of persistent excitation, bounds are imposed during the estimation process that ensure the identified model has pole locations away from the imaginary axis of the s-plane. The characteristic polynomial can be written using Eqn. (4.8), and the constraints imposed of the problem can be expressed using Eqn. (4.9).

$$s^3 + a_1 s^2 + a_2 s + a_3 = (s+p)(s^2 + 2\zeta\omega_n s + \omega_n^2) \tag{4.8}$$

$$\text{Constraints: } \begin{matrix} p \geq p_{min} & \omega_n \geq \omega_{n,min} \\ \zeta \geq \zeta_{min} & \zeta \leq \zeta_{max} \end{matrix} \tag{4.9}$$

Where $p_{min} > 0$, $\omega_{n,min} > 0$, and $\zeta_{max} > \zeta_{min} > 0$ to ensure stability of the system. It is also assumed that $\zeta_{max} > 1$.

In previous studies, it was found that in most cases the unconstrained estimation using Eqn. (4.7) would yield a model that also satisfied the imposed constraints in Eqn. (4.9). In this thesis, the full LS method was

implemented by solving the unconstrained eight parameter estimation from Eqn. (4.7) and the identified parameters were verified to fall within the constraints specified in Eqn. (4.9). This previous model, which will be referred to as the "Full LS Solution", will be used as a benchmark for the new proposed model, subsequently referred to as the "Pole Search with LS Projection" model.

4.3 Pole Search with LS Projection

The pole search with LS projection method can identify a model of the machine tool using in-process gathered data. This means that the data collection process no longer has to interrupt the production machine's operation. In Figure 4-2, the block diagram for the new proposed model can be seen.

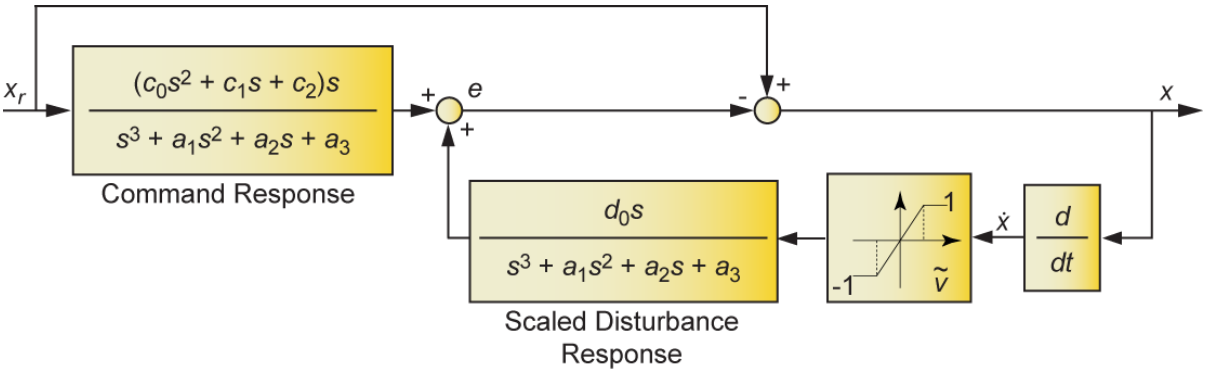


Figure 4-2 Closed-loop model for predicting tracking error and feed drive position

The proposed model predicts the instantaneous tracking error, $e(s)$ using Eqns. (4.10) and (4.11), rather than the instantaneous axis position, x, like the full LS solution discussed in Section 4.2.

$$e(s) = \underbrace{\frac{(c_0s^2 + c_1s + c_2)s}{s^3 + a_1s^2 + a_2s + a_3}}_{G_{e,xr(s)}} x_r(s) + \underbrace{\frac{(K/J)s}{s^3 + a_1s^2 + a_2s + a_3}}_{G_{e,d(s)}} d(s) \tag{4.10}$$

$$\left.\begin{aligned} c_0 &= 1 - b_0 \\ c_1 &= a_1 - b_1 \\ c_2 &= a_2 - b_2 \end{aligned}\right\} \tag{4.11}$$

$d(t)$ is the friction disturbance, and is estimated at small velocities, as being viscous using a Coulomb friction model as seen in Eqn. (4.12), where $\tilde{v}$ is the velocity transition band and d_0' is the normalized control Coulomb friction input for positive and negative motion.

$$d(t) = d_0' sat(\dot{x}/\tilde{v}) \tag{4.12}$$

By grouping together the d_0' and the gain of the disturbance response, K/J, a single gain for the scaled friction input between -1 and +1 can be calculated using Eqn. (4.13).

$$d_0 = d_0' \left(\frac{K}{J}\right) \tag{4.13}$$

Numerical differentiation of the measured encoder readings are noisy, even after filtering. Many solutions have been investigated to alleviate this problem. The first solution was to use a pure $sign()$ function, however, the noisy signals could lead to a false detection of sudden friction force transitions. Another solution that was explored for simulations was to use a block diagram like the one in Figure 4-2 to estimate the axis position and tracking errors. However, when this model was coupled with a feedback loop, this resulted in limit cycle type oscillations due to the discontinuous transitions around zero velocity. The last solution, which is implemented in the proposed method, is to use a saturation function. By using this function, the model acts as viscous friction around the small velocities while still being able to estimate the friction parameter and predict the tracking error due to the friction force.

Substituting Eqns. (4.12) and (4.13) into Eqn. (4.10), the expression in Eqn. (4.14) is obtained, where $P(s)$ is the characteristic polynomial. In this equation, the pole locations are chosen such that $P(s) = s^3 + a_1 s^2 + a_2 s + a_3 = (s+p)(s^2 + 2\zeta\omega_n s + \omega_n^2)$. The pole locations $p, w_n,$ and ζ are chosen from within a search space that the user specifies, to ensure that the model does not become unstable. In the proposed method, the numerators and lumped friction coefficients can be estimated by a LS sub-problem as seen in Eqn. (4.15).

$$\underbrace{(s^3 + a_1 s^2 + a_2 s + a_3)}_{P(s)} \cdot e = (c_0 s^2 + c_1 s + c_2) \cdot s \cdot x_r + s\left[d_0 sat\left(\frac{\dot{x}}{\tilde{v}}\right)\right] \tag{4.14}$$

$$\begin{aligned} e &= \frac{c_0 s^3}{P(s)} x_r + \frac{c_1 s^2}{P(s)} x_r + \frac{c_2 s}{P(s)} x_r + \frac{s}{P(s)}\left[d_0 sat\left(\frac{\dot{x}}{\tilde{v}}\right)\right] \\ &= \underbrace{\frac{s^3 x_r}{P(s)}}_{\dddot{x}_{rf}} c_0 + \underbrace{\frac{s^2 x_r}{P(s)}}_{\ddot{x}_{rf}} c_1 + \underbrace{\frac{s x_r}{P(s)}}_{\dot{x}_{rf}} c_2 + \underbrace{\frac{s}{P(s)} sat\left(\frac{\dot{x}}{\tilde{v}}\right)}_{d_{nf}} d_0 \end{aligned} \tag{4.15}$$

The actual axis velocity, $\dot{x}$, is estimated using the measured velocity, $\hat{\dot{x}}_f$, which is calculated by numerically differentiating and filtering the measured encoder position readings. A moving average filter is used to ensure that there is zero phase shift with respect to time. By using this substitution, the noise issues are diminished when determining the direction of the instantaneous friction force, specifically during small velocities.

$P(s)$ is a third order polynomial since the model only considers the rigid body dynamics and integral action in the controller. This means that as long as the commanded position, x_r, is continuous, the filtered derivatives of the command profile $\dot{x}_{rf}$, $\ddot{x}_{rf}$, and $\dddot{x}_{rf}$ are also continuous. The saturation function's output

is 0th degree continuous, but after passing it through $s/P(s)$, two degrees of continuity will be added. The resulting signal is the normalized friction force (between -1 and +1), d_{nf}, that has been differentiated once and filtered through the system's closed-loop characteristic polynomial.

The four model parameters can be estimated by minimizing the LS problem in Eqns. (4.16), where Y is the vector containing the tracking error measurements, θ_2 is the parameter vector containing the unknown model parameters, and Φ_2 is the regressor matrix containing the signals from the experimental measurements.

$$\left.\begin{aligned} Y &= [e_1\ e_2 \dots e_N]^T \\ \theta_2 &= [c_0\ c_1\ c_2\ d_0]^T \\ \Phi_2 &= \begin{bmatrix} \dddot{x}_{rf,1} & \ddot{x}_{rf,1} & \dot{x}_{rf,1} & d_{nf,1} \\ \dddot{x}_{rf,2} & \ddot{x}_{rf,2} & \dot{x}_{rf,2} & d_{nf,2} \\ \vdots & \vdots & \vdots & \vdots \\ \dddot{x}_{rf,N} & \ddot{x}_{rf,N} & \dot{x}_{rf,N} & d_{nf,N} \end{bmatrix} \end{aligned}\right\} \qquad (4.16)$$

$$\theta_2 = (\Phi_2^T \Phi_2)^{-1} \Phi_2^T Y$$

The step by step implementation of the proposed model is illustrated in Figure 4-3.

1. Characteristic polynomials are constructed using candidate poles chosen from within a pre-established stable search space.
2. Construct the regressors by filtering the experimental data through the characteristic polynomials to estimate the numerator terms.
3. Apply LS to solve for the numerator and friction parameters.
4. Each candidate model is evaluated by the objective function which penalizes the root mean square (RMS) of the prediction error, as seen in Eqn. (4.17), where N is the total number of samples, e_{actual} is the measured tracking error and e_{pred} is the predicted tracking error. The pole set, and the numerator and friction parameters with the lowest RMS value is chosen as the model. The global search function [48] from MATLAB's Global Optimization Toolbox was used to minimize the objective function. This search function was used as it provided the best and most robust parameter convergence.

$$f = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(e_{actual} - e_{pred}\right)^2} \qquad (4.17)$$

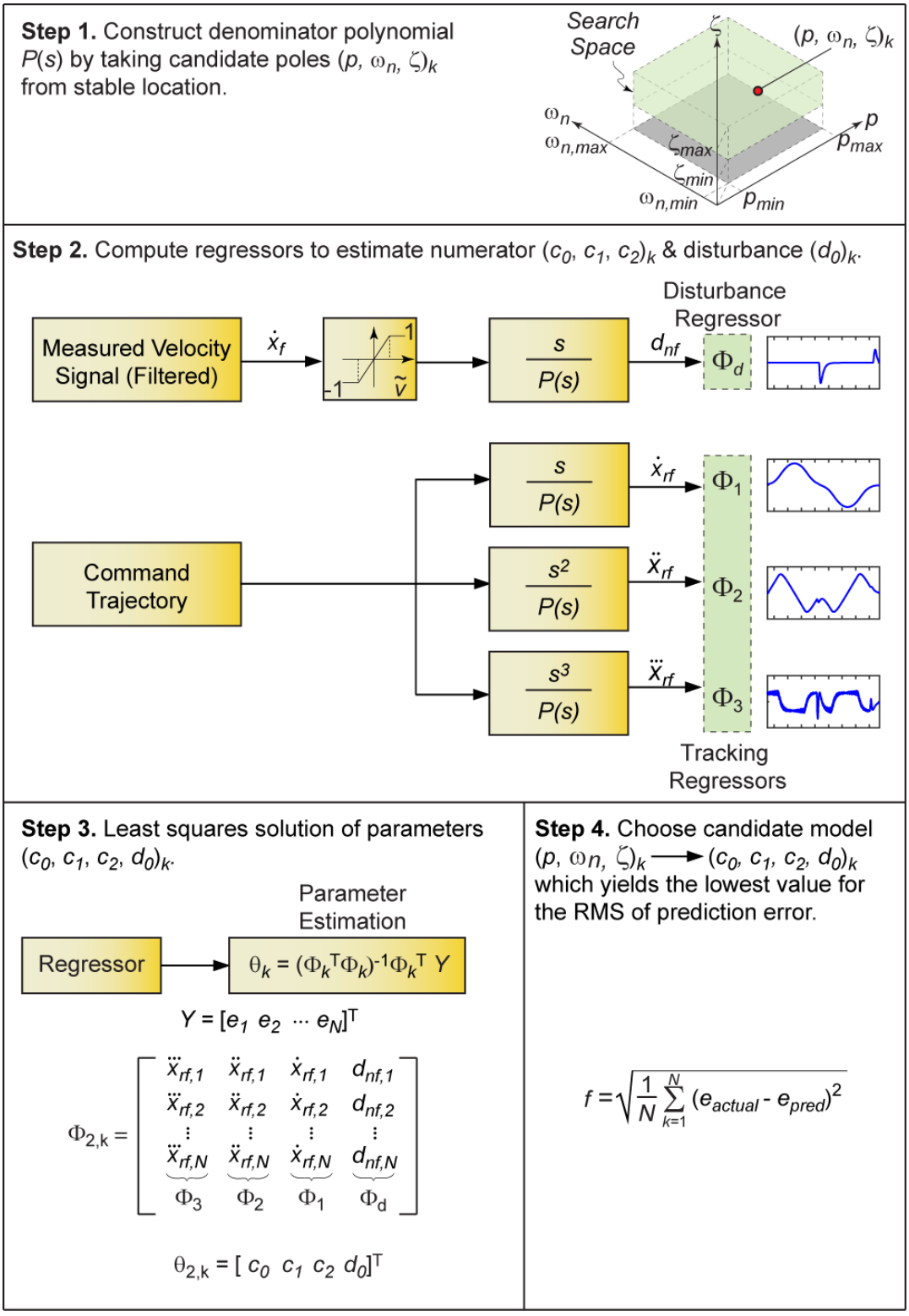


Figure 4-3 Proposed SISO procedure for identifying virtual feed drive dynamics

4.4 Experimental Validation

The Deckel Maho 80P hi-dyn five-axis machining center, seen in Figure 4-4, was used to validate the proposed method. The machine tool has a Heidenhain TNC 430N controller with a built-in scope function which was used to collect the commanded position and measured position of the linear encoder with 0.1 µm resolution. The scope function allows for 4096 samples to be collected at a sampling period of 0.6 ms.

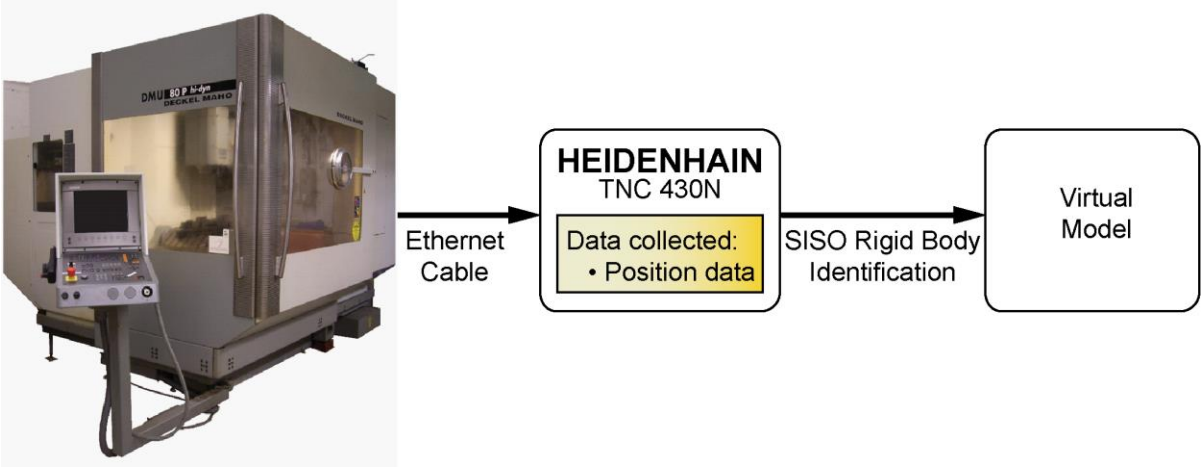


Figure 4-4 Experiment identification process for SISO model

Three different trajectories were captured on this machine tool: a diamond shaped toolpath with 20 mm edge length, a circle shaped toolpath with 20 mm radius, and a varying distance linear interpolation (pseudo-random) toolpath. The diamond toolpath is used for model identification, while the circle and linear interpolation toolpaths are used for model verification. The trajectory data range selected for fitting the model started when the machine was in motion and stopped a few seconds after the machine finished moving to capture the settling response.

4.4.1 Results

The bounds for the stable search space can be seen in Table 4-1, where 833.33 Hz corresponds to the Nyquist frequency of the experimental data. It should be noted that the tracking errors are relatively large on this experimental setup because the feedforward action on the CNC controller is turned off, which was verified by checking the TNC 430N controller parameters. Aggressive trajectories were tested on this machine tool, with tangential feedrate, acceleration and jerk values reaching up to 137 mm/s, 5764 mm/s^2, and 5.51×10^6 mm/s^3, respectively.

Table 4-1 SISO pole search bounds for X- and Y- axes of Deckel Maho 80P

Parameters	Min	Max
p [Hz]	1	833.33
ω [Hz]	1	833.33
ζ[]	0.2	2

The estimated parameters can be seen in Table 4-2. The x- and y- axes have relatively close natural frequencies, w_n, at 15.79 and 11.18 Hz respectively, as well as close damping ratios, ζ, at 1.12 and 1.22 respectively. The higher frequency x- and y- poles, p, at 88.55 and 209.74 Hz are not similar, however these poles do not have a large influence on the overall response, which indicates that the machine tool has 2nd order dominate dynamics. Since there is no pole-zero cancellation, it is evident that there is no feedforward control being used by the machine tool (which is true in this experimental case). During some of the LS projection cases, some of the pole set candidates lead to d_0 having a negative value, these instances were excluded from being potential models.

Table 4-2 SISO identified parameters for X- and Y- axes of Deckel Maho 80P

Parameters	**X Axis**		**Y Axis**	
p [Hz]	88.55		209.74	
ω [Hz]	15.79		11.18	
ζ[]	1.12		1.22	
c_0	0.01		1.04	
c_1	1.21E+03		2.34E+03	
c_2	6.89E+04		8.19E+04	
d_0	699.04		714.99	
Tracking TF Zeros	**Freq. [Hz]**	**Damp. [ ]**	**Freq. [Hz]**	**Damp. [ ]**
z_1	48.8	-0.8	3326.3	1.0
z_2	48.8	-0.8	33.3	-1.0
z_3	9.3	1.0	5.8	1.0

The full LS based model is used as a benchmark for the new pole search LS projection method. The measured and predicted axis positions and tracking errors for the diamond, circle and pseudo-random trajectories can be seen in Figure 4-5, Figure 4-6, and Figure 4-7, respectively. The contour error prediction results for the diamond and circle toolpaths can be seen in Figure 4-8. Summaries for the prediction of full LS and LS projection method can be found in Table 4-3.

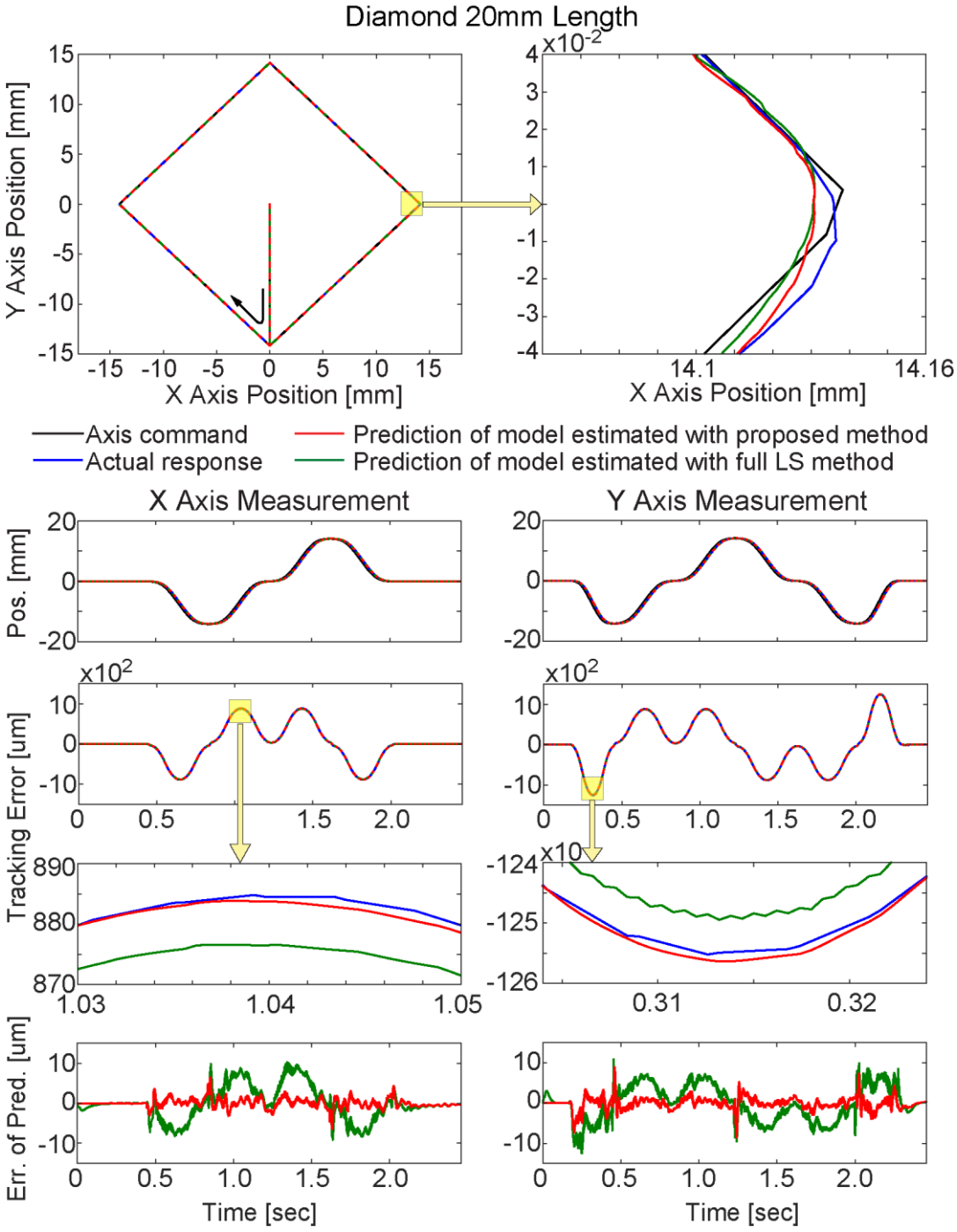


Figure 4-5 Actual and SISO predicted servo performance for a 20mm length diamond toolpath

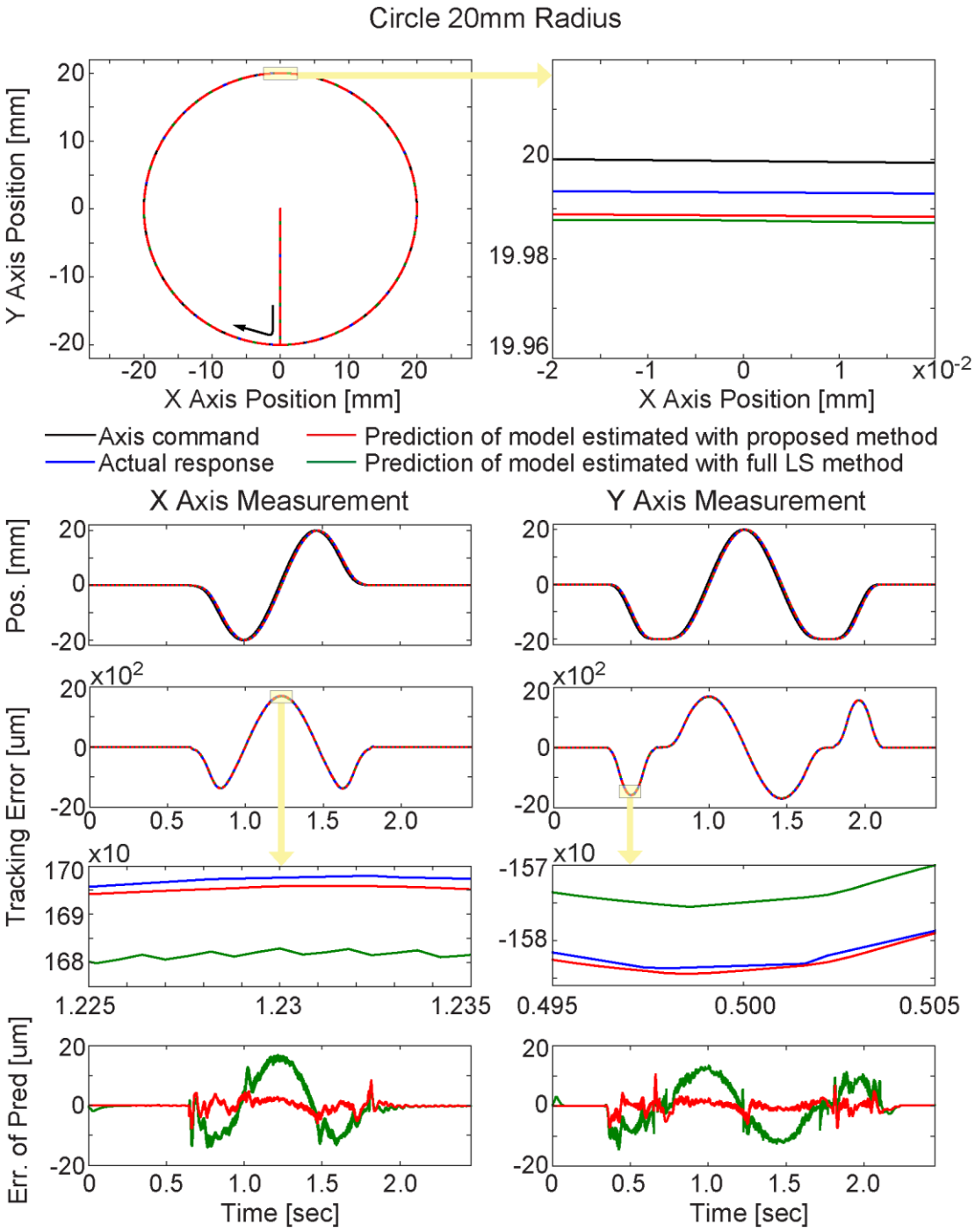


Figure 4-6 Actual and SISO predicted servo performance for a 20mm radius circle toolpath

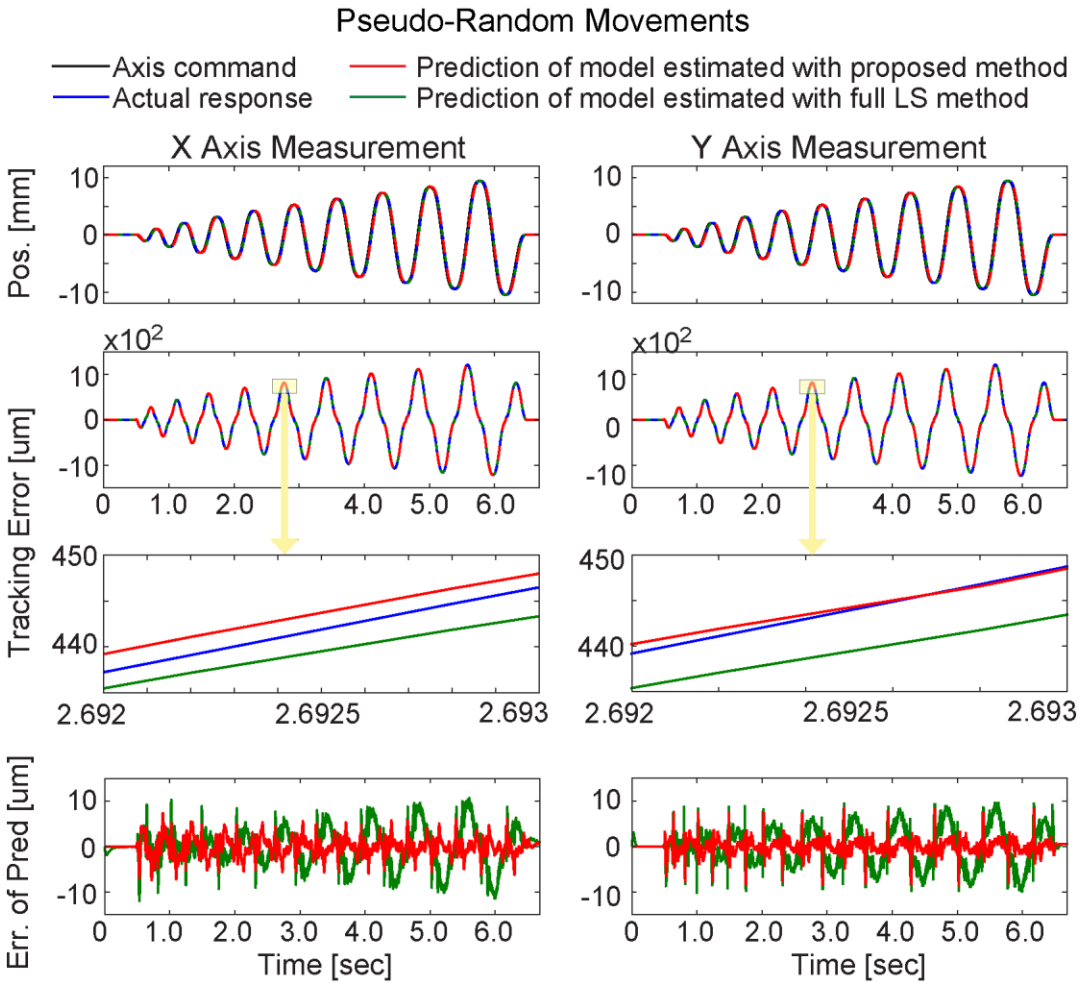


Figure 4-7 Actual and SISO predicted servo performance for a varying distance linear interpolation toolpath

Table 4-3 SISO tracking and contouring error prediction for Deckel Maho 80P

Discrepancy in Servo Error of Prediction	Diamond		Circle		Varying Dist. Linear Interpolation	
	Proposed Method	Full LS	Proposed Method	Full LS	Proposed Method	Full LS
X axis tracking error: RMS (MAX) [µm]	1.16 (6.45)	4.07 (10.29)	1.56 (8.45)	6.49 (16.97)	2.09 (7.41)	4.17 (11.96)
Y axis tracking error: RMS (MAX) [µm]	1.59 (9.09)	4.36 (12.65)	1.73 (10.68)	6.47 (14.62)	1.71 (8.70)	4.21 (10.35)
Contouring error: RMS (MAX) [µm]	0.90 (6.93)	1.16 (6.61)	1.39 (7.94)	2.14 (11.19)	-	-

It can be seen from Figure 4-5 and Table 4-3, that the proposed method is able to more consistently predict the tracking error when compared to the full LS method. Based on the maximum values, the discrepancy in tracking error prediction was reduced from 10.3 to 6.5 µm for the x axis, and 12.7 to 9.1 µm for the y axis. Therefore, there has been a 28-37% improvement in the prediction of the tracking error. While this improvment may not seem significant, it can be seen from the bottom panels of Figure 4-5, which shows the error of prediction, that the largest prediction errors occur during the velocity reversals where frictional effects are dominant. Although not a focus of this thesis, there is thorough research being conducted in the Precision Controls Laboratory to develop a more accurate friction model that will be coupled with the presented rapid identification methods. When comparing the RMS values of the prediction discrepancy, the x-axis prediction error was reduced from 4.1 to 1.2 µm while the y-axis error was decreased from 4.4 to 1.6 µm, improving the prediction by 64-71% for both axes.

Similar trends can be seen in Figure 4-6 and Table 4-3 for the circular trajectory. In this case, it can be seen that the prediction discrepancy due to the dynamic response of the sinusoidal inputs of both the x- and y- axes have more influence than the discrepancy due to the friction transitions. For the circle trajectory, the maximum and RMS prediction error has decreased by 27-50% and 73-76%, respectively. The same observations can be made for the varying distance linear interpolation toolpath, as seen in Figure 4-7 and Table 4-3. In this case, the maximim prediction discrepancy has been reduced by 16-38% and the RMS discrepancy has been reduced by 50-59%.

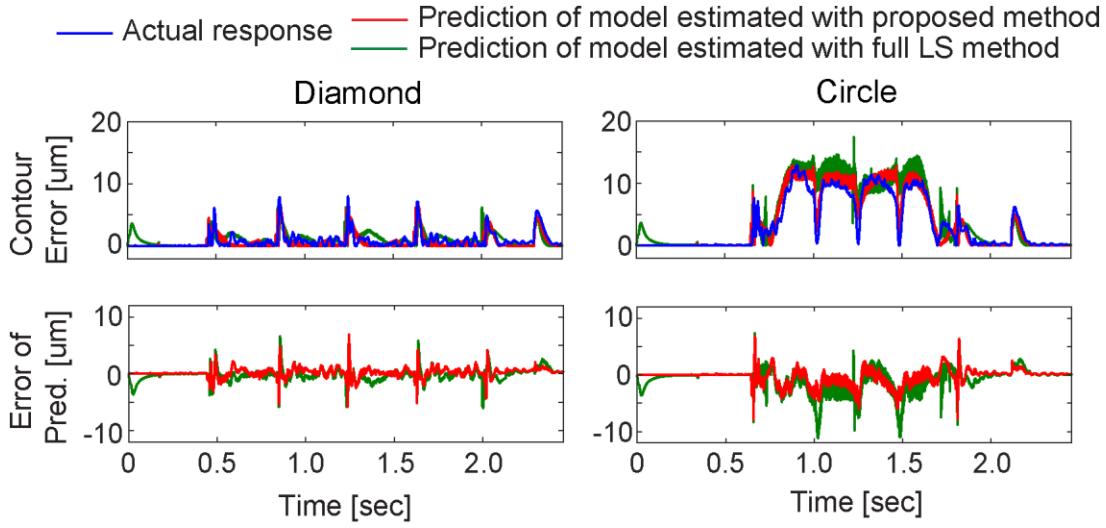


Figure 4-8 Contour errors for diamond and circle toolpaths on Deckel Maho 80P

From Figure 4-8 and Table 4-3, it can be observed that the proposed method's maximum prediction discrepancy for the contour error of the diamond trajectory is marginally worse than the full LS model.

Nonetheless, the RMS prediction error has improved by 22%. This is due to the well-matched x-y dynamics of the machine tool, which means that the contour errors are already very small to begin with. For the circle trajectory case, there is clear improvement for both maximum and RMS prediction discrepancy (35% and 29%, respectively).

In machine tools that have well matched dynamics, like the Deckel Maho 80P, improving the contour error prediction is more challenging than improving the tracking error prediction. This is because the instantaneous tool position has the tendency to follow the commanded tool path during linear and low curvature paths. However, increasing the accuracy of the tracking error prediction is still a key step to increasing the accuracy of the contour error prediction, especially in trajectories that have large velocity, acceleration and jerk magnitudes or have a high curvature.

4.5 GUI Development

A series of GUI modules have been developed for the SISO rapid identification method for process planners and engineers to use in industry. The first module is used to identify a model for a single axis from data collected from the machine tool. The next interface is used to verify the models identified in the first module from a second data set collect from the machine tool. The third and final GUI is then used by the process planners to simulate new trajectories using the identified model.

4.5.1 Rapid Identification GUI

In the first GUI, as seen in Figure 4-9, a SISO rigid body model can be fit from experimental data gathered from a FANUC or Siemens controller. There are six main steps to this module:

1. The figure seen in section 1 of the GUI is updated throughout the rapid identification process with the tracking error (mm) and the position (mm), shown in the top and bottom plot respectively.
2. In section 2, the user can load experimental data gathered through a .txt file. The file must contain the time (sec), measured position (mm), and tracking error (mm) to be used for training the model. The derivatives of the experimental data are calculated and filtered through an averaging filter of *n*th order, which is user defined in the GUI, to reduce the encoder quantization.
3. The user is able to select the upper and lower bounds for optimization in this step. This search space table will automatically be populated with suggested values once the experimental data is loaded into the GUI, however it is also editable if the user wishes to change the values.

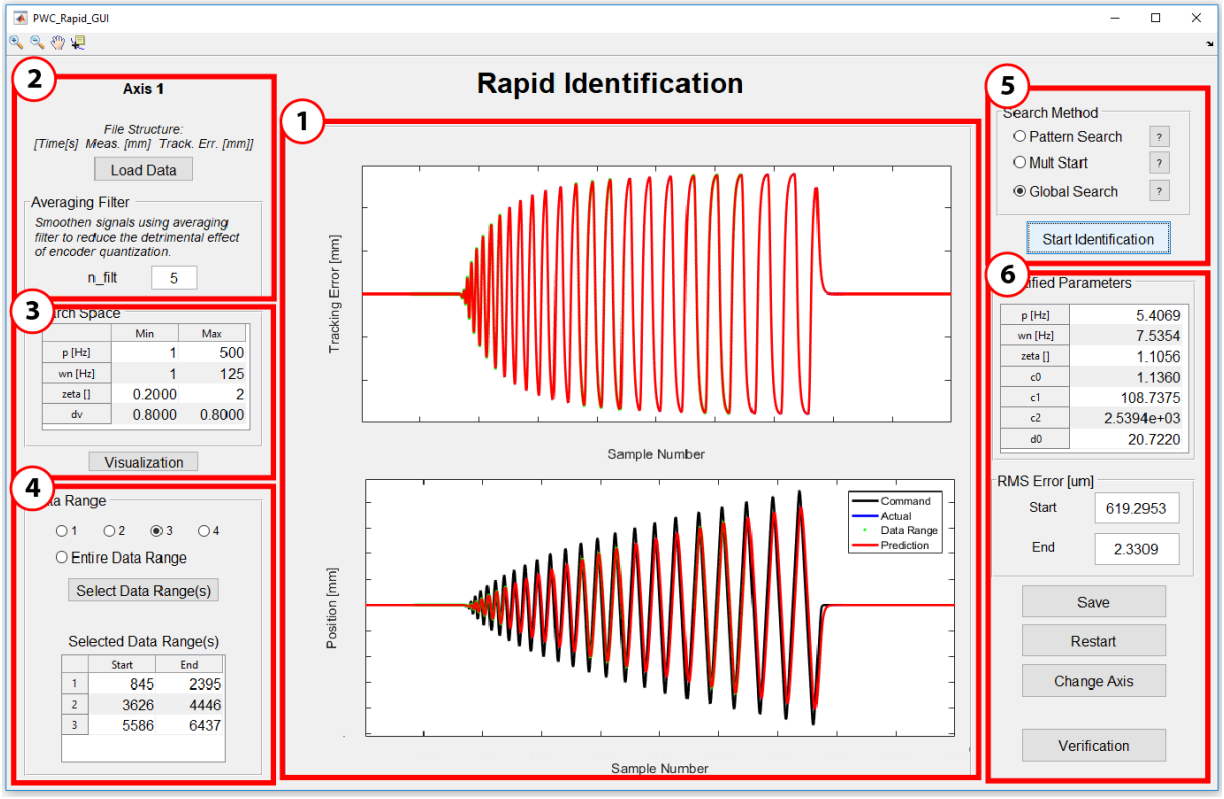


Figure 4-9 SISO rapid identification GUI

4. In section 4, the user can select up to four data ranges to use to train the rapid identification model. After the user selects the number of data ranges they wish to include in the identification, cross hairs will appear in the GUI which are used to select the start and endpoints of the data ranges. The users can also further modify and update these data ranges in the "Selected Data Range(s)" table.

5. In the fifth section, the user can choose between three different optimization methods that is included in MATLAB: 1) pattern search [49], 2) multi search [50], or 3) global search [48]. After the user selects which optimization method to use, the rapid identification process will begin.

6. In the final section, the identified parameters will be shown in the "Identified Parameters" table along with the initial and final root mean squared error (RMS) once the optimization is completed. The identified parameters can be saved into a .txt file for future use.

4.5.2 Verification GUI

The verification module, as seen in Figure 4-10, will be used to verify the model identified in the rapid identification module. In this module, trajectories that have movements in two axes will be required, to determine the contouring accuracy of the models. Note that A1 = axis 1 and A2 = axis 2.

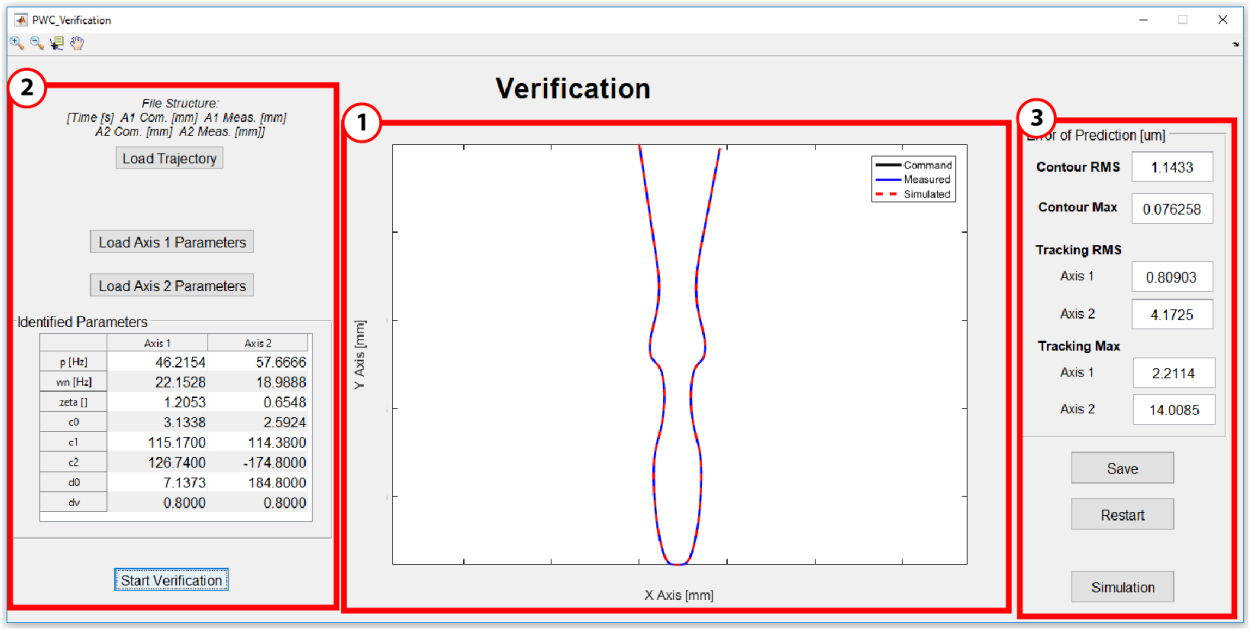


Figure 4-10 SISO verification GUI

1. The main figure in section 1 will be updated during the verification process with position (mm) data.
2. The user can select a .txt file that contains the experimental data that will be used for verification. The file structure should be as follows: time (sec), A1 measured position (mm), A1 tracking error (mm), A2 measured position (mm), and A2 tracking error (mm). Next the user must load in the identified parameters, saved from the first module, for axis 1 and axis 2. After all of the required files have been selected, the verification process can be started.
3. Once the verification has finished, the RMS and maximum contour error and tracking error will be populated in the right hand side of the GUI. A popup window will appear to show the position, tracking, and contour error for the commanded, measured and simulated profiles.

4.5.3 Simulation GUI

In the last module, as seen in Figure 4-11, the user can simulate a new process on the virtual machine tool given a trajectory. This module has 3 main parts:

1. The main figure in this section will be populated throughout the simulation process with the commanded and simulated trajectory.
2. In this step, the user imports an inverse time NC code [51] from the “Feedrate Optimization Module” developed in the lab [52]. After importing the trajectory, the user must import axis parameters for the

two axes that will be simulated, which have been identified and verified from the two previous modules.

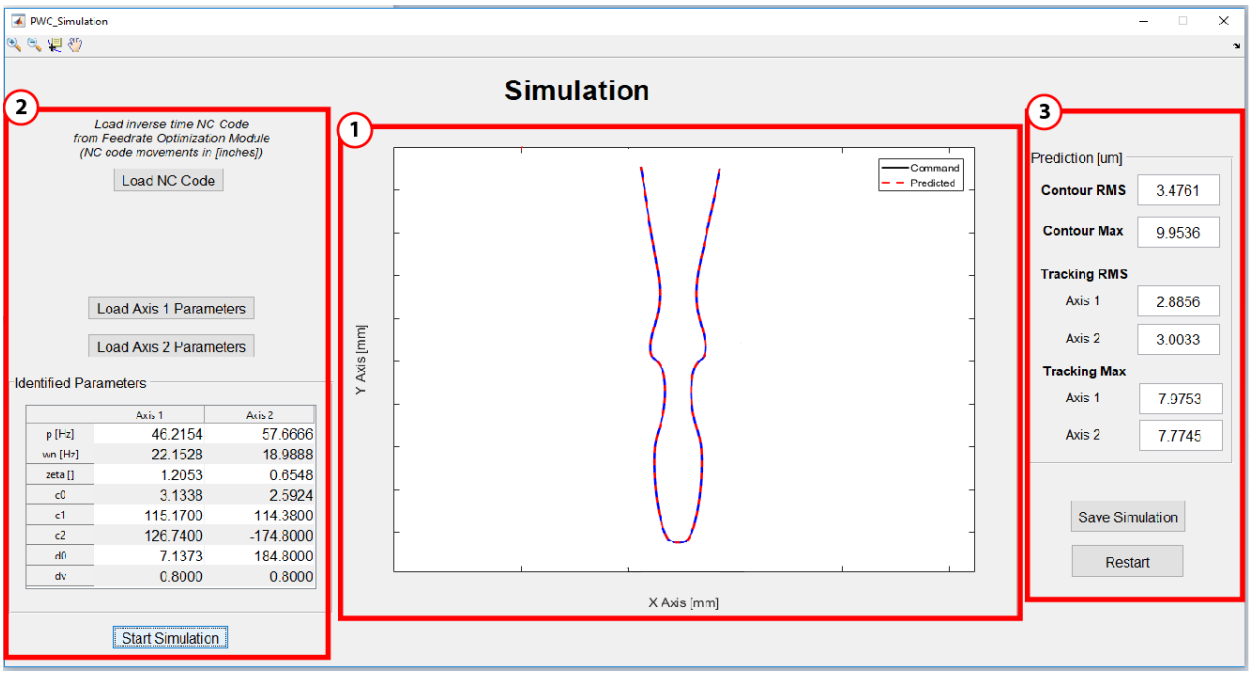


Figure 4-11 SISO simulation GUI

3. After the simulation has been completed, the RMS and maximum contour error and tracking error will be populated in the right hand side of the GUI. The results of this simulation can be saved for future plotting. A popup window will appear that displays the position, tracking and contour error for the commanded and simulated profiles.

4.6 Conclusions

In this chapter an improved single input – single output rigid body rapid identification model is presented. There are two main improvements from the previously researched full LS method, the first is improved parameter convergence characteristics and the second is the use of commanded signal derivatives in the regressor matrix. The new method was validated experimentally on a Deckel Maho five-axis machine tool. From the three different trajectory cases collected from the feed drive, it is observed that the measured servo errors can be predicted with an improvement of 16-50% when comparing the magnitudes of the tracking errors. The model is also able to predict the contouring error with 22-35% improvement in RMS error. However, there is still room for improvement, specifically with regards to the friction model that is important when modelling the dynamics during velocity reversal portions of the trajectory

Chapter 5
MIMO and Flexible Dynamic Model for Rapid Identification

5.1 Introduction

In newer machine tools, the performance of the feed drives are evolving to become more productive while still meeting part and process accuracy requirements. These developments in machine tool technology indicate that the digital twins of these tools should also become more advanced to include important phenomena such as elastic and vibratory dynamics, multi input – multi output (MIMO) coupling, etc. In this chapter, a MIMO model that builds upon the SISO model introduced in Chapter 4 is presented. In Section 5.2, the new proposed model will be outlined in detail, and in Section 5.3 experimental results and discussion will be presented.

5.2 Scaled MIMO LTI System

In the new model, as seen in Figure 5-1, the nonlinear (friction) dynamics are separated from the linear time-invariant (LTI) dynamics, $\boldsymbol{P}$, of the machine tool. The input and output signals from the machine tool are scaled to ensure that the MIMO transfer functions are estimated without any bias. $\boldsymbol{w}(t)$ are the scaled inputs, which include the reference signals $\boldsymbol{w_r}(t)$ (such as position, velocity, and/or acceleration commands), and disturbance signals $\boldsymbol{w_d}(t)$ (such as friction and process forces). $\boldsymbol{z}(t)$ are the scaled outputs such as position, velocity, acceleration, control signal, motor current, servo regulation error or any other output that can be measured and related to the inputs.

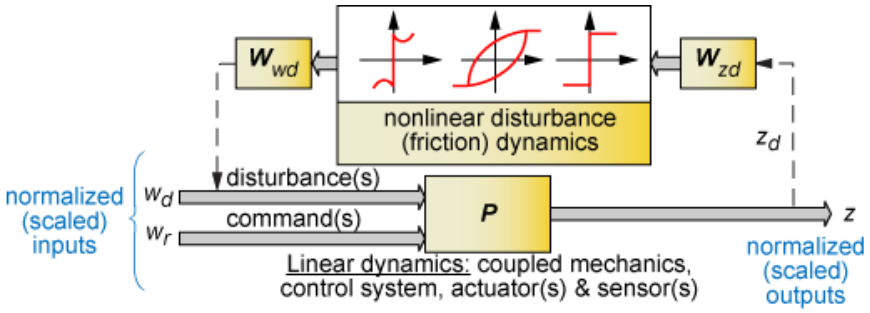


Figure 5-1 MIMO LTI feed drive model coupled with nonlinear dynamics

A more detailed view of the generalized dynamics can be seen in Figure 5-2. The new model is able to capture the mechanics of the motion system, sensor and actuator dynamics ($\boldsymbol{M}$), as well as dynamics from the controls (such as the feedback control ($\boldsymbol{K}$), feedforward control ($\boldsymbol{F}$) and trajectory pre-filter ($\boldsymbol{T}$)). The feedback and feedforward dynamics are typically unaffected during the execution of G01 (linear

interpolation), G02 (clockwise circular interpolation), and G03 (counter-clockwise circular interpolation) commands on machine tools. This allows the LTI assumption for the model to remain valid.

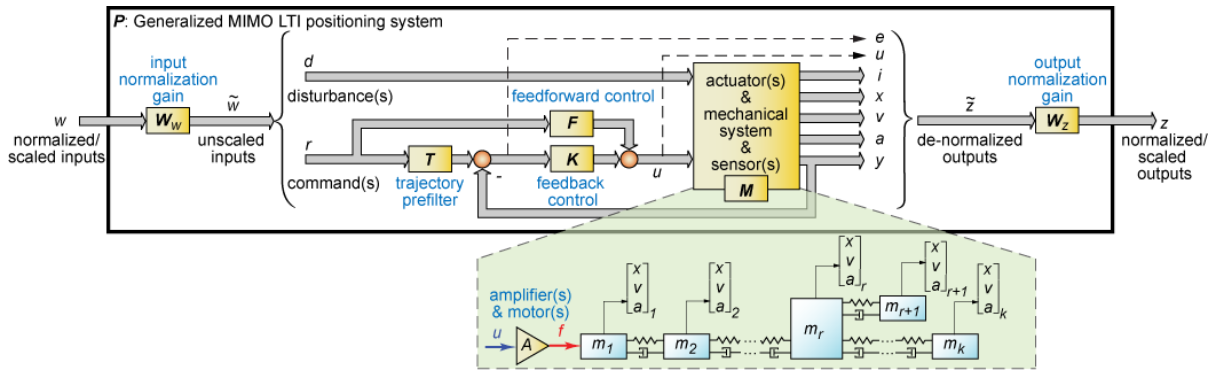


Figure 5-2 Overview of the dynamics of various feed drive system components captured by the scaled MIMO LTI model

Some possible feed drive models that can be identified using the proposed MIMO LTI rapid identification model can be seen in Figure 5-3. From this figure, it can be seen that the model is able to identify both coupled and decoupled models for a variety of both inputs and outputs.

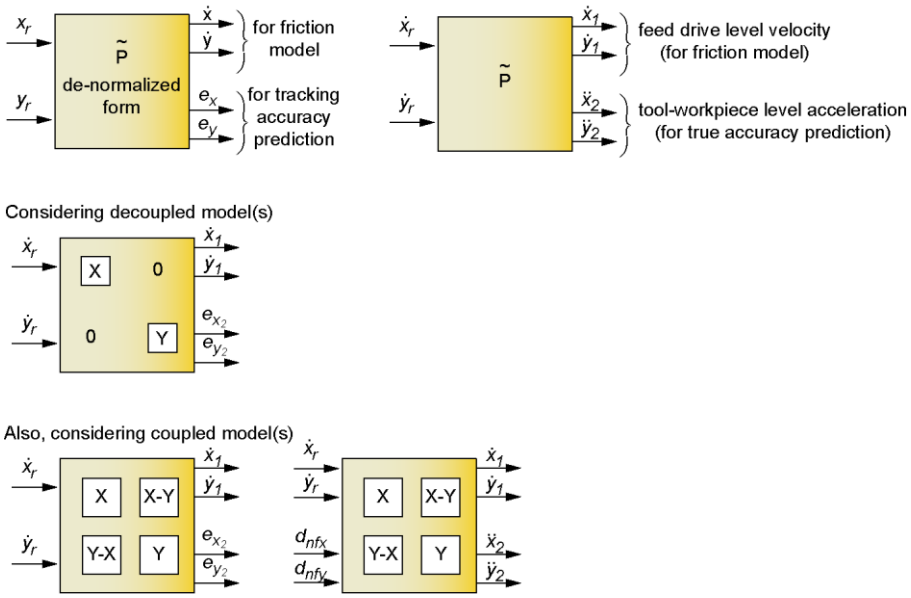


Figure 5-3 Possible feed drive model structures

The scaled and unscaled dynamics of the MIMO system can be expressed using Eqn. (5.1), where $\tilde{\boldsymbol{P}}$ is the unscaled dynamics, and $\boldsymbol{W}_w$ and $\boldsymbol{W}_z$ are the input and output scaling matrices, respectively. Like in Chapter 4, it is assumed that the poles of the system are stable, as the pole locations are subject to constraints

to ensure model stability, however complex poles will also be considered in this model. These poles can come from mechanical flexibilities (which lead to vibratory responses), trajectory prefiltering, and actuator/sensor dynamics. The denominator (characteristic) polynomial, $A(s)$ as seen in Eqn. (5.2), contains a specified combination of the total number of real poles, n_r, and the total number of complex conjugate poles, n_c, which make the polynomial nth order. p_k represents the real poles, and ω_k and ζ_k correspond to the natural frequency and the damping ratios of the complex poles, respectively.

$$\left.\begin{aligned} \boldsymbol{z}(s) &= \boldsymbol{P}(s)\cdot\boldsymbol{w}(s) && \text{Scaled dynamics} \\ \tilde{\boldsymbol{z}}(s) &= \tilde{\boldsymbol{P}}(s)\cdot\tilde{\boldsymbol{w}}(s) = \boldsymbol{W}_z\boldsymbol{P}(s)\boldsymbol{W}_w\cdot\tilde{\boldsymbol{w}}(s) && \text{Unscaled dynamics} \end{aligned}\right\} \tag{5.1}$$

$$\begin{aligned} A(s) &= A_r(s)\cdot A_c(s) = s^n + a_1 s^{n-1} + a_2 s^{n-2} + \cdots + a_{n-1}s + a_n \\ A_r(s) &= \prod_{k=1}^{n_r}(s - p_k), \qquad A_c(s) = \prod_{k=1}^{n_c}\left(s^2 + 2\zeta_k\omega_k s + \omega_k^2\right) \end{aligned} \tag{5.2}$$

The generalized plant $\boldsymbol{P}(s)$ has N_i inputs and N_o outputs. The concurrent application of inputs $k_i = 1, 2, \ldots, N_i$ results in the response of a single output k_o seen in Eqn. (5.3). Every transfer function that relates the output, k_o, to the input, k_i, each have their own distinct numerator terms $b_0, b_1, \ldots, b_n$. The denominator polynomial is assumed to be common for all input/output (i/o) transfer functions because they correspond to the same LTI system.

$$z_{k_o}(s) = \sum_{k_i=1}^{N_i}\left[\frac{b_0^{k_o,k_i}s^n + b_1^{k_o,k_i}s^{n-1} + \cdots + b_n^{k_o,k_i}s^0}{s^n + a_1 s^{n-1} + \cdots + a_n}\right] w_{k_i}(s) \tag{5.3}$$

Rearranging Eqn. (5.3), Eqn. (5.4) can be derived below. It can be seen that the numerator terms are linearly arranged unknowns and they are the participation factors of the input, k_i, on the output, k_o. Each line in the equation below corresponds to the response of a single input for $k_i = 1, 2, \ldots, N_i$.

$$\begin{aligned} z_{k_o}(s) = {} & \underbrace{\frac{s^n}{A(s)}w_1(s)}_{\varphi_0^1}\cdot b_0^{k_o,1} + \underbrace{\frac{s^{n-1}}{A(s)}w_1(s)}_{\varphi_1^1}\cdot b_1^{k_o,1} + \cdots + \underbrace{\frac{s^0}{A(s)}w_1(s)}_{\varphi_n^1}\cdot b_n^{k_o,1} \\ & + \underbrace{\frac{s^n}{A(s)}w_2(s)}_{\varphi_0^2}\cdot b_0^{k_o,2} + \underbrace{\frac{s^{n-1}}{A(s)}w_2(s)}_{\varphi_1^2}\cdot b_1^{k_o,2} + \cdots + \underbrace{\frac{s^0}{A(s)}w_2(s)}_{\varphi_n^2}\cdot b_n^{k_o,2} \\ & + \cdots \\ & + \underbrace{\frac{s^n}{A(s)}w_{N_i}(s)}_{\varphi_0^{N_i}}\cdot b_0^{k_o,N_i} + \underbrace{\frac{s^{n-1}}{A(s)}w_{N_i}(s)}_{\varphi_1^{N_i}}\cdot b_1^{k_o,N_i} + \cdots + \underbrace{\frac{s^0}{A(s)}w_{N_i}(s)}_{\varphi_n^{N_i}}\cdot b_n^{k_o,N_i} \end{aligned} \tag{5.4}$$

If $A(s)$ is known or a candidate model has been defined, a least squares (LS) parameter estimation can be used to calculate the optimal solution for the numerator parameters. The regressors, $\varphi_n^{k_i}$, can be computed

through simulations as shown in Figure 5-4. The predetermined characteristic polynomial $A(s)$ is used to form the single input – multi output (SIMO) elemental transfer function $\boldsymbol{G}_0(s)$. If one or more of the pole locations are known, these poles can be specified and locked to be excluded from the optimization process. For computational stability, the regressors are calculated using discrete-time state-space models ($\boldsymbol{G}_0 \rightarrow \boldsymbol{G}_{0,ssd}$).

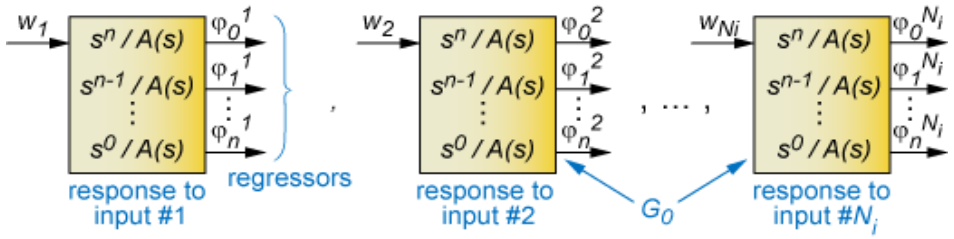


Figure 5-4 Estimation of regressors for solving the numerator polynomials

5.2.1 Construction of Regressors via Discrete-Time State-Space Model

$\boldsymbol{G}_0$ is the SIMO continuous transfer function which is transformed to the state space model $\boldsymbol{G}_{0,ssd}$ and discretized using the Tustin (bilinear) approximation. $\boldsymbol{G}_{0,ssd}$ has only $n = n_r + 2n_c$ states, this means that only $(n+1)$ regressors must be calculated for each LS problem. This saves a significant amount of computational load over using $n+1$ transfer functions, each of an n^{th} order. The $ss()$ command in MATLAB is used to convert the transfer function into a controllable canonical form (CCF) state space model using Eqn. (5.5) [53]. The $ss()$ function is similar to the $tf2ss()$ function in MATLAB, however the $balance()$ function is used to rescale the state vector, which improves the eigenvalue accuracy in subsequent calculations [53] [54].

$$H(s) = \frac{E(s)}{F(s)} = \frac{e_1 s^{g-1} + \cdots + e_{g-1}s + e_g}{f_1 s^{j-1} + \cdots + f_{j-1}s + f_j} = C(sI - E)^{-1}F + D$$
$$\dot{x} = Ex + Fu \qquad (5.5)$$
$$y = Cx + Du \text{ [55]}$$

The Tustin approximation was used since it allows for a system with a full order numerator to be discretized ($D \neq 0$ type of state space system) and has the best match in the frequency domain between the continuous and discretized models [56]. Since the transformations are dependent on the state space matrices and time delays in the system, the states are not preserved using the Tustin approximation which may lead to deviations in the true pole locations. However, the LS problem still finds a best fit model based on the generated regressor profile, which guarantees the inclusion of the dynamics that produce the regressors and

the corresponding responses in the model. Using Eqn. (5.6), the s and z domain transfer functions can be approximated and the continuous transfer functions can be discretized [56], as seen in Eqn. (5.7).

$$z = e^{sT_s} \approx \frac{1 + sT_s/2}{1 - sT_s/2} \tag{5.6}$$

$$H_d(z) = H(s'), \qquad s' = \frac{2}{T_s} \cdot \frac{z-1}{z+1} \tag{5.7}$$

5.2.2 Scaling of the Regressors in Computing the Numerator Term

Regressors that are computed using the original discrete-time state-space system, $\varphi_o^{k_i}, \dots, \varphi_n^{k_i}$, will be very different in magnitude. This is due to the signals being time derivatives of each other, ($\varphi_n^{k_i} = e^{at}$, $\varphi_o^{k_i} = a^n e^{at}$), for example if $a = 10$ and $n = 5$, then a^n will result in 10^5 which is 100000 times larger than the initial signal. In order to counteract this phenomenon, the weighting factor $r_l^{k_i}$ is calculated to normalize the original regressors between -1 and $+1$. The weighting factor can be calculated using Eqn. (5.8), where l is the order, k_i is the input number, k is the sample number, and φ is the original regressor. From this equation, the scaled regressor, $\bar{\varphi}$, can be calculated using Eqn. (5.9).

$$r_l^{ki} = \frac{1}{\left\|\varphi_l^{ki}\right\|_\infty}$$
$$\left\|\varphi_l^{ki}\right\|_\infty = \max\{\left|\varphi_l^{ki}(k)\right| : k = 1, 2, \dots, n\} \tag{5.8}$$

$$\bar{\varphi}_l^{ki} = r_l^{ki} \cdot \varphi_l^{ki} \tag{5.9}$$

After scaling the regressor matrix, using a given data set $t = 1, \dots, N_t$, where N_t is the total number of samples for output number k_o, the scaled numerator term(s), $\bar{b}^{ko}$, can be estimated using Eqn. (5.10) with a LS approximation. A visual representation of how the numerator is structured can be seen in Figure 5-5.

$$\underbrace{\begin{bmatrix} z_{k_o}(1) \\ \vdots \\ z_{k_o}(N_t) \end{bmatrix}}_{z_{k_o}\text{: measured scaled output}} = \underbrace{\begin{bmatrix} \overbrace{\bar{\varphi}_0^1(1) & \cdots & \bar{\varphi}_n^1(1)}^{\text{scaled regressors in response to input: \#1}} & \cdots & \overbrace{\bar{\varphi}_0^{N_i}(1) & \cdots & \bar{\varphi}_n^{N_i}(1)}^{\text{scaled regressors in response to input: \#}N_i} \\ \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ \bar{\varphi}_0^1(N_t) & \cdots & \bar{\varphi}_n^1(N_t) & \cdots & \bar{\varphi}_0^{N_i}(N_t) & \cdots & \bar{\varphi}_n^{N_i}(N_t) \end{bmatrix}}_{\boldsymbol{\Phi}\text{: regressor matrix}} \underbrace{\begin{bmatrix} \bar{b}_0^{k_o,1} \\ \vdots \\ \bar{b}_n^{k_o,1} \\ \vdots \\ \bar{b}_0^{k_o,N_i} \\ \vdots \\ \bar{b}_n^{k_o,N_i} \end{bmatrix}}_{\bar{b}^{ko}\text{: scaled numerator}} + \underbrace{\begin{bmatrix} e_{k_o}(1) \\ \vdots \\ e_{k_o}(N_t) \end{bmatrix}}_{e_{k_o}\text{: model prediction error}} \tag{5.10}$$

$$\min \frac{1}{2} e_{k_o}^T e_{k_o} \Rightarrow \hat{\bar{b}}^{k_o} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T z_{k_o}$$

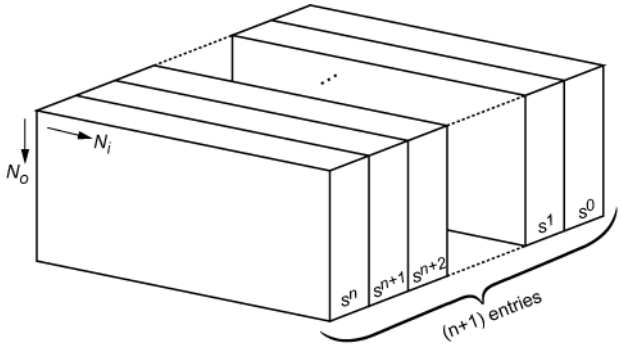


Figure 5-5 Numerator term matrix visualization

Once the scaled numerator terms have been estimated, the parameters must be unscaled which results in the participation factors of the regressors being calculated, so that Eqn. (5.11) remains valid.

$$\begin{aligned} z_{k_o} = {} & \bar{\varphi}_0^1 \bar{b}_0^{k_o,1} + \bar{\varphi}_1^1 \bar{b}_1^{k_o,1} + \cdots + \bar{\varphi}_n^1 \bar{b}_n^{k_o,1} \\ & + \bar{\varphi}_0^2 \bar{b}_0^{k_o,2} + \bar{\varphi}_1^2 \bar{b}_1^{k_o,2} + \cdots + \bar{\varphi}_n^2 \bar{b}_n^{k_o,2} \\ & + \cdots + \bar{\varphi}_0^{N_i} \bar{b}_0^{k_o,N_i} + \bar{\varphi}_1^{N_i} \bar{b}_1^{k_o,N_i} + \cdots + \bar{\varphi}_n^{N_i} \bar{b}_n^{k_o,N_i} \end{aligned} \tag{5.11}$$

$$z_{k_o} = \underbrace{\sum_{k_i=1}^{N_i} \sum_{l=0}^{n} \varphi_l^{k_i} b_l^{k_o,k_i}}_{\text{unscaled form}} = \underbrace{\sum_{k_i=1}^{N_i} \sum_{l=0}^{n} \bar{\varphi}_l^{k_i}(k) \bar{b}_l^{k_o,k_i}}_{\text{scaled form}}$$

This is achieved by noting Eqn. (5.9) and that $\bar{\varphi}_l^{k_o,k_i} \bar{b}_l^{k_o,k_i} = \varphi_l^{k_o,k_i} b_l^{k_o,k_i}$. From these two equations, Eqn. (5.12) can be formulated to calculate the final participation factors.

$$\underbrace{b_l^{k_o,k_i}}_{\substack{\text{denormalized} \\ \text{numerator} \\ \text{term(s)}}} = \underbrace{r_l^{k_i}}_{\substack{\text{regressor} \\ \text{normalization} \\ \text{factor}}} \underbrace{\bar{b}_l^{k_o,k_i}}_{\substack{\text{normalized} \\ \text{numerator} \\ \text{term}}} \tag{5.12}$$

In some cases, the numerator structure for the MIMO transfer function component P^{k_o,k_i} is known ahead of time, i.e. some of the b_l $(l = 0, \dots, n)$ terms are zero. To simplify the problem, these terms and their corresponding regressors will be excluded from the LS estimation.

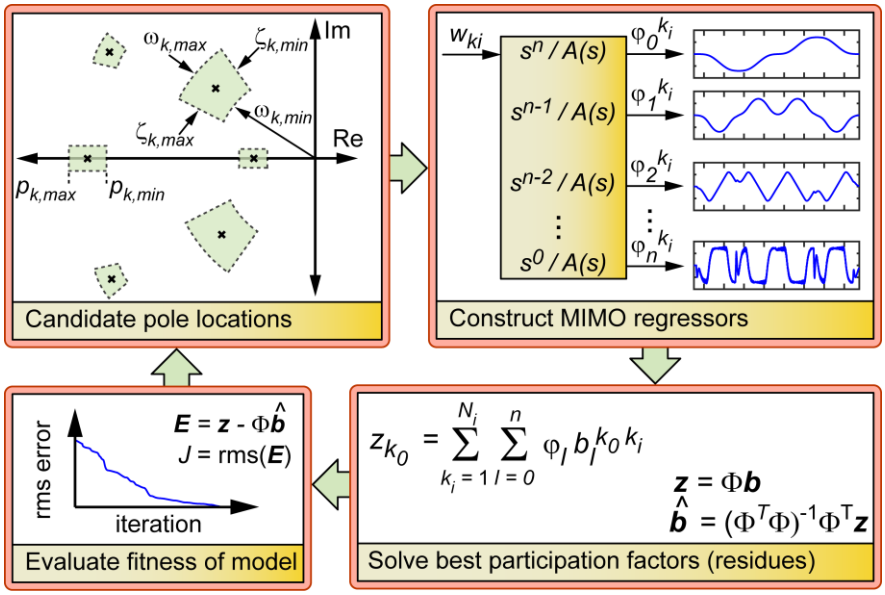


Figure 5-6 Pole search based estimation of MIMO LTI portion of model

The main scheme for the MIMO LTI algorithm can be seen above in Figure 5-6 and is implemented with the following steps:

1. Pole locations from the pre-defined stable search space are used to generate candidate models.
2. Construct the scaled SIMO discrete-time state-space regressors.
3. Apply LS and solve for the participation factors.
4. Reconstruct the predicted outputs and evaluate the RMS of the output prediction errors. The candidate pole set that has the lowest RMS error value is chosen as the identified model. As discussed in Chapter 4, MATLAB's internal global search was used to minimize the objective function.

5.2.3 D.C. Gain Condition

To derive models between the mechanical response to position commands using data collected from an accelerometer (or vibrometer) and the displacement response, a double (or single) integrator must be appended to the model estimated from the experimental data. This integration can cause drift issues in the resulting position response, therefore in this model the user is able to impose a d.c. gain (*0*th, *1*st, *2*nd, ... order) to the transfer functions. The necessary d.c. gain for the scaled transfer function, d^{k_o,k_i}, can be calculated using Eqn. (5.13), where $\tilde{d}^{k_o,k_i}$ is the d.c. gain specified for the unscaled MIMO LTI transfer function entry.

$$d^{k_o,k_i} = \frac{1}{W_z(k_o) \cdot W_w(k_i)} \tilde{d}^{k_o,k_i} \tag{5.13}$$

Considering the denormalized numerator form $b_0 \dots b_n$ and Eqn. (5.3), the relations in Eqn. (5.14) can be written for the *0*th, *1*st, and *2*nd order d.c. gain conditions. These relations can then be generalized to the d.c. gain condition of order q, as seen in Eqn. (5.15).

*0*th order d.c. gain condition:

$$[P^{k_o,k_i}(s)]_{s=0} = \frac{b_n}{a_n} = d^{k_o,k_i} = \text{desired d.c. gain}$$

*1*st order d.c. gain condition (assuming $b_n^{k_o,k_i} = 0$):

$$\left[\frac{1}{s} P^{k_o,k_i}(s)\right]_{s=0} = \frac{b_{n-1}^{k_o,k_i}}{a_n} = d^{k_o,k_i} \tag{5.14}$$

*2*nd order d.c. gain condition (assuming $b_{n-1}^{k_o,k_i} = b_n^{k_o,k_i} = 0$):

$$\left[\frac{1}{s^2} P^{k_o,k_i}(s)\right]_{s=0} = \frac{b_{n-2}^{k_o,k_i}}{a_n} = d^{k_o,k_i}$$

$$\left[\frac{1}{s^q} P^{k_o,k_i}(s)\right]_{s=0} = \frac{b_{n-q}^{k_o,k_i}}{a_n} = d^{k_o,k_i}, \text{assuming } b_{n-(q-1)}^{k_o,k_i} = \cdots = b_{n-1}^{k_o,k_i} = b_n^{k_o,k_i} = 0 \tag{5.15}$$

When there is a d.c. gain correction of order q, the numerator terms $b_n, b_{n-1}, \dots, b_{n-(q-1)}$ are set to zero and combined with Eqns. (5.13) and (5.15), b_{n-q} can be calculated as seen in Eqn. (5.16).

$$b_{n-q}^{k_o,k_i} = \frac{\tilde{d}^{k_o,k_i} \cdot a_n}{W_z(k_o) \cdot W_w(k_i)}$$

$$\left.\begin{aligned} b_n^{k_o,k_i} &= 0 \\ b_{n-1}^{k_o,k_i} &= 0 \\ &\vdots \\ b_{n-(q-1)}^{k_o,k_i} &= 0 \end{aligned}\right\} \tag{5.16}$$

Since $b_{n-q}, \dots, b_n$ are excluded from the LS estimation problem, as seen in Eqn. (5.17) , the corresponding $\varphi_{n-q}^{k_i} b_{n-q}^{k_o,k_i}$ terms must be applied as offsets to the output measurement vector z_{k_o}.

$$\varphi_0 b_0 + \cdots + \varphi_{n-(q+1)} b_{n-(q+1)} + \underbrace{\varphi_{n-q} b_{n-q}}_{\substack{\text{fixed} \rightarrow \text{remove} \\ \text{from parameter} \\ \text{estimation}}} + 0 + \cdots + 0 = z \tag{5.17}$$

Below is the pseudo code for how the d.c. gain offset correction has been implemented:

for $k_o = 1$ to N_o

    offset_correction = 0

    for $k_i = 1$ to N_i

if d.c. gain constraint of order "q" is requested for input channel k_i

$$\text{offset\_correction} = \text{offset\_correction} + \frac{\bar{\varphi}_q^{k_i}}{r_q^{k_i}} \cdot b_q^{k_o,k_i}$$

end

end

$z_{k_o} = z_{k_o} - \text{offset\_correction}$

Estimate numerator terms → $\hat{b}_{k_o} = (\Phi^T\Phi)^{-1}\Phi^T z_{k_o}$

Output prediction for channel → $\hat{z}_{k_o} = \underbrace{\Phi \cdot \hat{b}_{k_o}}_{\substack{\text{response due to} \\ \text{estimated} \\ \text{numerator terms}}} + \underbrace{\text{offset\_correction}}_{\substack{\text{response due to d.c. gain} \\ \text{constraint estimated} \\ \text{numerator term(s)}}}$

Prediction error for output channel → $e_{k_o} = z_{k_o} - \hat{z}_{k_o}$

end

calculate the overall RMS error for normalized MIMO LTI model prediction →

$e = [e_1^T, e_2^T, \dots, e_{N_o}^T],\ J = rms(e)$

5.2.4 Assembling Final Unscaled MIMO LTI Model

The regressor response due to the input $w_{k_i}(t)$ is $\varphi^{k_i}(t) = \left[\varphi_o^{k_i}(t) \dots \varphi_n^{k_i}(t)\right]^T$ or in the z-domain, $\varphi^{k_i}(z) = G_{0,ssd}(z)w_{k_i}(z)$. To compute the regressor response to N_i inputs, N_i state-space realizations are needed, as seen in Eqn. (5.18).

$$\left.\begin{aligned} \varphi^1(z) &= G_{0,ssd}(z)w_1(z) \\ &\vdots \\ \varphi^{N_i}(z) &= G_{0,ssd}(z)w_{N_i}(z) \end{aligned}\right\} \quad N_i \times n \text{ state systems} \tag{5.18}$$

However, when constructing each output z_{k_o}, the regressors may be weighted differently. For example, when $N_o = 2$ and $N_i = 3$:

$$z_1 = b^{1,1}G_{0,ssd}w_1 + b^{1,2}G_{0,ssd}w_2 + b^{1,3}G_{0,ssd}w_3$$

$$z_1 = \underbrace{\left[\begin{array}{c|c} A_{od} & B_{od} \\ \hline b^{1,1}C_{od} & b^{1,1}D_{od} \end{array}\right]}_{\substack{\text{discrete-time} \\ \text{state space system}}} w_1 + \left[\begin{array}{c|c} A_{od} & B_{od} \\ \hline b^{1,2}C_{od} & b^{1,2}D_{od} \end{array}\right] w_2 + \left[\begin{array}{c|c} A_{od} & B_{od} \\ \hline b^{1,3}C_{od} & b^{1,3}D_{od} \end{array}\right] w_3 \tag{5.19}$$

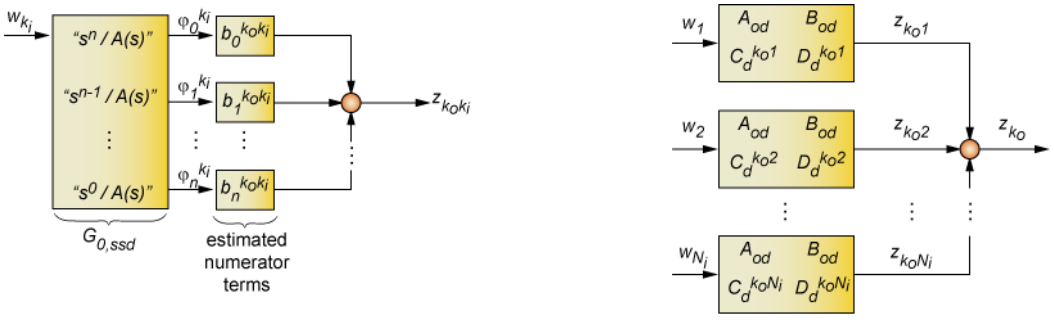
$$z_2 = \left[\begin{array}{c|c} A_{od} & B_{od} \\ \hline b^{2,1}C_{od} & b^{2,1}D_{od} \end{array}\right] w_1 + \left[\begin{array}{c|c} A_{od} & B_{od} \\ \hline b^{2,2}C_{od} & b^{2,2}D_{od} \end{array}\right] w_2 + \left[\begin{array}{c|c} A_{od} & B_{od} \\ \hline b^{2,3}C_{od} & b^{2,3}D_{od} \end{array}\right] w_3$$

Defining Eqn. (5.20), where ξ is the state vector, these terms (i.e. states) are common for both outputs z_1 and z_2.

$$\left.\begin{aligned} \xi_1(t+1) &= A_{od}\xi_1(t) + B_{od}w_1(t) \\ \xi_2(t+1) &= A_{od}\xi_2(t) + B_{od}w_2(t) \\ \xi_3(t+1) &= A_{od}\xi_3(t) + B_{od}w_3(t) \end{aligned}\right\} \tag{5.20}$$

For a single input, as illustrated in Figure 5-7(a), the component of output k_o due to the input k_i can be written as Eqn. (5.21), where the candidate models have been transformed into their state space equivalents $G_{0,ssd}$. In Figure 5-7(b), a schematic on how to combine a multi input – single output assembled system is shown.

$$\begin{aligned} z_{k_o,k_i} &= \underbrace{\left[b_0^{k_o,k_i}\ b_1^{k_o,k_i}\ \ldots\ b_n^{k_o,k_i}\right]}_{\substack{\text{vector of numerator} \\ \text{terms } b^{k_o k_i}}} \{\boldsymbol{C}_{od}(sI - \boldsymbol{A}_{od})^{-1}\boldsymbol{B}_{od} + \boldsymbol{D}_{od}\} \\ &= \left\{ \underbrace{b^{k_o,k_i}\boldsymbol{C}_{od}}_{C_d^{k_o,k_i}} (sI - \boldsymbol{A}_{od})^{-1}\boldsymbol{B}_{od} + \underbrace{b^{k_o,k_i}\boldsymbol{D}_{od}}_{D_d^{k_o,k_i}} \right\} \end{aligned} \tag{5.21}$$



(a) (b)

Figure 5-7 Elemental discrete-time state-space model(s) (a) SIMO (b) MIMO

The overall state-transition and input matrices for the assembled MIMO scaled system can be written as seen in Eqn. (5.22), where A_d is a $(N_i \times n) \times (N_i \times n)$ matrix and B_d is a $(N_i \times n) \times N_i$ matrix.

$$\underbrace{\begin{bmatrix} \xi_1(t+1) \\ \xi_2(t+1) \\ \vdots \\ \xi_{N_i}(t+1) \end{bmatrix}}_{\xi(t+1)} = \underbrace{\begin{bmatrix} \boxed{A_{od}} & 0 & \cdots & 0 \\ 0 & \boxed{A_{od}} & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \boxed{A_{od}} \end{bmatrix}}_{A_d} \underbrace{\begin{bmatrix} \xi_1(t) \\ \xi_2(t) \\ \vdots \\ \xi_{N_i}(t) \end{bmatrix}}_{\xi(t)} + \underbrace{\begin{bmatrix} \boxed{B_{od}} & 0 & \cdots & 0 \\ 0 & \boxed{B_{od}} & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \boxed{B_{od}} \end{bmatrix}}_{B_d} \underbrace{\begin{bmatrix} w_1(t) \\ w_2(t) \\ \vdots \\ w_{N_i}(t) \end{bmatrix}}_{w(t)} \tag{5.22}$$

The output z_{k_0} can be written in terms of the state vector ξ_{k_i} of the elemental SIMO systems $G_{0,ssd}$, where each system receives one input channel w_{k_i}, as seen in Eqn. (5.23). In this equation C_d is a $N_o \times (N_i \times n)$ matrix and D_d is a $N_o \times N_i$ matrix.

$$\begin{aligned} z_{k_0} &= b^{k_0,1}(C_{od}\xi_1 + D_{od}w_1) + b^{k_0,2}(C_{od}\xi_2 + D_{od}w_2) + \cdots \\ &\quad + b^{k_0,N_i}\left(C_{od}\xi_{N_i} + D_{od}w_{N_i}\right) \end{aligned}$$

$$\underbrace{\begin{bmatrix} z_1(t) \\ z_2(t) \\ \vdots \\ z_{N_o}(t) \end{bmatrix}}_{\substack{z(t)\text{ normalized} \\ \text{output variables}}} = \underbrace{\begin{bmatrix} \boxed{b^{1,1}C_{od}} & \boxed{b^{1,2}C_{od}} & \cdots & \boxed{b^{1,N_i}C_{od}} \\ \boxed{b^{1,2}C_{od}} & \boxed{b^{2,2}C_{od}} & \cdots & \boxed{b^{2,N_i}C_{od}} \\ \vdots & \vdots & \ddots & \vdots \\ \boxed{b^{N_o,1}C_{od}} & \boxed{b^{N_o,1}C_{od}} & \cdots & \boxed{b^{N_o,N_i}C_{od}} \end{bmatrix}}_{C_d} \underbrace{\begin{bmatrix} \xi_1(t) \\ \xi_2(t) \\ \vdots \\ \xi_{N_i}(t) \end{bmatrix}}_{\xi(t)}$$

$$+ \underbrace{\begin{bmatrix} \boxed{b^{1,1}D_{od}} & \boxed{b^{1,2}D_{od}} & \cdots & \boxed{b^{1,N_i}D_{od}} \\ \boxed{b^{1,2}D_{od}} & \boxed{b^{2,2}D_{od}} & \cdots & \boxed{b^{2,N_i}D_{od}} \\ \vdots & \vdots & \ddots & \vdots \\ \boxed{b^{N_o,1}D_{od}} & \boxed{b^{N_o,1}D_{od}} & \cdots & \boxed{b^{N_o,N_i}D_{od}} \end{bmatrix}}_{D_d} \underbrace{\begin{bmatrix} w_1(t) \\ w_2(t) \\ \vdots \\ w_{N_i}(t) \end{bmatrix}}_{w(t)} \tag{5.23}$$

From Eqns. (5.22) and (5.23), the final normalized MIMO LTI model can be assembled, as seen in Eqn. (5.24). The unscaled MIMO model can then be calculated using Eqn. (5.25).

$$P = \left[\begin{array}{c|c} A_d & B_d \\ - & - \\ C_d & D_d \end{array}\right] \tag{5.24}$$

$$\begin{aligned} \tilde{P} &= W_z P W_w = W_z\{C_d(zI - A_d)^{-1}B_d + D_d\}W_w \\ \tilde{P} &= \underbrace{W_z C_d}_{\tilde{C}_d}(zI - A)^{-1}\underbrace{B_d W_w}_{\tilde{B}_d} + \underbrace{W_z D_d W_w}_{\tilde{D}_d} \\ \tilde{P}(z) &= \tilde{C}_d(zI - A)^{-1}\tilde{B}_d + \tilde{D}_d \end{aligned} \tag{5.25}$$

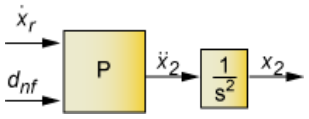


Figure 5-8 Example feed drive model structure with double integrator

5.2.5 Numerical Implementation of D.C. Gain

When trying to predict a position response from an MIMO LTI acceleration (or vibration) model, a double (or single) integrator must be appended to the identified model. For example, if a plant P_{k_o} represents the normalized acceleration response, $\ddot{x}$, to the normalized commanded velocity, $\dot{x}_r$, and the normalized Coulomb friction approximation, d_{n_f}, in order to obtain the normalized position response, a double integrator would need to be appended to the model, as seen in Figure 5-8. Since the numerator terms $b_0 s^n + b_1 s^{n-1} + \cdots + b_{n-1} s^1 + b_n s^0$ are estimated using regressors that are generated via discrete-time state space models which emulate the responses of $\frac{s^n}{A(s)}, \frac{s^{n-1}}{A(s)}, \ldots, \frac{s^0}{A(s)}$, the effect of appending integrator(s) to a certain output can be achieved by shifting the numerator terms belonging to that output channel to lower orders. For example, consider the expected position response to position command and disturbance input as seen in Eqn. (5.26). Investigating the acceleration response of Eqn. (5.26) yields Eqn. (5.27). The higher order terms can be neglected, or the non-causal terms can also be estimated by applying the commanded acceleration directly as an additional input, as seen in Figure 5-8.

$$\underbrace{x(s)}_{\text{output \#1}} = \frac{b_o^{1,1}s^3 + b_1^{1,1}s^2 + b_2^{1,1}s + b_3^{1,1}}{\underbrace{s^3 + a_1 s^2 + a_2 s + a_3}_{A(s)}} \underbrace{x_r(s)}_{input\ \#1} + \frac{b_o^{1,2}s^3 + b_1^{1,2}s^2 + b_2^{1,2}s + \overbrace{b_3^{1,2}}^{=0}}{A(s)} \underbrace{d_{nf}(s)}_{input\ \#2} \tag{5.26}$$

$$\ddot{x}(s) = \frac{\boxed{b_0^{1,1}s^5 + b_1^{1,1}s^4} + b_2^{1,1}s^3 + b_3^{1,1}s^2}{s^3 + a_1 s^2 + a_2 s + a_3} x_r(s) + \frac{\boxed{b_0^{1,2}s^5 + b_1^{1,2}s^4} + b_2^{1,2}s^3 + (0)s^2}{s^3 + a_1 s^2 + a_2 s + a_3} d_{nf}(s) \tag{5.27}$$

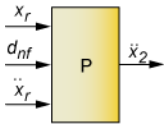


Figure 5-9 Applying commanded acceleration to estimate higher order terms

To simplify the acceleration response estimation, the s^5 and s^4 terms are neglected as seen in Eqn. (5.28). Afterwards, Eqn. (5.29) is achieved by applying double integration to construct the position response. The shifting caused by the addition of the two integrators is illustrated in Figure 5-10.

$$\ddot{x}(s) \cong \frac{b_0^{1,1}s^3 + b_1^{1,1}s^2 + (0)s^1 + 0s^0}{A(s)} x_r(s) + \frac{b_0^{1,2}s^3 + (0)s^2 + (0)s^1 + 0s^0}{A(s)} d_{nf}(s)$$

$$\ddot{x}(s) \cong \frac{b_0^{1,1}s^3 + b_1^{1,1}s^2}{A(s)} x_r(s) + \frac{b_0^{1,2}s^3}{A(s)} d_{nf}(s) \tag{5.28}$$

$$x = \frac{1}{s^2}\ddot{x} = \frac{b_0^{1,1}s^1 + b_1^{1,1}}{A(s)} x_r + \frac{b_0^{1,2}s^1}{A(s)} d_{nf} \tag{5.29}$$

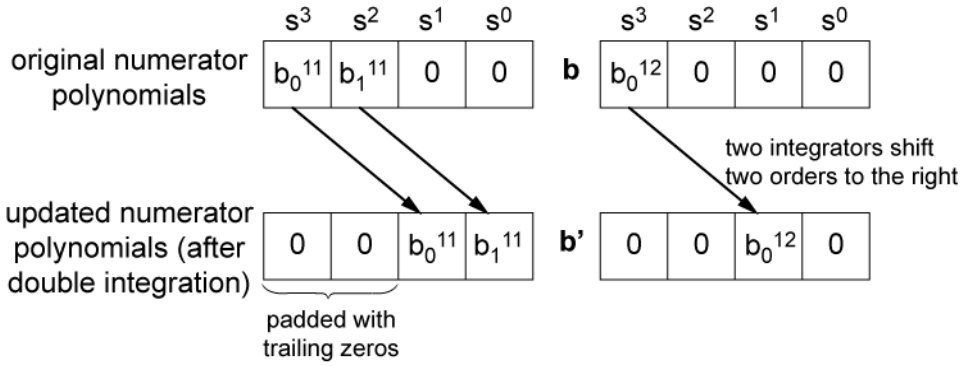


Figure 5-10 Shifting polynomial numerators after double integration

After the necessary numerator vectors are updated, the normalized MIMO LTI system is reassembled, and the model can be denormalized once again.

5.3 Experimental Validation

This new MIMO model has been validated experimentally on a five axis milling machine from the P&WC manufacturing line in Longueil, Quebec. The servo data was collected at a sampling time of 1ms using the software Servo Viewer from the FANUC controller. The concept of the experimental identification process can be seen in Figure 5-11.

Two different trajectories were tested on this machine, a spiral trajectory with movements in the x- and y- axes, and a roughing operation for an air foil blade. The spiral trajectory will be used for parameter identification and the air foil blade process will be used for validation of the model. Only the x- and y-axes were identified for this machine tool.

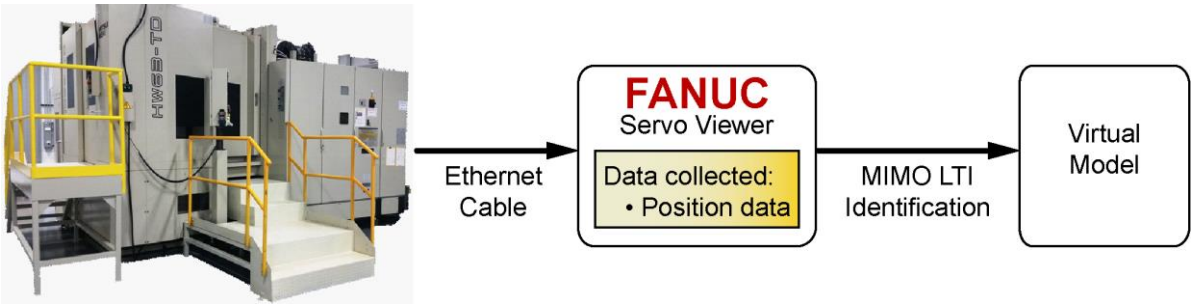


Figure 5-11 Experiment identification process for MIMO LTI model (representative picture of machine-tool obtained from [57])

5.3.1 Results

Three models will be considered in this analysis, which are illustrated in Figure 5-12 below, a *3*rd order decoupled model (the x- and y-axes will be identified separately) and two *6*th order coupled model (x- and y-axes will have the same denominator but different numerator terms). The decoupled model (model 1) has two inputs, the commanded axis position and normalized Coulomb friction, and two outputs the axis tracking error and the axis velocity. The *6*th order model has four inputs, the commanded position and normalized Coulomb friction for both the x- and y-axes, and four outputs, the tracking error and axis velocity for both axes. Model 2 has search bounds based on the model 1 parameters, while model 3 has search bounds that are more general. This was done to determine if the algorithm is able to identify similar dynamics with different search spaces and to determine if the identified parameters are able to predict similar dynamics if the parameters are not similar. Also by limiting the search space for model 2, this shortens the time spent on identification. The bounds for the stable search space are seen in Table 5-1, where 500 Hz is the Nyquist frequency of the data.

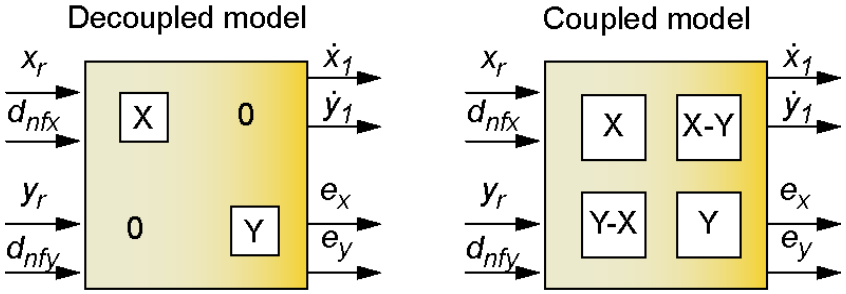


Figure 5-12 Different models considered for the five axis milling machine

Table 5-1 MIMO pole search bounds for X- and Y- axes of the five axis milling machine

Parameters	Model 1		Model 2		Model 3	
	Min	Max	Min	Max	Min	Max
$p_{r,1}$ [Hz]	0.016	500.0	2.00	7.00	0.016	500.00
$p_{r,2}$ [Hz]	-	-	1.00	6.00	0.016	500.00
$w_{k,1}$ [Hz]	1.592	500.0	15.00	30.00	1.59	500.00
$\zeta_{k,1}$ []	0.01	2	0.10	2.00	0.010	4.00
$w_{k,2}$ [Hz]	-	-	20.00	35.00	1.59	500.00
$\zeta_{k,2}$ []	-	-	0.10	2.00	0.010	4.00

The estimated parameters for both models can be seen in Table 5-2 below. In model 1, the decoupled model, it can be seen that the x- and y-axes have close real poles, at 5.23 and 4.23 Hz respectively, and relatively close natural frequencies, at 20.68 and 28.04 Hz respectively. It is also observed that the zeros for both axes are also similar, this indicates that the axes have well-tuned dynamics between both axes. However, there is a slight discrepancy in the damping ratio for model 1, this can be from using smooth trajectories to tune the control system in time-domain, which can prevent oscillatory behaviours from occurring.

Comparing model 1 and model 2, it can be seen that similar dynamics have been captured. The x- and y-axis natural frequencies identified in model 1 are also captured by model 2, with a difference of 1- Hz, with similar damping ratios. The real poles identified by model 2 are also similar to the independently identified poles from model 1, the first real pole in model 2, $p_{r,1}$, is very similar to the x-axis real pole, 5.23 and 4.90 Hz respectively. There is a bigger discrepancy between the second pole in model 2, $p_{r,2}$, and the y-axis real pole.

Observing model 3, it can be seen that there are some similarities when comparing to model 1 and 2. The two real poles, 29.642 and 29.640 Hz, has a similar frequency to the complex conjugate poles identified in model 1 and 2, 28.04 Hz identified for the y-axis of model 1 and 30.74 Hz for model 2. The first complex pole of model 3, $w_{k,1} = 3.73$ Hz, is similar to the real poles identified of both model 1, 4.23 Hz for the real pole of the y-axis, and model 2, 2.53 and 4.90 Hz. The second complex pole of model 3, $w_{k,2} = 22.15$ Hz, is similar to the complex poles identified by model 1, 20.68 Hz for the x-axis, and model 2, 22.08 Hz. It should also be noted that the zeros for model 2 and 3 are very similar, approximately 1-20% difference. Therefore, although the real and complex conjugate poles are not exact matches between all three models, it can be seen that similar dynamics are captured between the different configurations.

Table 5-2 MIMO identified parameters for X- and Y- axes of the five axis milling machine

Param.	Model 1				Model 2				Model 3			
	X Axis		Y Axis		X Axis		Y Axis		X Axis		Y Axis	
$p_{r,1}$ [Hz]	5.23		4.23		4.90				29.642			
$p_{r,2}$ [Hz]	-		-		2.53				29.640			
$w_{k,1}$ [Hz]	20.68		28.04		22.08				3.73			
$\zeta_{k,1}$ []	0.29		0.76		0.10				1.10			
$w_{k,2}$ [Hz]	-		-		30.74				22.15			
$\zeta_{k,2}$ []	-		-		0.87				0.08			
Track. TF Zeros	**Freq [Hz]**	**Damp []**	**Freq [Hz]**	**Damp []**	**Freq [Hz]**	**Damp []**	**Freq [Hz]**	**Damp []**	**Freq [Hz]**	**Damp []**	**Freq [Hz]**	**Damp []**
z_1	0.00	-1.00	0.00	-1.00	0.00	-1.00	0.00	-1.00	0.00	-1.00	0.00	-1.00
z_2	1.24	0.24	1.07	0.18	27.19	0.38	22.61	0.22	26.38	0.35	22.54	0.21
z_3	1.24	0.24	1.07	0.18	27.19	0.38	22.61	0.22	26.38	0.35	22.54	0.21
z_4	-	-	-	-	2.74	1.00	2.52	1.00	2.83	1.00	2.61	1.00
z_5	-	-	-	-	1.14	0.26	1.13	0.13	1.14	0.26	1.12	0.13
z_6	-	-	-	-	1.14	0.26	1.13	0.13	1.14	0.26	1.12	0.13

Below in Table 5-3 summaries for the tracking and contour prediction can be seen for the training spiral trajectory and the verification roughing operation. In Figure 5-13 and Figure 5-14, the commanded, measured, and predicted axis positions for all the models are shown for the spiral and the roughing trajectories respectively. The contouring error prediction results can be seen in Figure 5-15.

From Table 5-3, it can be seen that model 2 and model 3 are able to predict the RMS and maximum tracking and contouring errors very similarly, however model 3 has a slight improvement in prediction between 0.1 to 1.3% improvement. The *6*th order model with the smaller search space converges onto parameters more quickly than the configuration with the larger search space, however the *3*rd order decoupled parameters would have to be identified first to determine the smaller search space.

Table 5-3 Tracking and contouring error prediction for Deckel Maho 80P MIMO

Discrepancy in Servo Error of Prediction	Spiral			Air Foil Blade		
	Model 1 - Decoupled	Model 2 - Coupled	Model 3 - Coupled	Model 1 - Decoupled	Model 2 - Coupled	Model 3 - Coupled
X axis tracking error: RMS (MAX) [µm]	0.34 (2.65)	**0.30 (2.84)**	**0.30 (2.84)**	1.82 (20.33)	**1.79** (19.94)	**1.79 (19.88)**
Y axis tracking error: RMS (MAX) [µm]	0.30 (1.96)	**0.26** (1.68)	**0.26 (1.65)**	1.25 (11.36)	1.06 **(11.17)**	**1.05** (11.19)
Contouring error: RMS (MAX) [µm]	0.31 (2.05)	**0.27** (2.01)	**0.27 (1.98)**	1.88 (22.72)	1.83 (22.60)	**1.82 (22.53)**

From Figure 5-13 and Table 5-3, it can be seen that all the models are able to predict the tracking response to the machine tool. However, it is evident that both model 2 and 3 is able to more closely predict both the tracking and contour error, with model 3 having a slight advantage. For model 2 and 3, there is a 14 and 17% improvement from model 1 when comparing the RMS value of tracking error prediction of the x-and y-axes respectively. When comparing the maximum value of tracking error prediction, there is an improvement when using the coupled models, approximately 7 to 19% for the x- and y-axes respectively.

Observing Figure 5-14 and Table 5-3, it can be seen that by using the identified transfer functions from the spiral trajectory, the tracking response of a production part can be estimated. It should be noted that the tracking and contour error of the air foil has a high initial prediction discrepancy for the measured experimental data, this is due to the trajectory not having zero initial conditions which is an assumption that is made during simulation. From Table 5-3, it can be seen that the estimation of the x-axis prediction is similar between all the models, about 2% improvement in both the RMS tracking error prediction and in maximum tracking error prediction using model 3. However model 3 provides a better estimation for the y-axis positioning from 1.25 to 1.05 μm, about 19% improvement in RMS tracking error estimation.

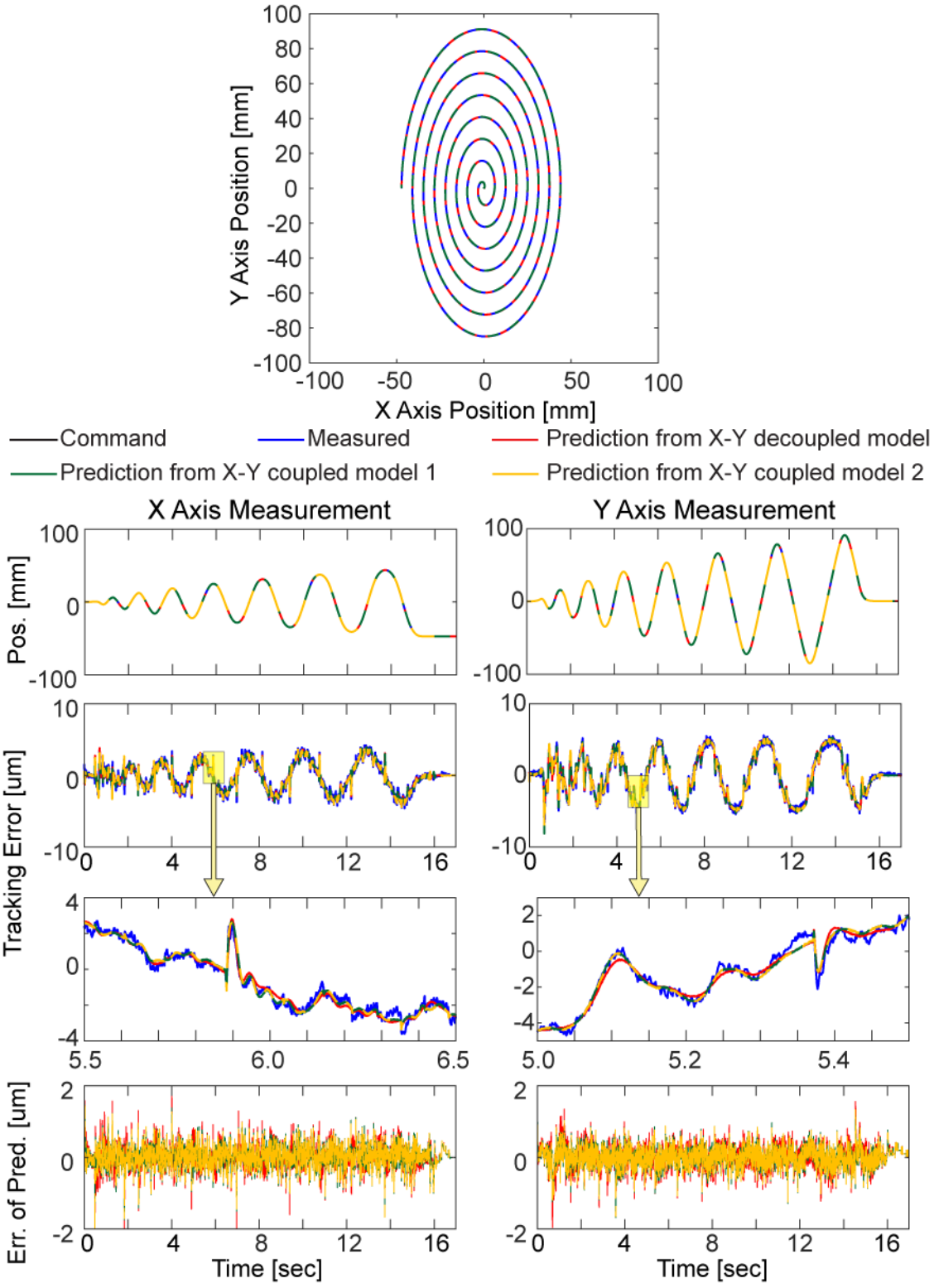


Figure 5-13 Spiral identification P&WC results

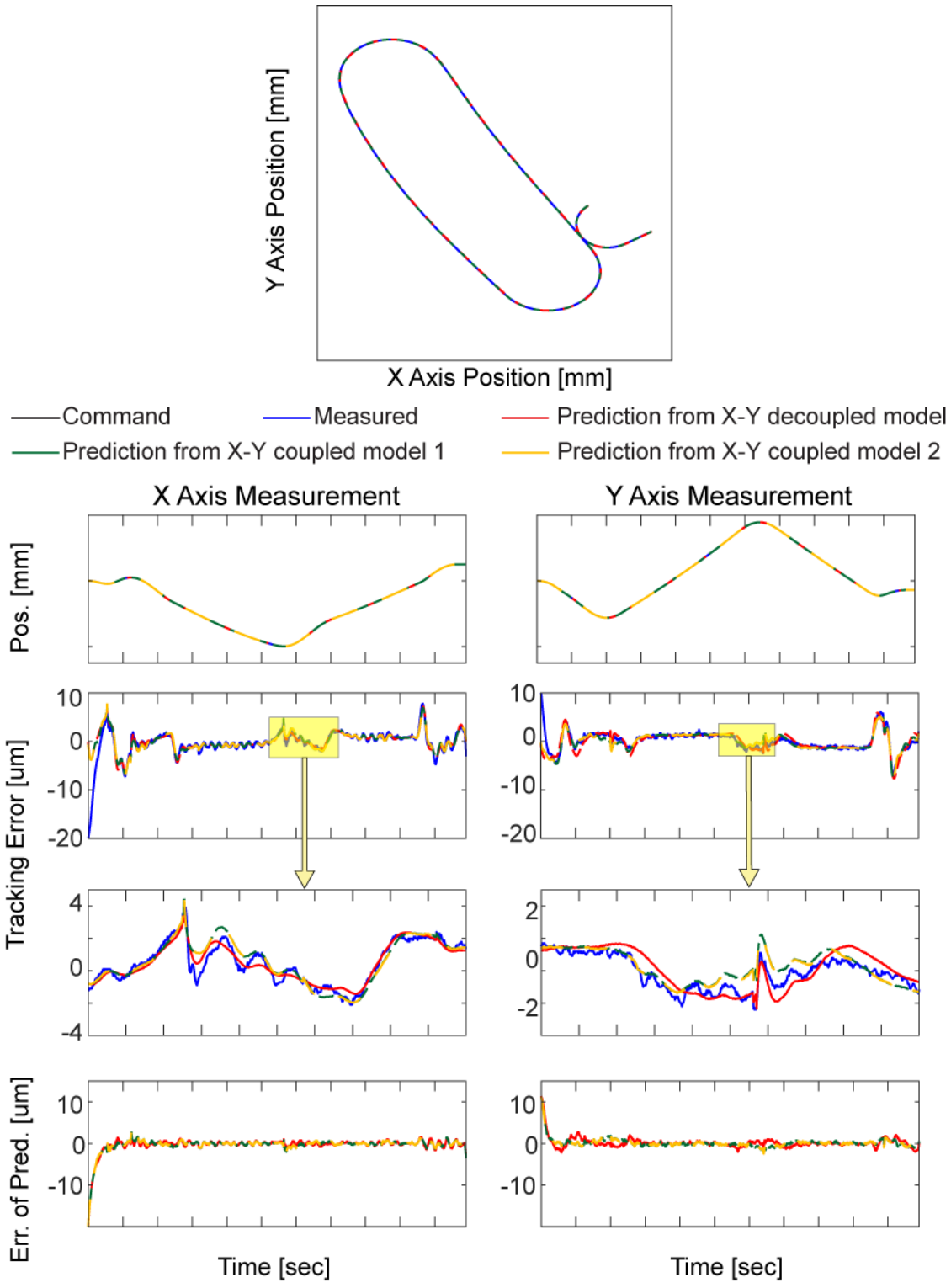


Figure 5-14 Air foil blade verification P&WC results

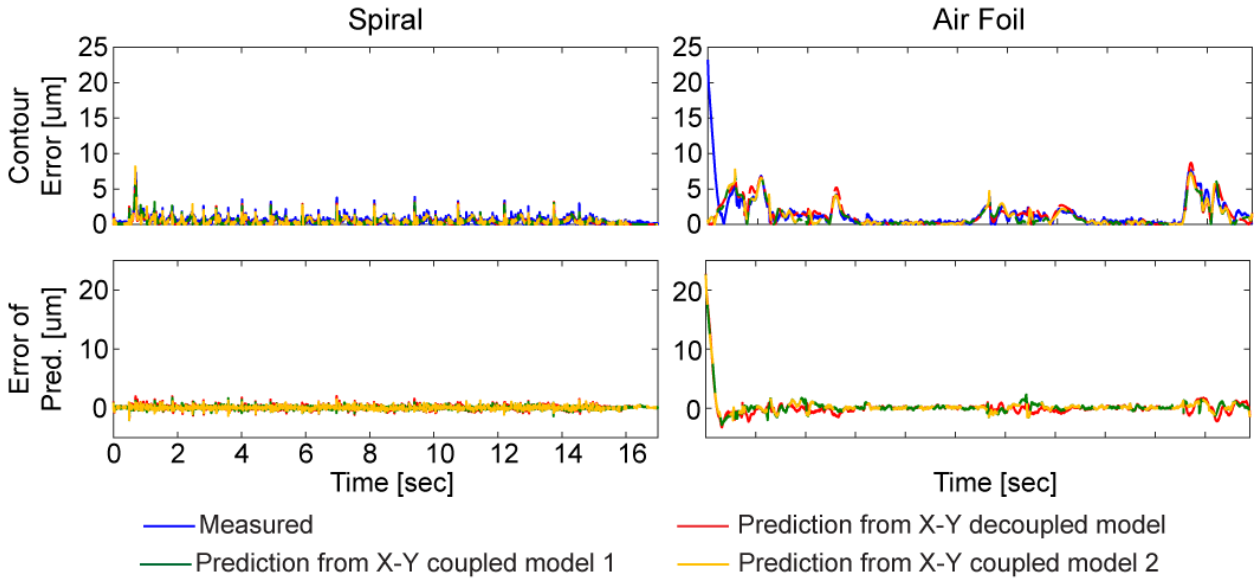


Figure 5-15 Contour error for spiral and fanbalde trajectories

The contouring error and contour error prediction can be seen in Figure 5-15. For the contouring error of the spiral trajectory, there is a 17% improvement for when using model 3 comparing the RMS contour prediction error and 4% improvement when estimating maximum contour error over model 1. When predicting the tracking of the air foil, the RMS tracking error prediction improved by 3% while the maximum prediction error improved by 1% when using model 3 instead of model 1.

5.4 Conclusions

A MIMO model with the capability of identifying flexible dynamics is presented in this chapter, which is a generalized algorithm built on the SISO model from Chapter 4. Transfer functions between different input and outputs can be identified for machine tools, provided that the relationships are linear-time invariant. Two trajectories were run on a five axis milling machine, a spiral trajectory used for identification of transfer functions and an air foil roughing operation used for parameter verification. Three different models were identified from the experimental data, the first is a *3*rd order decoupled model (one model was identified per axis) and two *6*th order coupled models were identified using different search bounds. Both models are able to capture the dynamics of the machine tool, however the *6*th order model with a larger search space has reduced tracking and contour prediction error, 2-19% improvement in estimating both the RMS tracking error and the maximum tracking error. It should be noted that the *6*th order model with a smaller search space is also able to predict the same metrics as the *6*th order model with the larger search space, within 0.1 to 0.4%. Since the smaller search space is able to converge to parameters more quickly,

it could be advantageous to use this configuration in real world applications where parameters have already been identified for a machine tool.

Chapter 6
Experimental Setup for MIMO Rapid Identification

6.1 Introduction

A simplification that is often made when developing new identification models is that machine tools are rigid. However, for large machine tools and machine tools with large overhang, this simplification is not well founded. Vibratory modes could also be introduced by the end user of machine tools, from the fixturing, tooling or even materials used in the process. A new model was proposed in Chapter 5 that can take into consideration the vibrations of machine tools while developing a dynamic model. This new model was able to identify the vibration modes in simulations, however this capability has yet to be verified with experimental data. The small/medium sized machine tools that are readily available in the lab to run experimental tests do not exhibit vibratory dynamics. To achieve the goal of gathering data which contains a vibration mode on a rigid machine tool, an interface is needed between the feed drive and the spindle to simulate the dynamics, which can be achieved by a flexure.

The flexure mount design in this chapter is based on the design of another flexure designed for chatter testing in the lab by Kenneth Wong [58]. The main goal of this flexure is to provide an experimental setup that is both repeatable and safe to use to collect data on machine tools. This setup is to be compatible with two test beds available to the lab, the Deckel Maho machining center (DMG) (a 5-axis machine tool), and the Precision Machine Tool (PMT) (a 3 to 5 axis reconfigurable platform).

6.2 Designing the Experimental Setup

The flexure mount is designed to be able to adapt to different experimental setups, with one or two modes of vibrations that occur in one or two axes, as illustrated in Figure 6-1. There are three configurations for the flexure mount: a) one flexible mode in one direction, b) two flexible modes both in the same direction, and c) two flexible modes in opposite directions. These modes of vibration are achieved by making the bending directions stiffer in two directions by using sheet metal side plates between the base plates.

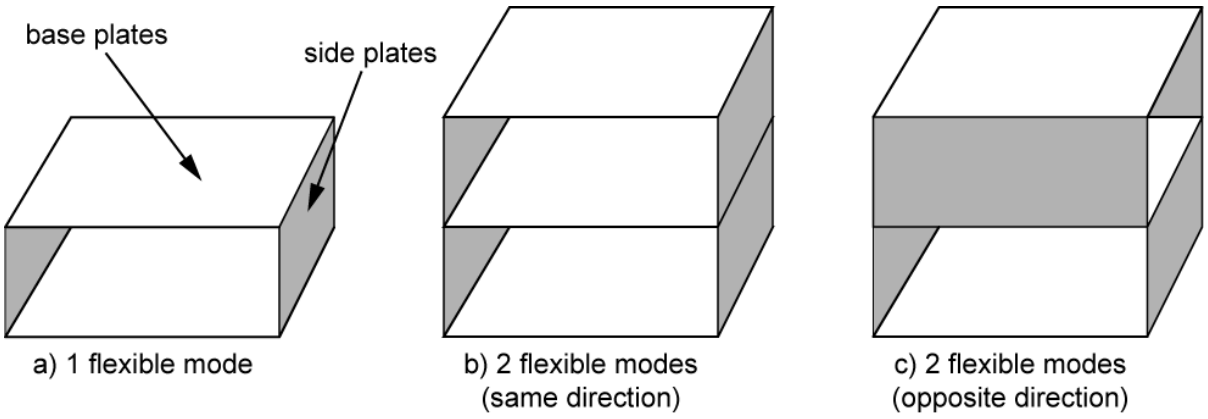
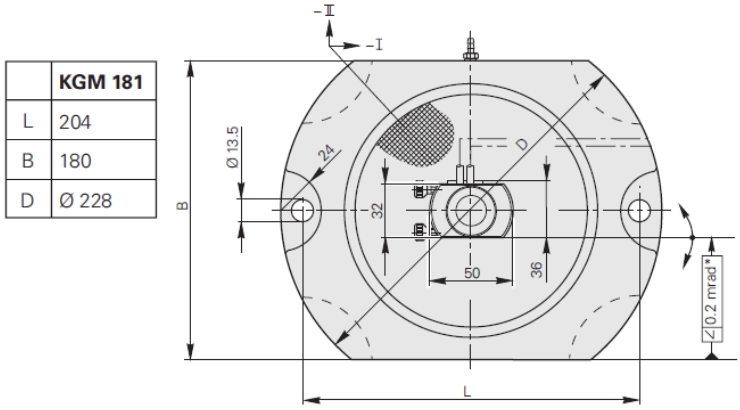


Figure 6-1 Preliminary sketch of different flexure mount configurations

6.2.1 Base Plate Design

Several design constraints were placed for the base plate design of the flexure, particularly the thicker plates of metal that will give mass to the flexure setup. The flexure would either have two or three base plates (depending on the experimental setup). The material used for the base plates is aluminum for easier machining and to ensure that the setup would not rust in a laboratory environment. The major constraints were the size of the KGM, a 2D grid encoder that will be used for model verification, and the hole locations of the machine tools that will be used to secure the flexure. The KGM has a length of 204 mm and a width of 180 mm, as seen in Figure 6-2.

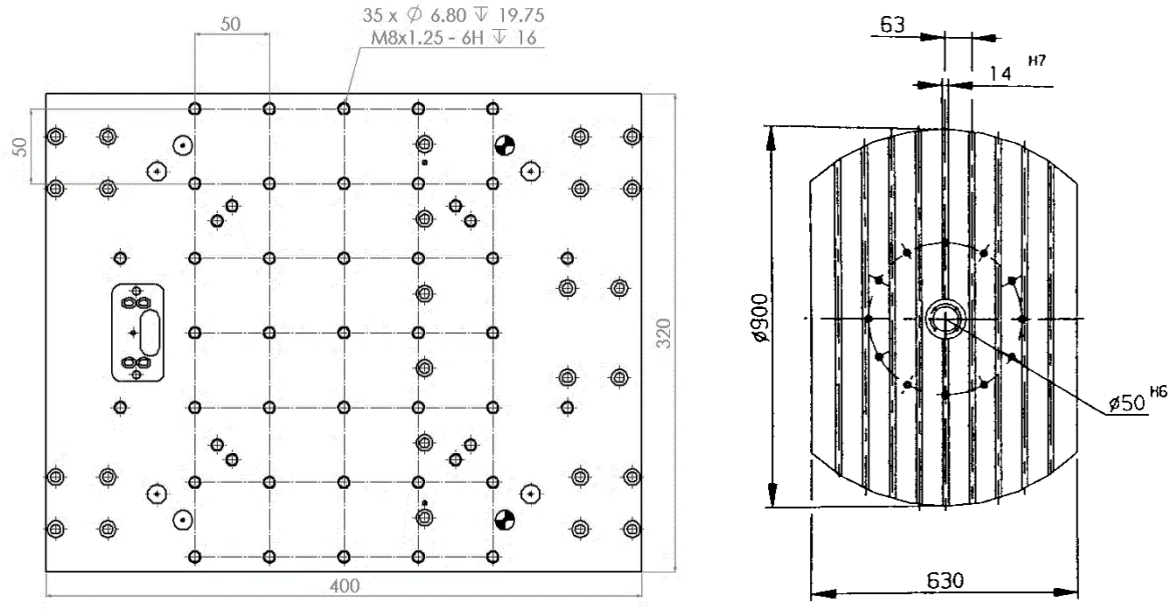


	KGM 181
L	204
B	180
D	Ø 228

Figure 6-2 KGM grid encoder dimensions [59]

The bottom base plate would need to be secured to the two machine tools. The PMT table, in Figure 6-3 (a), has M8 holes that are in a grid pattern 50 mm apart. The DMG machine tool has slots that are 63 mm apart along with 1/2" holes in a circular pattern, as seen in Figure 6-3 (b). Therefore, it was decided

that the length and width of the flexure would be 250 by 250 mm with different hole patterns for the M8 and 1/2" holes for mounting.



(a) (b)

Figure 6-3 Table dimensions for (a) PMT stage [mm], (b) Deckel Maho machining center [mm] [60]

The maximum force the flexure was designed to withstand is the peak force of the PMT stage. The feed drive in the PMT stage is an ETEL ILM-60-060 linear magnetic motor, which has a peak force of 1270 N [61]. Class 12.9 metric screws, which have a minimum yield strength of 1100 MPa [62], will be used throughout the design on the flexure. The maximum transverse shear stress the bolts will encounter under the peak force is calculated using Eqn. (6.1), where τ_{max} is the maximum transverse shear stress for a circular cross section, V is the shear force, and d is the diameter.

$$\tau_{max} = \frac{4V}{3A} = \frac{16V}{3\pi d^2} \tag{6.1}$$

After substituting the corresponding values and adding a factor of safety of 1.5, the maximum transverse shear stress for a single M8 fastener would be 50.5MPa, which is less than the minimum yield strength of a single bolt.

M6 bolts are used in the flexure design to secure the side plates to the base plates because they are readily available and meet the minimum required tensile strength. The maximum tensile stress, σ_{TS}, these bolts will encounter is 67 MPa which is calculated using Eqn. (6.2), where F is the maximum force and A_t is the bolt tensile area. This tensile stress is less than the minimum yield strength of each bolt.

$$\sigma_{TS} = \frac{F}{A_t} \tag{6.2}$$

6.2.2 Side Plate Design

The material used for the side plates is stainless steel so that the setup would not rust over time. The thickness of the plates is 16 gauge. The height of the sheet metal plates will be varied in the next section to get the desired modes of flexibility. To simulate frequency response in Solidworks, the joint stiffness of the setup must be calculated. The parameters for the side plate fasteners calculations can be found in Table 6-1.

Table 6-1 Parameters for side plate design

Parameter	Value
Fastener head diameter (D)	10 mm
Bolt diameter (d)	6 mm
Steel modulus of elasticity (E)	207 GPa
Unthreaded cross-sectional area (A_d)	28.7 mm²
Threaded cross-sectional area (A_t)	20.1 mm²
Washer thickness	1.75 mm
Side plate thickness	1.52 mm
Total material thickness (t)	3.27 mm
Threaded grip (l_t)	20 mm
Unthreaded grip (l_d)	0 mm

First the bolt stiffness, k_b, must be calculated with Eqn. (6.3) using the parameters above. The bolt stiffness is calculated as 208 kN/mm per bolt.

$$k_b = \frac{A_d A_t E}{A_d l_t + A_t l_d} \text{ [62]} \tag{6.3}$$

Next, the material stiffness, k_m, must be calculated. Since the frustum of the bolt does not exceed the diameter of the washer and the washer and side plate have the same modulus of elasticity, the two items are considered as one thickness. The material stiffness is calculated as 4974 kN/mm per bolt.

$$k_m = \frac{0.5774\pi E d}{ln\frac{(1.155t+D-d)(D+d)}{(1.155t+D+d)(D-d)}} \text{ [62]} \tag{6.4}$$

The bolt and material stiffness are applied in Solidworks over a contact surface rather than at the bolt locations. Therefore, the contact surface area between the baseplate and the sideplates was also taken into account during the simulation to determine an equivalent stiffness, as seen in Eqn. (6.5).

$$\left.\begin{aligned} k_{m_{sim}} &= \frac{k_m \times A_{bolt}}{A_{surface}} \\ k_{b_{sim}} &= \frac{k_b \times A_{bolt}}{A_{surface}} \end{aligned}\right\} \tag{6.5}$$

The top mass and the bottom mass were designed with different hole patterns and therefore have different total bolt and material stiffness values, as seen in Table 6-2.

Table 6-2 Total bolt and material stiffness for different configurations

	4 bolts	**5 bolts**
Total bolt stiffness [kN/mm]	192	300
Total material stiffness [kN/mm]	4595	7179

Finally, the joint stiffness k can be calculated using Eqn. (6.6), and the final results can be seen in Table 6-3.

$$\frac{1}{k} = \frac{1}{k_b} + \frac{1}{k_m} \tag{6.6}$$

Table 6-3 Joint stiffness for different configurations

	4 bolts	**5 bolts**
Joint stiffness [kN/mm]	18.45	28.82

6.2.3 Solidworks Simulation

The thicknesses and size of the base plates are fixed in this design (based on the constraints and available thicknesses of stock material). The only parameter that could be varied was the height of the sheet metal side plates to get the different frequency modes. There were three main criteria for determining the frequency modes, 1) the modes should be greater than 20 Hz, 2) the modes should be more than 20 Hz apart and 3) the modes should not be multiples of each other. By varying the heights of the sheet metal side plates, the natural frequencies are simulated in Solidworks, and the results can be seen in Table 6-4. Additionally, the final side plate heights can be seen in Table 6-5.

Table 6-4 Frequency simulation results for flexure

Frequency	**One Mode**	**Two-Mode**	
		Same Direction	**Opposite Direction**
1	27.33 Hz	23.07 Hz	27.60 Hz
2	-	59.70 Hz	64.87 Hz

Table 6-5 Final side plate height

Side Plate Height	Single Mode	Two-Mode	
		Same Direction	Opposite Direction
Side plate 1	130 mm	130 mm	130 mm
Side plate 2	-	100 mm	70 mm

In Figure 6-4, Figure 6-5, and Figure 6-6, the simulated deformations of the single mode, two-mode same direction, and two-mode opposite direction flexures can be seen. In can be seen in the two-mode same direction flexure, that both the top and middle base plates are moving in the same direction for the first mode, while for the second mode, the plates are moving in opposite directions. For the two-mode opposite direction flexure, the simulations show that the only the top base plate moves during the first mode, however both the top and middle base plates move as one mass for the second mode.

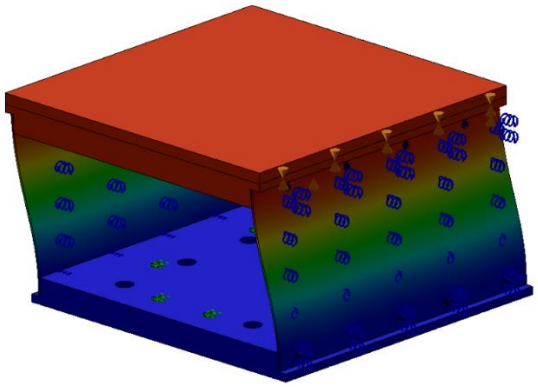
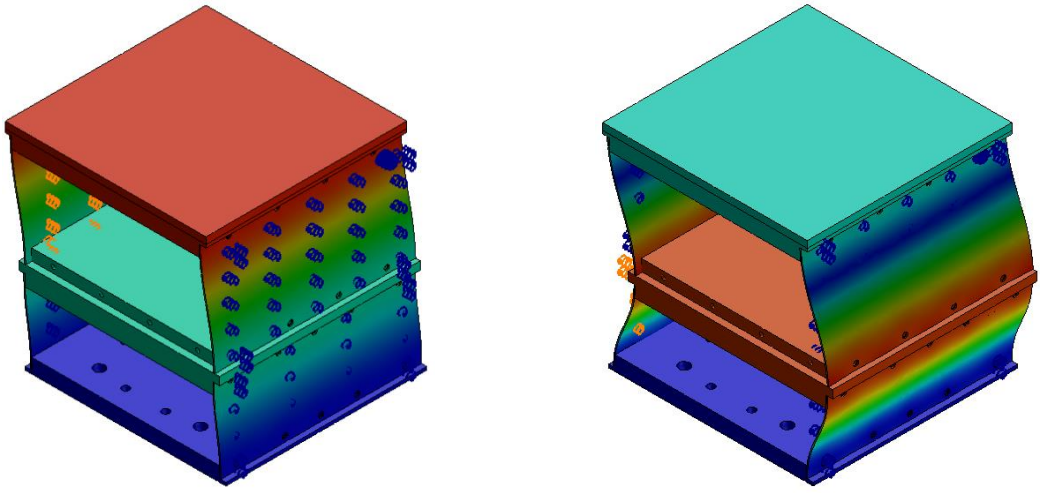
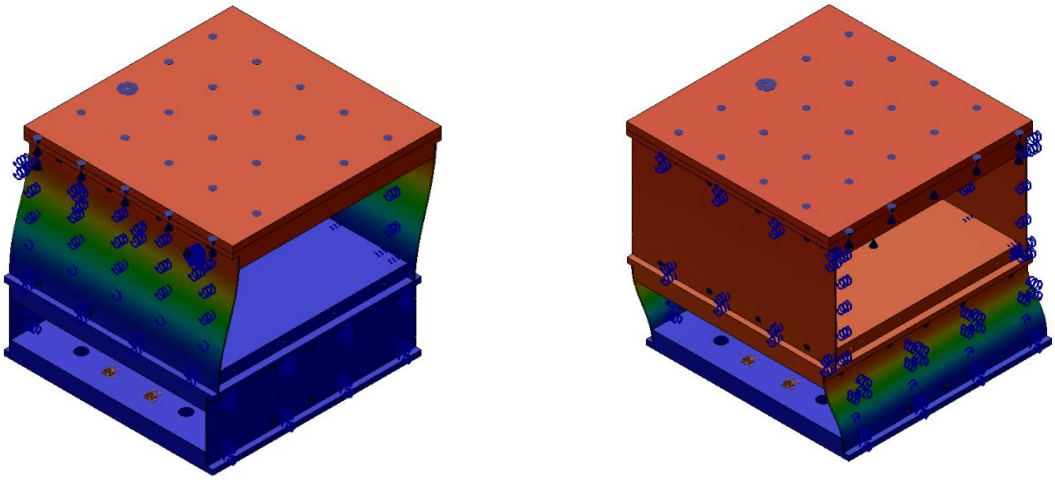


Figure 6-4 Simulated deformation for single mode flexure



(a) (b)

Figure 6-5 Simulated deformation for two-mode flexure same direction (a) *1*st and (b) *2*nd mode



(a) (b)

Figure 6-6 Simulated deformation for two-mode flexure opposite direction (a) *1*st and (b) *2*nd mode

6.3 Verification of Design

To verify the simulation results, impact hammer testing was performed on the different configurations to determine the real frequencies of the flexure, the roving hammer method was used in these tests. The experimental setup was mounted on the PMT stage and marked with multiple points as seen in Figure 6-7, where the red boxes indicate the location of the accelerometer, the green dots show the locations of impact, and the arrows indicate the direction of impact. Note that in the two flexible modes (opposite direction) setup, there are also impact locations in the middle and bottom base plates directly below the locations of the top base plate in the y-axis, however they are not shown in the figure.

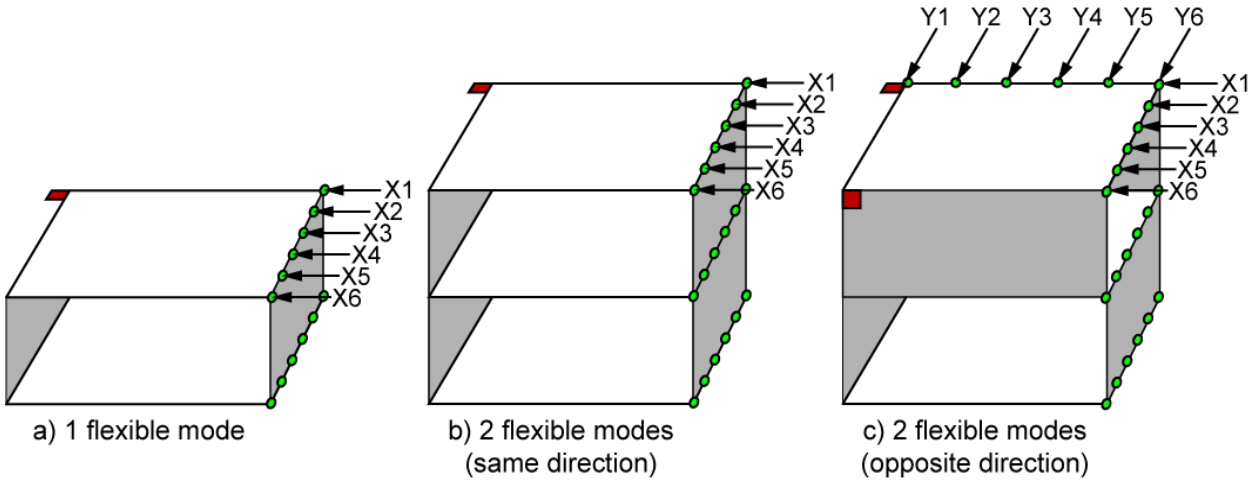


Figure 6-7 Impact points for tap testing of flexure

The peak picking method was used to determine the dynamics of the flexure. Frequency response function (FRF) measurements were taken using the software CutPRO. The complete peak picking data results can be found in 0, a summary of these results can be seen in Table 6-6.

Table 6-6 Actual modes for flexures

Frequency	One Mode	Two-Mode	
		Same Direction	Opposite Direction
1	28 Hz	23.5 Hz	29 Hz
2	-	61 Hz	62 Hz

In Table 6-7, Table 6-8 and Table 6-9, the mode shapes sketched from the peak picking method for the one mode flexure, the two-mode same direction flexure and the two-mode opposite direction flexure can be seen, respectively. The red and blue box is the outline of the flexure structure and the black box is the displacements of the mode shape (with a scaling factor).

From the simulated deformations above, the mode shape sketches below, and the modes seen in Table 6-4 and Table 6-6, it can be seen that the simulations from Solidworks have accurately predicted both the mode shapes and the frequency of the modes of the actual flexure. For the single mode flexure, the Solidworks simulation estimated the frequency would occur at 27.33 Hz and in the actual flexure, the mode was measured to be at 28 Hz. The mode shapes for the two-mode flexures have also been simulated accurately with Solidworks. During the first mode of the two-mode same direction flexure, it can be seen in both the simulation and the actual setup that both the top and middle base plates are moving in the same direction. However, in the second mode, the top plate is moving in the direction of impact while the middle plate is moving against the direction. For the two-mode opposite direction flexure, it can be seen that in both the simulation and the real flexure that only the top base plate shows any movement during the first mode. However, in the mode shape sketch of the second mode for the opposite direction flexure, the deflection is very small and cannot be seen clearly. The simulations were able to predict the vibration modes of the flexure with 95-98% accuracy, with a maximum absolute difference of 2.87 Hz.

Table 6-7 Sketches of actual modes for one mode configuration

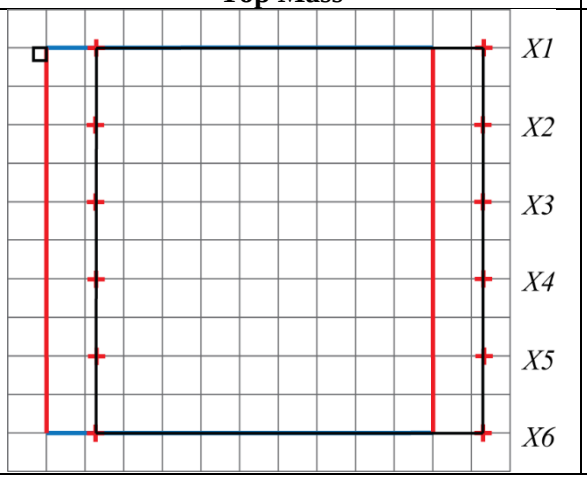
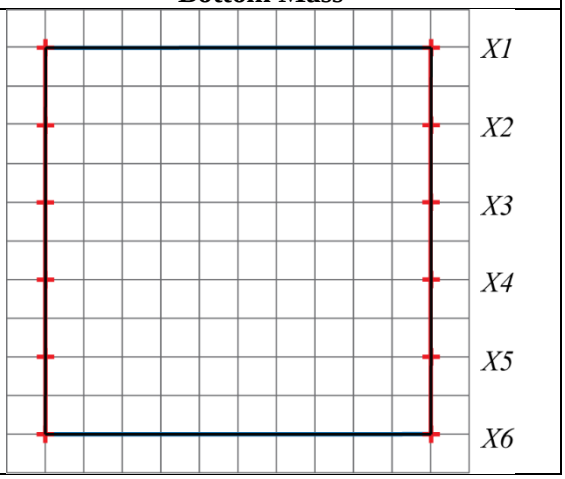
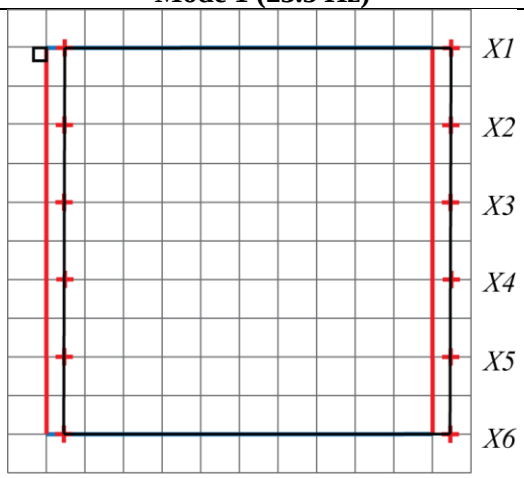
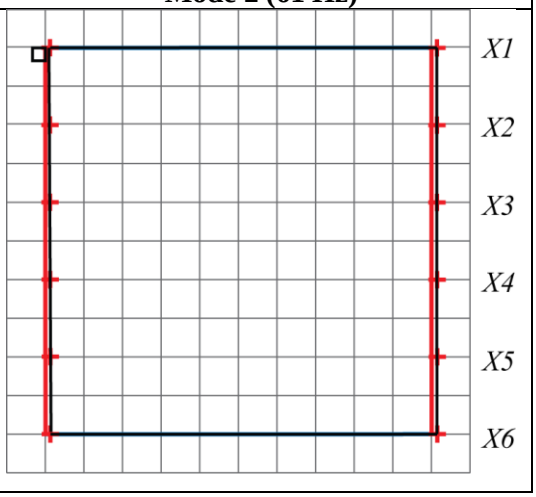
Top Mass	Bottom Mass
	

Table 6-8 Sketches of actual modes for two-mode same direction configuration

Mass	Mode 1 (23.5 Hz)	Mode 2 (61 Hz)
Top		

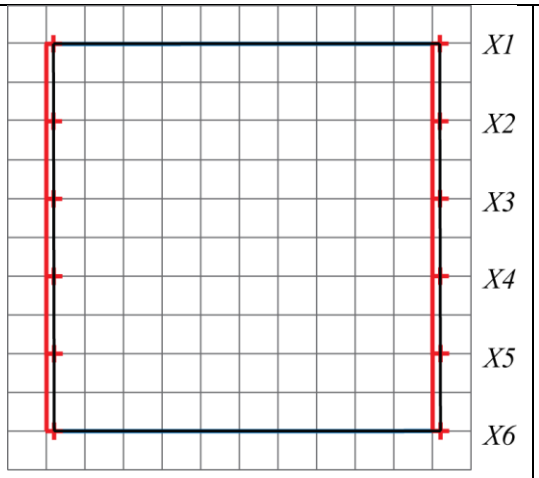
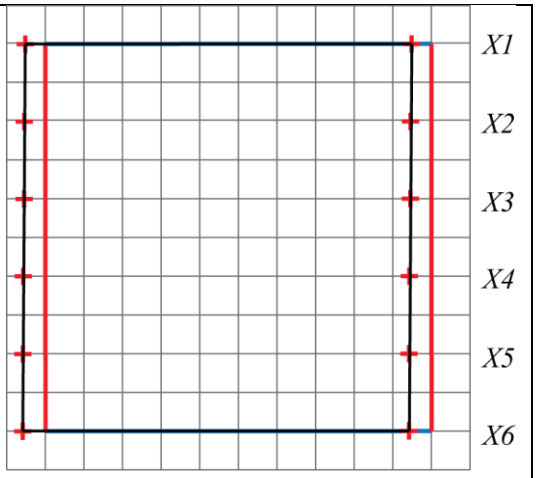
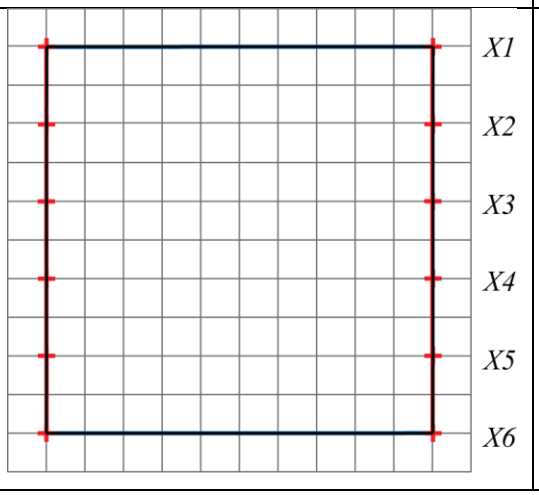
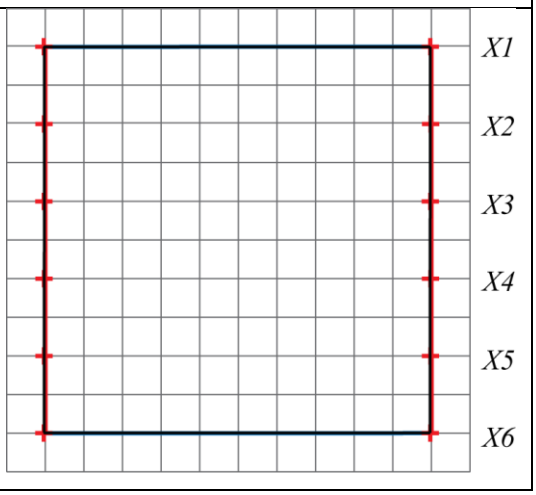
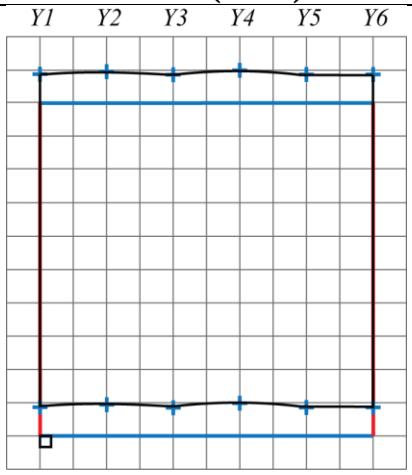
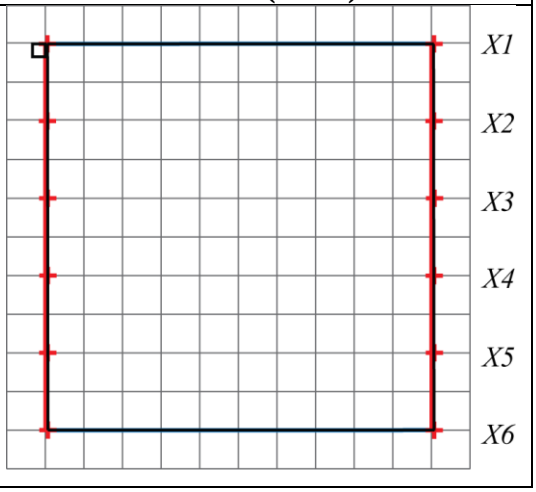
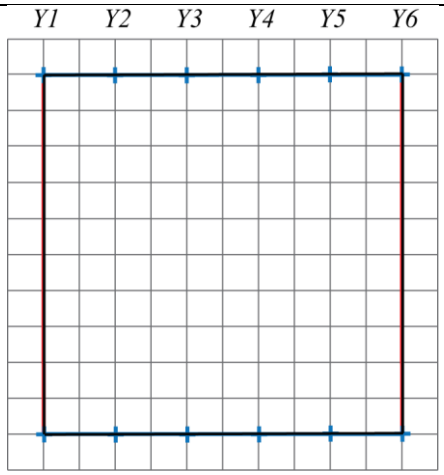
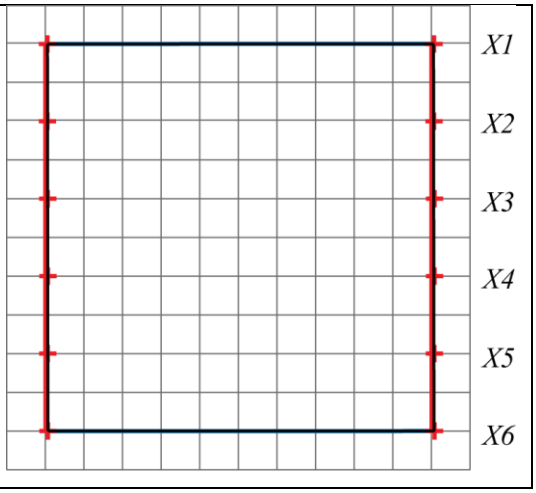
Middle		
Bottom		

Table 6-9 Sketches of actual modes for two-mode opposite direction configuration

Mass	Mode 1 (29 Hz)	Mode 2 (62 Hz)
Top		

Middle		

6.4 Conclusions

In this chapter, a flexure was designed to have different configurations and to be mountable on two different machine tools. The designs were simulated in Solidworks to determine the frequency modes and mode shapes, which are dependent on the height of the side plates. The flexure was then built based on the design and tap testing was conducted on the flexure to analyze the actual vibration modes and mode shapes. It was determined that the Solidworks simulation was able to accurately predict the actual frequency modes and mode shapes. This flexure will be used in the next chapter for experimental validation of the MIMO LTI model presented in Chapter 5 in order to predict vibration modes on a rigid machine tool.

Chapter 7
Use of Acceleration Sensors for MIMO Rapid Identification

7.1 Introduction

To verify that the model discussed in Chapter 5 can identify distinct vibration modes in machine tools, the flexure experimental setup in Chapter 6 was built and tested to show that it has distinct vibration modes. The flexure was then installed onto the Deckel Maho 80P hi-dyn five-axis machining center, which has rigid body dominant dynamics, to collect experimental data that contains one vibratory mode for verification of the MIMO LTI model.

To collect experimental data for the flexible models, measurements on the machine tool were taken using three different acquisition systems as seen in Figure 7-1. The first system is the encoder readings taken from the feed drive's controller. The second acquisition system is two 3-axis accelerometers, one mounted onto the table of the machine tool and the other mounted on the spindle head. The final acquisition system is a KGM grid encoder. The second and third acquisition systems are able to measure the true response of the machine tool at the tool-workpiece interface, which is important for more accurate predictions of the final part that will be machined.

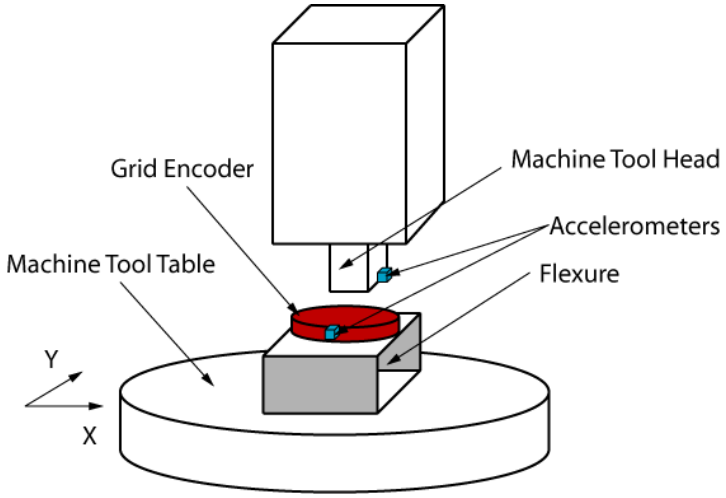


Figure 7-1 Sketch of flexible experimental setup

A KGM grid encoder is a 2D encoder that is used to calculate the contouring accuracy of a machine tool within ±2 μm [59]. The grid encoder has two main components, a grid plate which is mounted to the machine tool table and the scanning head which is placed into the spindle holder. The scanning head does

not make contact with the grid plate during measurements, instead there is a gap of 0.5±0.05 mm between the two components [59].

The purpose of this experimental setup is twofold, the first is to use the grid encoder to verify that the MIMO model can identify a vibratory model using the KGM acceleration measurements, and the second is to use accelerometer signals to identify a model. The KGM, although it is a very precise measurement tool, it is not practical to setup in a manufacturing environment as coolant or lubricant can drop onto the encoder surface and it is very time consuming to calibrate. However, accelerometers are more durable and are very quick and easy to setup onto the machine tools, therefore providing a more practical way of collecting true tool-workpiece motion data on the manufacturing floor.

7.2 Experimental Setup

The flexure was mounted onto the table of the machine tool, with sideplates facing the y direction, as seen in Figure 7-2. The motor encoder positions were collected using the internal oscilloscope of the Heidenhain TNC 430N controller at a sampling rate of 0.6 ms for 4096 samples. The KGM grid encoder relative position measurements were captured using the Heidenhain software ACCOM at a sampling rate of 0.1 ms for 60000 samples. The 6 different channels of the accelerometers ($x_{head}, y_{head}, z_{head}, x_{table}, y_{table}$, and z_{table}) were gathered using the Siemens software LMS Signature Acquisition at a sampling rate of 0.488 ms for 22528 samples.

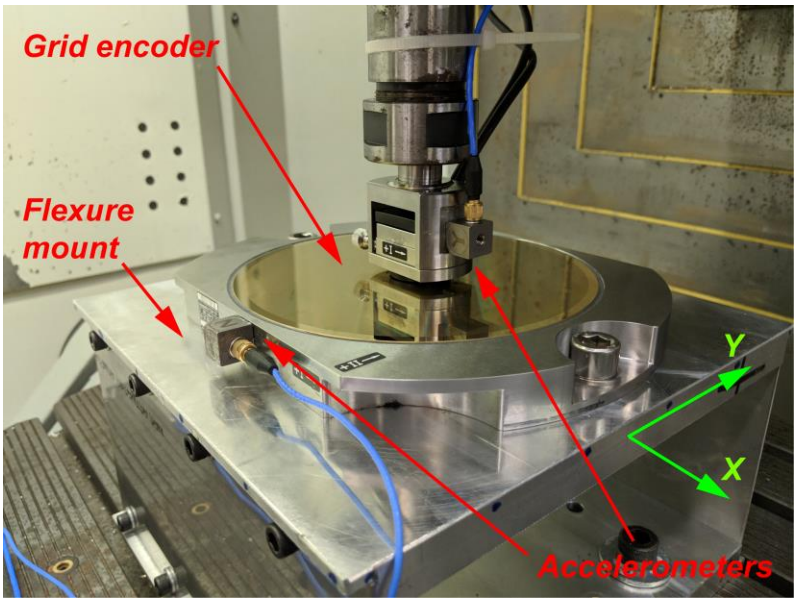


Figure 7-2 Setup for testing MIMO high-order rapid identification

The trajectories captured in these experiments only move in the x- and y-axes. The DMG machine tool head only moves in the x-direction and the machine table only moves in the y-direction. Therefore, the accelerometer mounted onto the tool head should theoretically only register accelerations in the x-axis and the accelerometer on the table should only register accelerations in the y-axis, while both accelerometers should not register any accelerations in the z-direction. However, since the axes are driven by ball screw drives, there can be a coupling effect between the axes resulting in accelerations captured in the other axes.

7.2.1 Data Processing

Since the data was collected on three separate data acquisition setups, processing of the raw data must be performed before using the MIMO LTI algorithm presented in Chapter 5. First the KGM grid encoder data and the accelerometer data was processed using the process seen in Figure 7-3. The KGM position data was down sampled to match the accelerometer data's sample time and was shifted by a small time increment. The shifted KGM data was then double differentiated to estimate the true tool-workpiece acceleration.

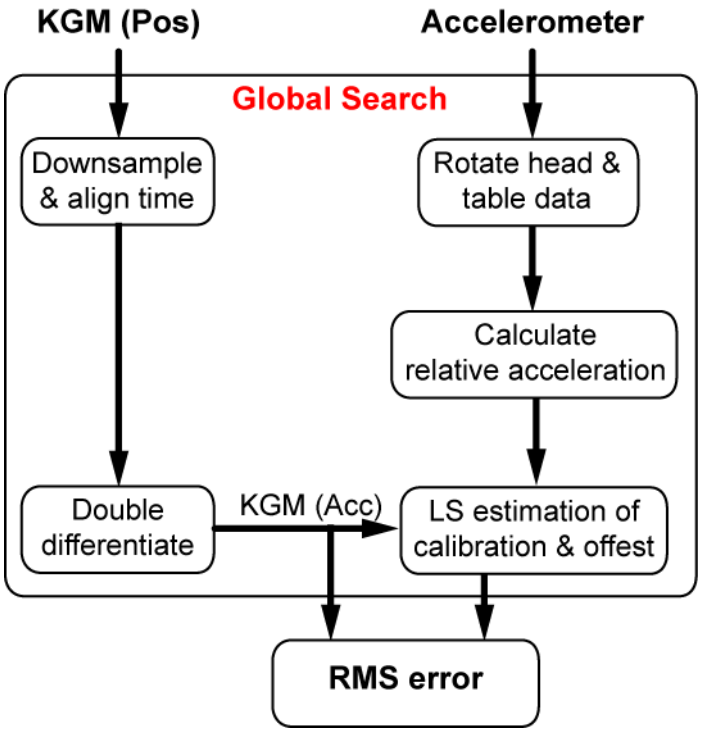


Figure 7-3 KGM and accelerometer data processing flowchart

The KGM, during installation and initialization on the machine tool, is calibrated for angular position [63]. To calibrate the angular position of the accelerometer measurements, both the head and table accelerometer

data were rotated by small angle increments using the rotation matrices in Eqn. (7.1). The diamond trajectory tests with 20 mm length sides were used to determine the rotation angles of the accelerometers since both the x- and y-axes were moving in these tests.

$$
\begin{aligned}
R_{xyz} &= R_x(\gamma) \cdot R_y(\beta) \cdot R_z(\alpha) \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos\alpha\cos\beta & -\sin\alpha\cos\beta & \sin\beta \\ \sin\alpha\cos\gamma + \cos\alpha\sin\beta\sin\gamma & \cos\alpha\cos\gamma - \sin\alpha\sin\beta\sin\gamma & -\cos\beta\sin\gamma \\ \sin\alpha\sin\gamma - \cos\alpha\sin\beta\cos\gamma & \cos\alpha\sin\gamma + \sin\alpha\sin\beta\cos\gamma & \cos\beta\cos\gamma \end{bmatrix}
\end{aligned} \tag{7.1}
$$

Then the relative acceleration was calculated by subtracting the accelerations measured at the head of the machine tool by the accelerations measured at the table. The shifted KGM acceleration is then used as the output vector in a simple LS estimation to calculate the calibration and offset factors for the relative acceleration to more closely match with the "true value". The RMS prediction error of the three axes are then calculated and added together to be used to minimize the optimization problem. MATLAB's global search function in the Global Optimization Toolbox was used for determining these parameters. The $\alpha, \beta,$ and γ rotations for the table and head accelerometers as well as the timing shifts can be seen in Table 7-1. The final results for the accelerometer and KGM alignment can be seen in Figure 7-4 and Figure 7-5 below.

Table 7-1 Accelerometer and KGM alignment parameters

	Head	**Table**
α [deg]	-2.58	10.00
β [deg]	-0.77	-7.00
γ [deg]	-0.64	0.02
Random		
KGM [sec]	2.5935	2.8477
Accelerometer [sec]	-3.3867	-3.4257
Diamond		
KGM [sec]	3.6374	
Accelerometer [sec]	-3.5797	

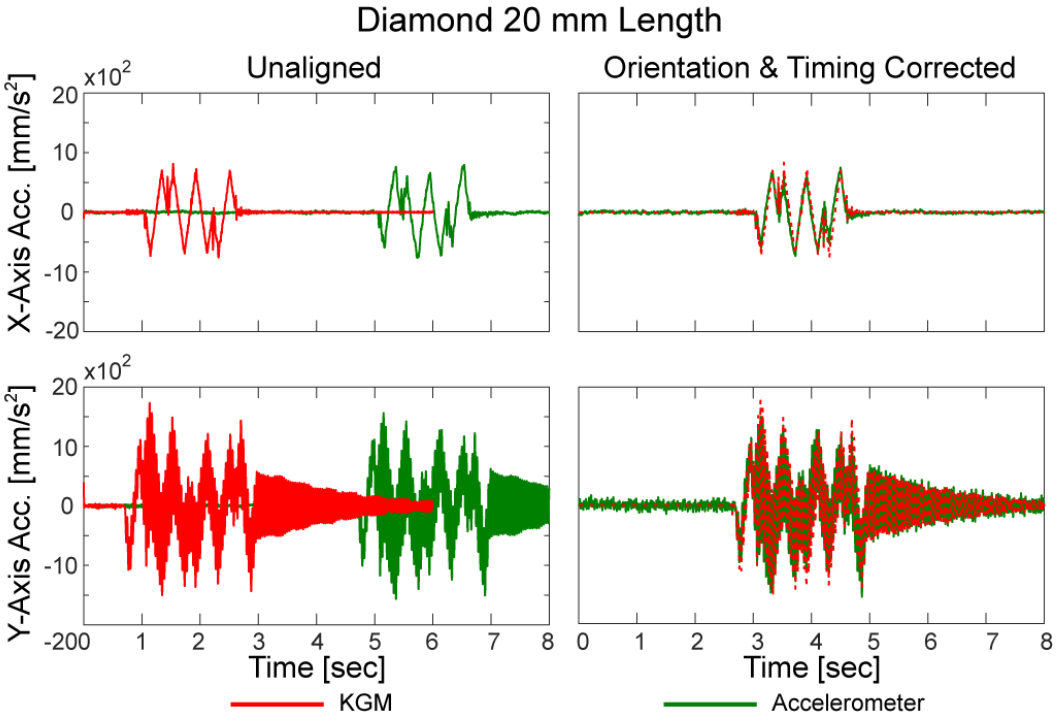


Figure 7-4 Accelerometer and KGM alignment for diamond with 20 mm length sides

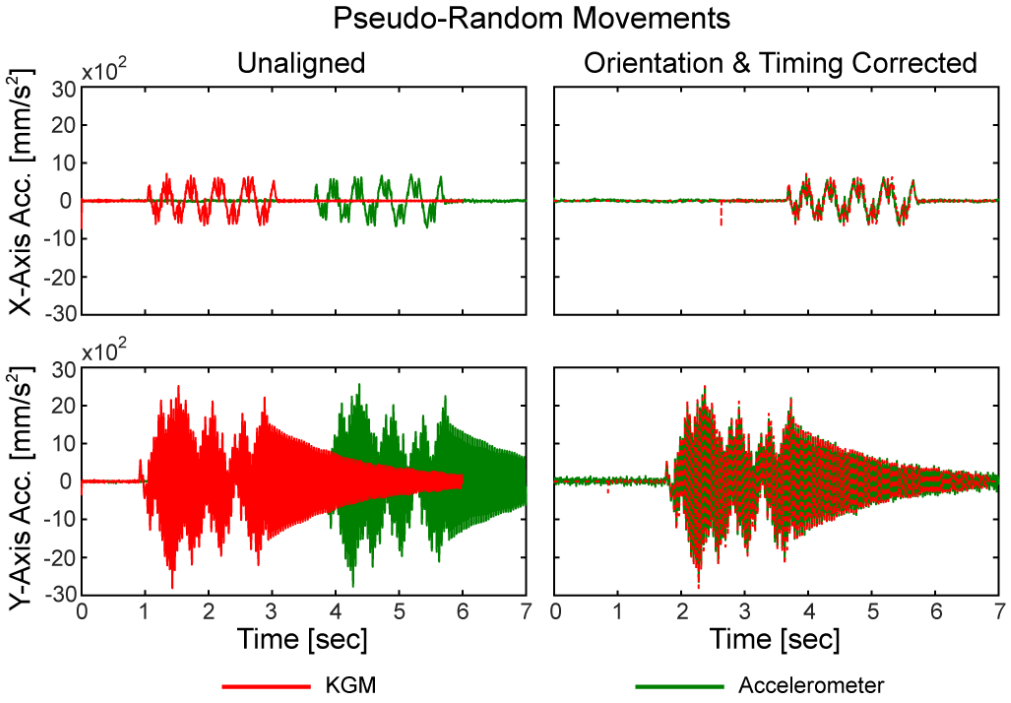


Figure 7-5 Accelerometer and KGM alignment for pseudo-random trajectory

The KGM and accelerometer data were also aligned to the Heidenhain encoder data's timing using the global search optimization function in MATLAB. The alignment results for the diamond and random trajectories can be seen in Figure 7-6 and Figure 7-7, respectively. It can be seen in Figure 7-6, that the position, tracking error and acceleration for the x-axis of the Heidenhain motor encoders and the KGM match well. However, although the accelerometer data does have a similar profile to the encoder and KGM acceleration signals, there is a slight mismatch as seen in the two bottom plots. Since the flexure is mounted in the y direction, the position, tracking error, and acceleration of the encoder and KGM for the y-axis do not match well, due to the additional vibratory dynamics. However, it can be seen that the acceleration calculated by double differentiation of the KGM position data and the accelerometer data match very well, in amplitude and timing. For the random trajectories in Figure 7-7, there seems to be less significant drift between the KGM acceleration and the accelerometer data, this could be due to the fact that these trajectories were executed one axis at a time.

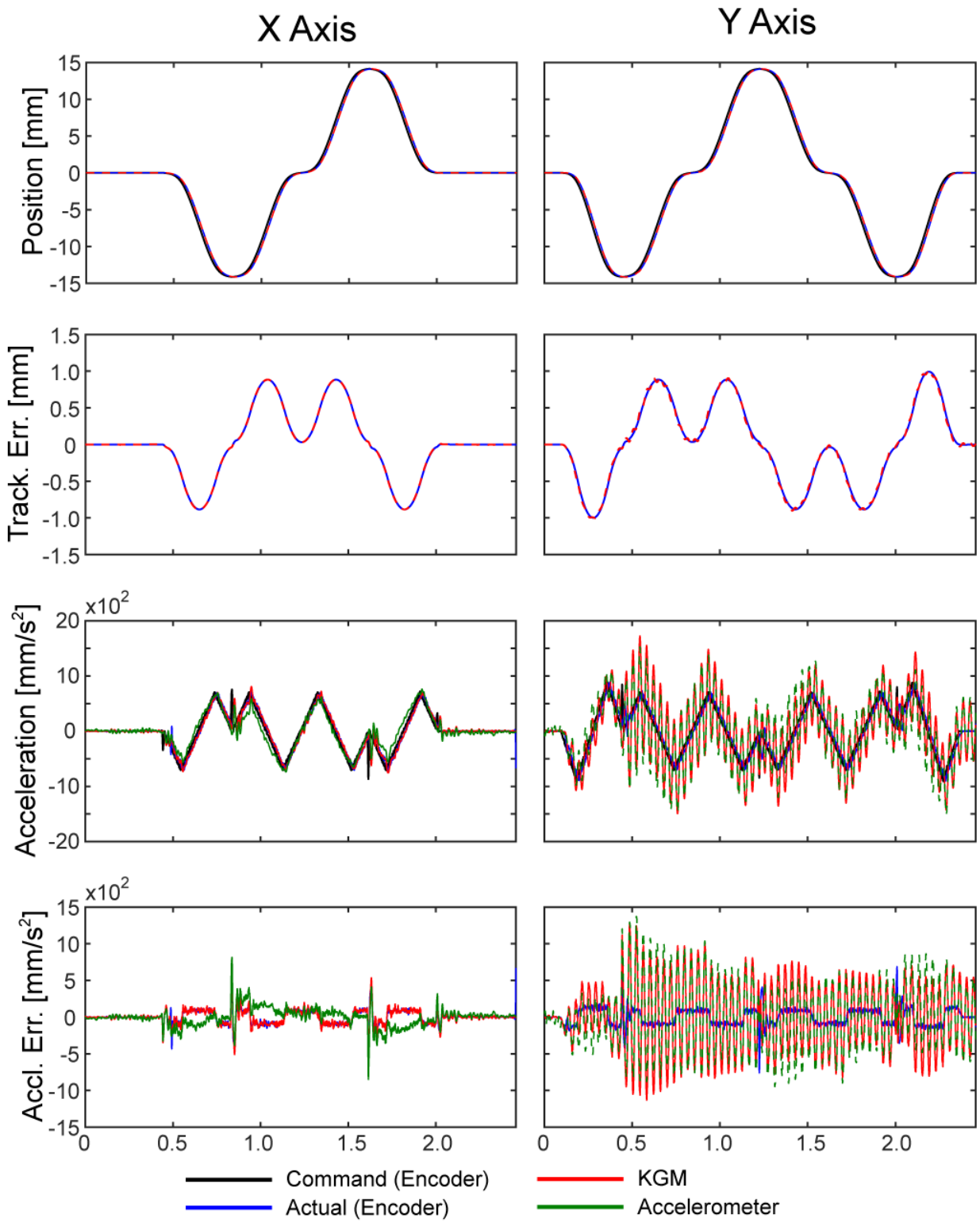


Figure 7-6 Diamond 20 mm length aligned data

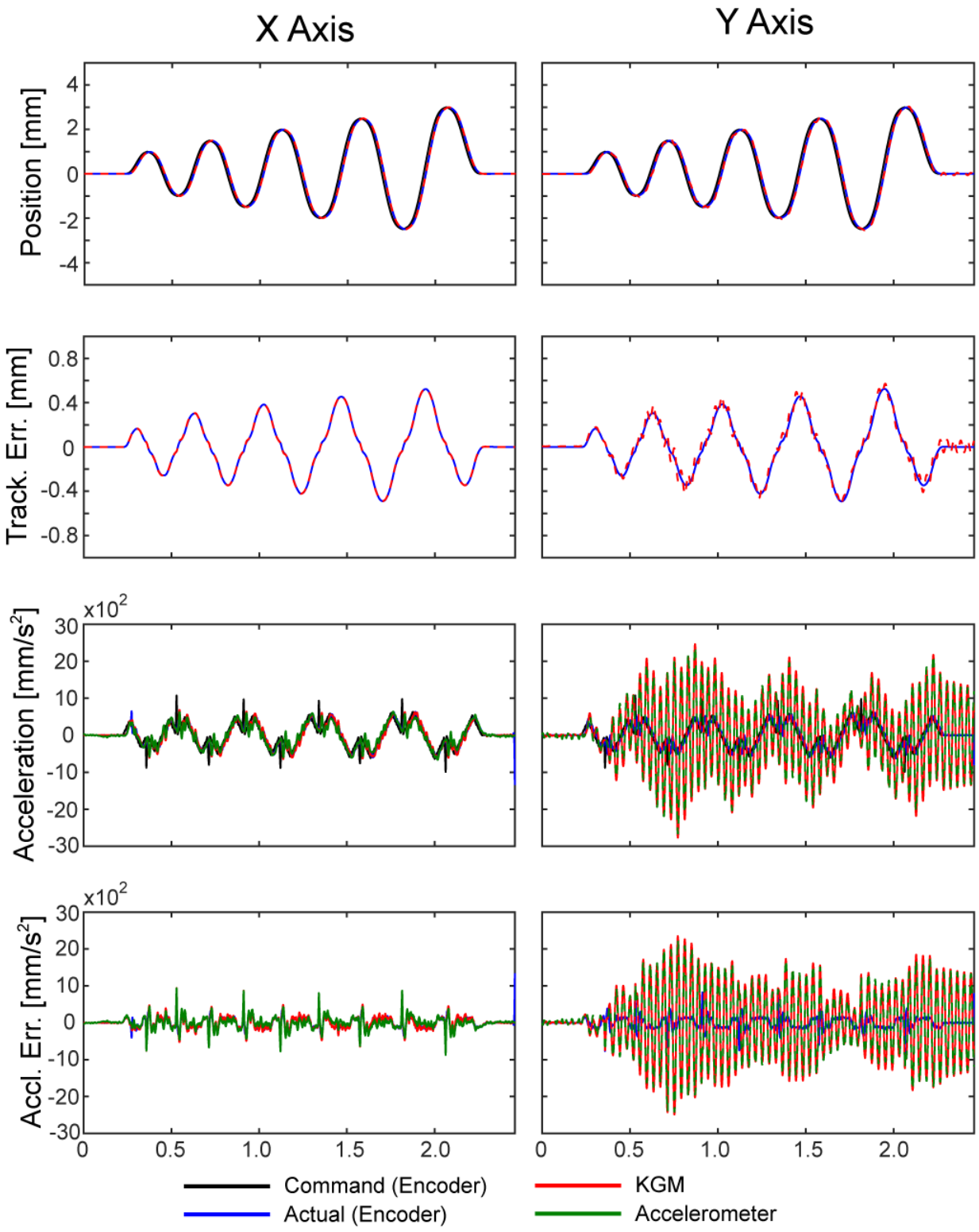


Figure 7-7 Random x and y movements - tested independently, with aligned data

7.2.2 Results

In Figure 7-8, the configuration of the identification model is shown. Both the x- and y-axes have two inputs: 1) normalized (Coulomb-type) friction and 2) the position command captured from the CNC controller, and each axis has two outputs: 1) CNC feed drive velocity response (used to estimate the friction state) and 2) the true response of the machine tool at the tool-workpiece level. Three models were identified for this experimental setup: model 1 is identified through KGM position measurements, model 2 is identified through KGM acceleration measurements and model 3 is identified through accelerometer measurements. These models will be referred to as KGM-position, KGM-acceleration, and accelerometer throughout the rest of the chapter.

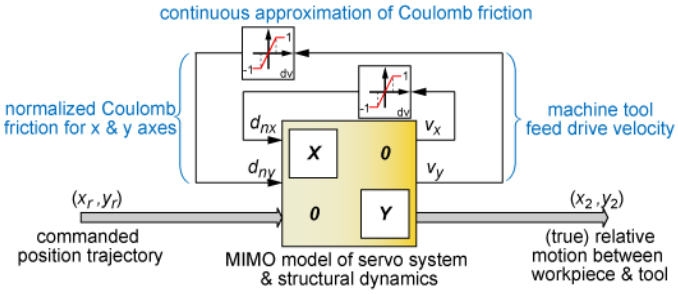


Figure 7-8 Structure for MIMO model identification on machining center

It was decided that the x-axis would have a *3*rd order rigid body model with the pole search bounds defined in Table 7-2 and that the y-axis would have a *5*th order model with the search space defined in Table 7-3. It is commonly known that a rigid body model with an integrator in the controller will have a *3*rd order transfer function [44], and that any additional flexible mode will add two extra orders per mode [64]. The maximum value for both p and $w_{k,1}$ are the Nyquist frequency of the sampling data. Since the flexure mode is known to be around 28Hz, the second frequency, $w_{k,2}$, has a search space between 20 and 40.

Table 7-2 Pole search bounds for X axis of Deckel Maho 80P and flexure setup

Parameters	Min	Max
p [Hz]	0.02	833.33
$\omega_{k,1}$ [Hz]	0.02	833.33
$\zeta_{k,1}$ []	0.01	2

Table 7-3 Pole search bounds for Y axis of Deckel Maho 80P and flexure setup

Parameters	Min	Max
p [Hz]	0.02	833.33
$\omega_{k,1}$ [Hz]	0.02	833.33
$\zeta_{k,1}$ []	0.01	4
$\omega_{k,2}$ [Hz]	20	40
$\zeta_{k,2}$ []	0.001	0.5

To identify parameters for the experimental setup, pseudo-random movements were used to train the model. The identified parameters for both the x- and y-axes for the KGM-position, KGM-acceleration and accelerometer are shown in Table 7-4. It should be noted that the local minima were stored during the optimization process, and all local minima were tested to ensure the best position response. All three models have similar dynamics in the x-axis, the identified complex pole frequencies are very similar, with natural frequencies identified at 25.25 Hz, 28.72 Hz, and 28.64 Hz for the KGM-position, KGM-acceleration, and accelerometer models respectively. However the real poles are quite different, with identified parameters at 14.22, 25.68, and 98.88 Hz for the KGM-position, KGM-acceleration, and accelerometer configurations respectively.

It can be observed that all of the models were able to capture the flexible dynamics of the flexure in the y-axis. All three models were able to identify the natural frequency at 25.76 Hz, with very similar damping ratios of 0.0052 [] for the KGM-position model, 0.0055 [] for the KGM-acceleration model and 0.0057 [] for the accelerometer model. Therefore it has been verified that the MIMO LTI algorithm can identify flexible modes in machine tools.

Again the real pole and first natural frequency are quite different between the three models, however these are not the dominate dynamics in the y-axis, therefore these contributions would be more difficult to capture. Although both KGM models have real and complex poles that are similar. The frequencies of the higher order zeros of the tracking transfer function for the y-axis are also very similar, 31.65, 36.43 and 35.27 Hz for the KGM-position, KGM-acceleration and accelerometer models respectively.

Table 7-4 Identified parameters for X- and Y- axes of Deckel Maho 80P and flexure setup

Param.	KGM - Position				KGM - Acceleration				Accelerometer			
	X Axis		Y Axis		X Axis		Y Axis		X Axis		Y Axis	
p_r	14.22		17.58		25.68		14.18		98.88		368.65	
$w_{k,1}$	25.25		33.18		28.72		25.15		28.64		74.58	
$\zeta_{k,1}$	0.24		0.24		0.36		0.17		0.85		1.40	
$w_{k,2}$	-		25.76		-		25.76		-		25.76	
$\zeta_{k,2}$	-		0.0052		-		0.0055		-		0.0057	
Track. TF Zeros	**Freq. [Hz]**	**Damp. []**	**Freq. [Hz]**	**Damp. []**	**Freq. [Hz]**	**Damp. []**	**Freq. [Hz]**	**Damp. []**	**Freq. [Hz]**	**Damp. []**	**Freq. [Hz]**	**Damp. []**
z_1	0.00	-1.00	0.00	-1.00	0.00	-1.00	0.00	-1.00	0.00	-1.00	0.00	-1.00
z_2	26.29	0.72	31.65	-0.12	0.00	-1.00	0.00	-1.00	0.00	-1.00	0.00	-1.00
z_3	26.29	0.72	31.65	-0.12	48.89	-1.00	56.32	-1.00	172.59	-1.00	2220.94	-1.00
z_4	-	-	26.57	0.72	-	-	36.43	0.41	-	-	35.27	-0.68
z_5	-	-	26.57	0.72	-	-	36.43	0.41	-	-	35.27	-0.68

In Figure 7-9, Figure 7-10, and Figure 7-11, the commanded, measured, and KGM and accelerometer simulated axes positions are presented for the pseudo-random, diamond and circle trajectories respectively. The contouring error and prediction error of the diamond and circle trajectories can be seen in Figure 7-12. In Table 7-5, the tracking and contouring prediction errors can be found for all three trajectories.

Overall it can be seen that the KGM-position identified model is has the best prediction characteristics. For the diamond trajectory, the RMS tracking error prediction is 771-953% closer for the x-axis and 85-123% closer for the y-axis, with an improvement of 349-370% in the RMS contour error prediction when compared to the accelerometer and KGM-acceleration models The same can be seen for the circle trajectory where the maximum tracking error discrepancy is 676-828% closer for the x-axis and 56-83% closer for the y-axis. The RMS contour prediction error is improved by 233-245% and the maximum contour prediction error is improved by 148-164% when comparing to the accelerometer and KGM-acceleration models. There is also similar improvements when using the KGM-position model over the KGM-acceleration and accelerometer models in the pseudo-random trajectories, with an improvement between 19-758% for the RMS tracking error prediction and 28-500% for the maximum prediction error. In the rest of the result section, the KGM-acceleration and accelerometer models will be compared to determine if a model built from acceleration measurements is feasible.

The reconstructed position response of the pseudo-random trajectory can be seen in Figure 7-9. It can be seen in the figure that the MIMO LTI model is able to predict the position response from the acceleration

input in the y-axis very well. However, there is a significant discrepancy for both the KGM and accelerometer reconstruction in the x-axis. One possible reason for this discrepancy is the alignment of the data collected from the three different data acquisition systems. Since the y-axis has a more distinct sinusoidal signal due to the flexure, it was easier to align the data, however the x-axis has a smoother movement, therefore there could be a slight time shift in the data that affected the identification of parameters. It is recommended in future testing to include a rapid movement before the execution of the NC code to more easily align the data. From Table 7-5, it can be seen that the accelerometer model overall provides a better prediction of both the RMS and maximum tracking prediction error for both the x- and y-axes. The accelerometer model is able to predict the tracking error with 33 and 16% improvement in RMS tracking error prediction, as well as 51% and 11% improvement in predicting the maximum value of the tracking error for the x- and y-axes respectively.

Table 7-5 Tracking and contouring prediction for Deckel Maho 80P and flexure setup

Discrepancy in Servo Error Prediction [µm]	Diamond 20			Circle			Varying Dist. Linear Interpolation		
	KGM: Pos	KGM: Acc	Acc	KGM: Pos	KGM: Acc	Acc	KGM: Pos	KGM: Acc	Acc
X axis track. error: RMS (MAX)	**2.64** **(8.56)**	27.84 (56.63)	23.02 (47.87)	**2.73** **(11.46)**	43.57 (106.33)	36.37 (88.88)	**1.88** **(6.48)**	16.14 (38.92)	12.12 (25.70)
Y axis track. error: RMS (MAX)	**5.57** **(17.19)**	12.44 (30.34)	10.31 (25.03)	**8.77** **(23.43)**	16.37 (42.79)	13.07 (36.60)	**7.26** **(16.93)**	10.04 (24.19)	8.66 (21.71)
Contour error: RMS (MAX)	**3.28** **(15.94)**	15.40 (42.74)	14.72 (42.31)	**5.26** **(23.17)**	18.15 (61.14)	17.52 (57.54)	-	-	-

The identified transfer functions from the pseudo-random trajectories were then used for verification for the diamond and circle trajectories. Comparing the prediction errors for the diamond trajectories, it can be seen that the accelerometer has improved prediction characteristics over the KGM model for both the x- and y-axes. The RMS tracking error prediction for the x-axis decreased from 27.84 to 23.02 µm, a 21% improvement, and decreased from 12.44 to 10.31 µm in the y-axis, a 21% improvement. The maximum tracking error prediction also is improved while using the accelerometer data, by 18 and 21% for the x- and y-axes respectively. The RMS contouring error prediction has a 5% improvement when using the accelerometer model, and the maximum contour error prediction increases by 1% when using the same model. The circle trajectory results also have the same trend, where the RMS tracking error prediction has improved by 20 and 25% by using the accelerometer model and the maximum prediction error has improved

by 20 and 17% for the x- and y-axes respectively. Again, the RMS contouring error has improved by 4% for the accelerometer model and the maximum contouring prediction error has increased by 6% over the KGM model.

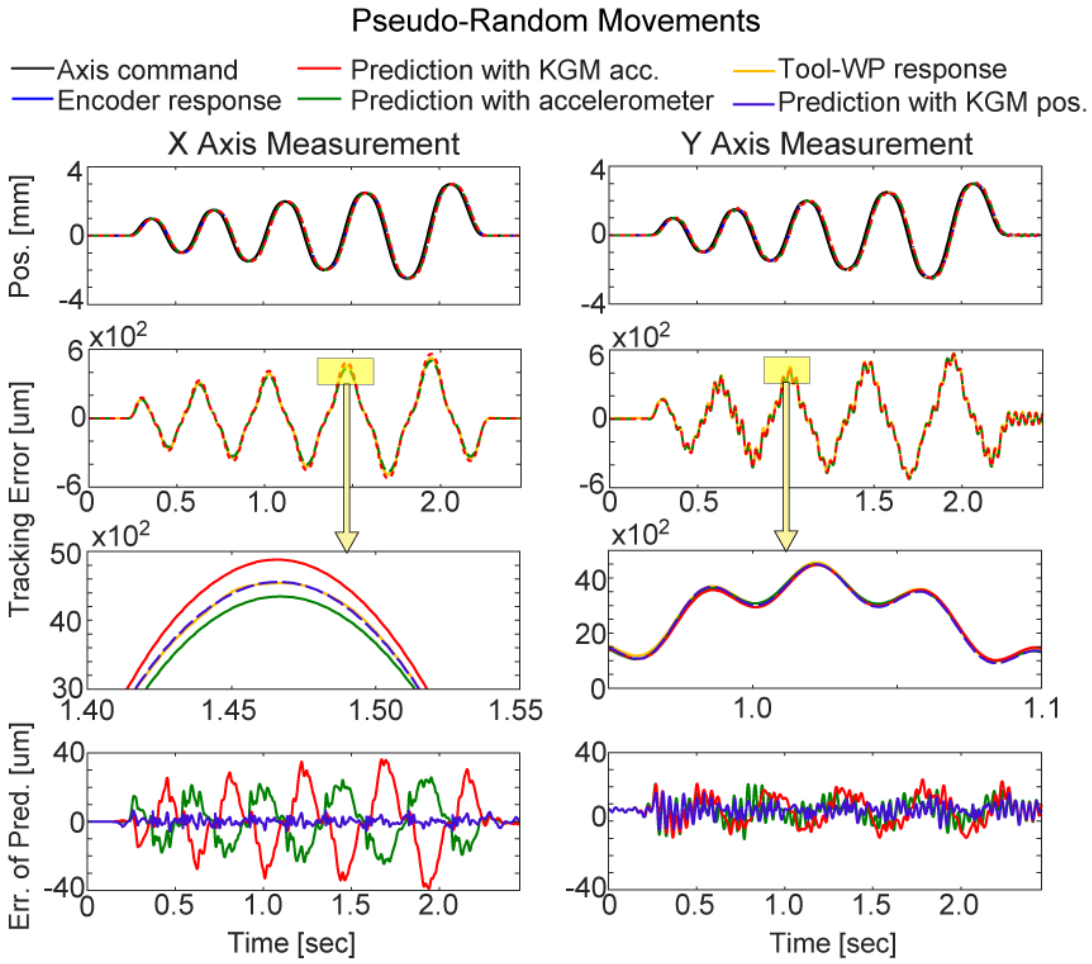


Figure 7-9 Actual and predicted servo performance for a varying distance linear interpolation toolpath using MIMO LTI Rapid Identification

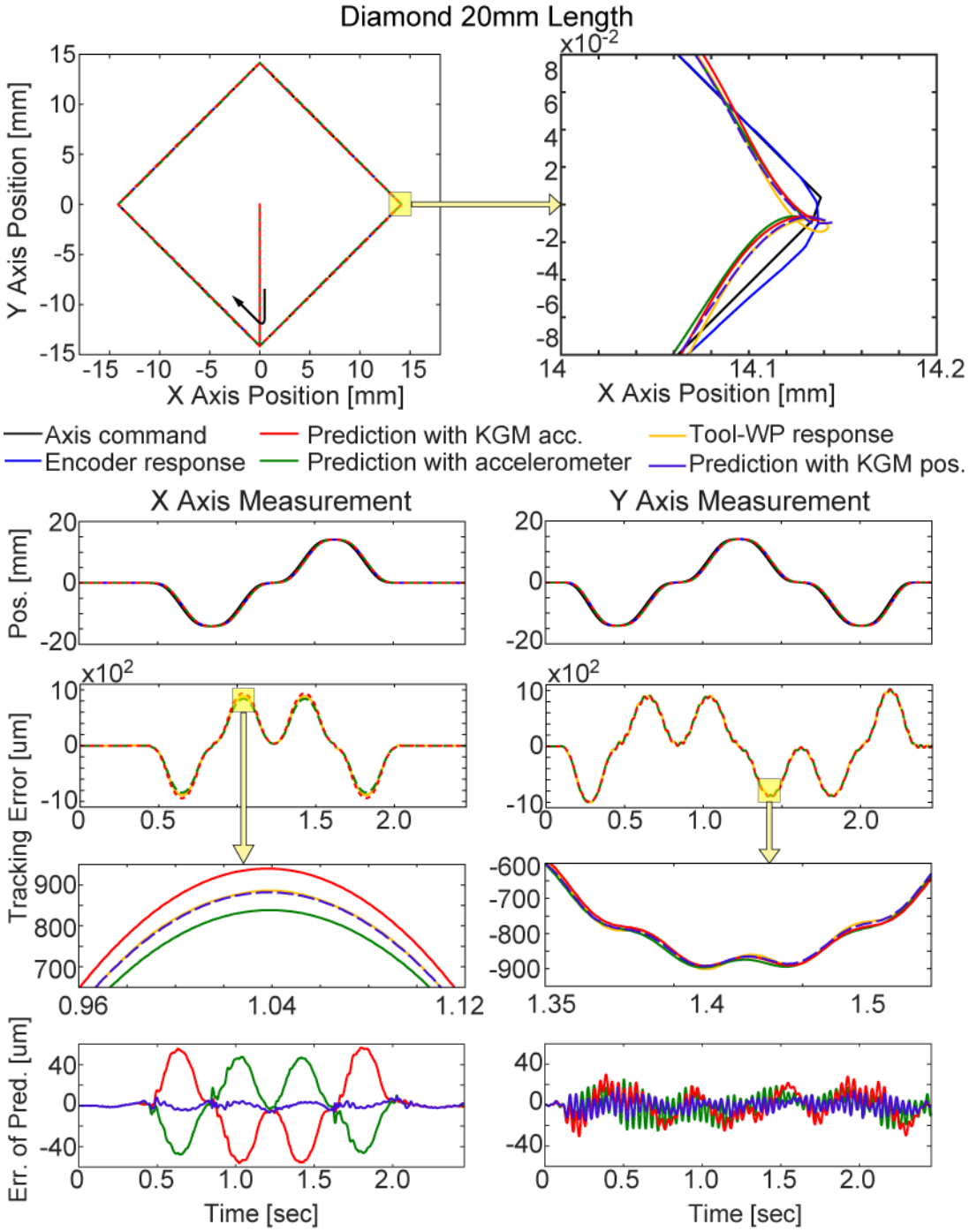


Figure 7-10 Actual and predicted servo performance for a 20mm length diamond toolpath using MIMO LTI Rapid Identification

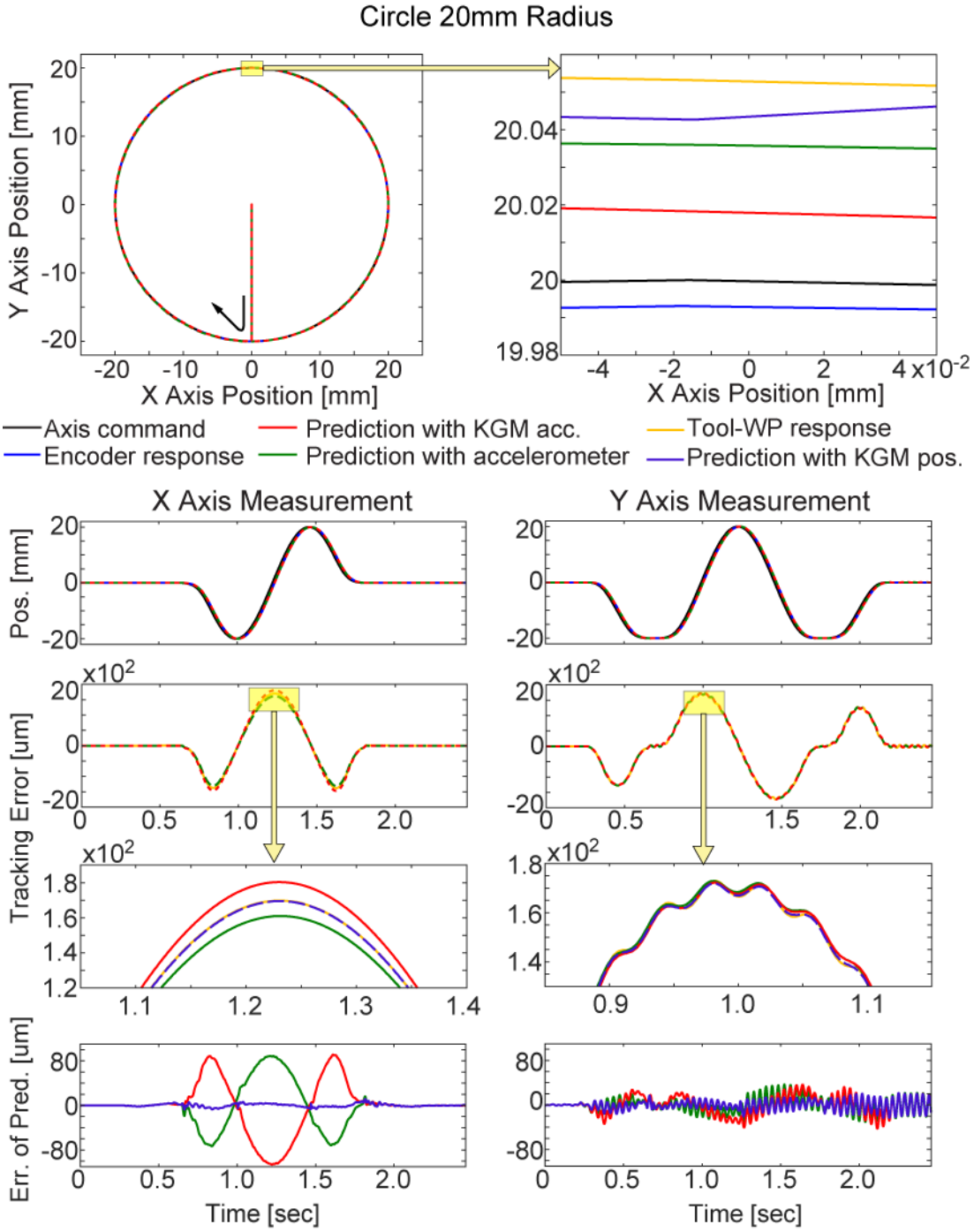


Figure 7-11 Actual and predicted servo performance for a 20mm radius circle toolpath using MIMO LTI Rapid Identification

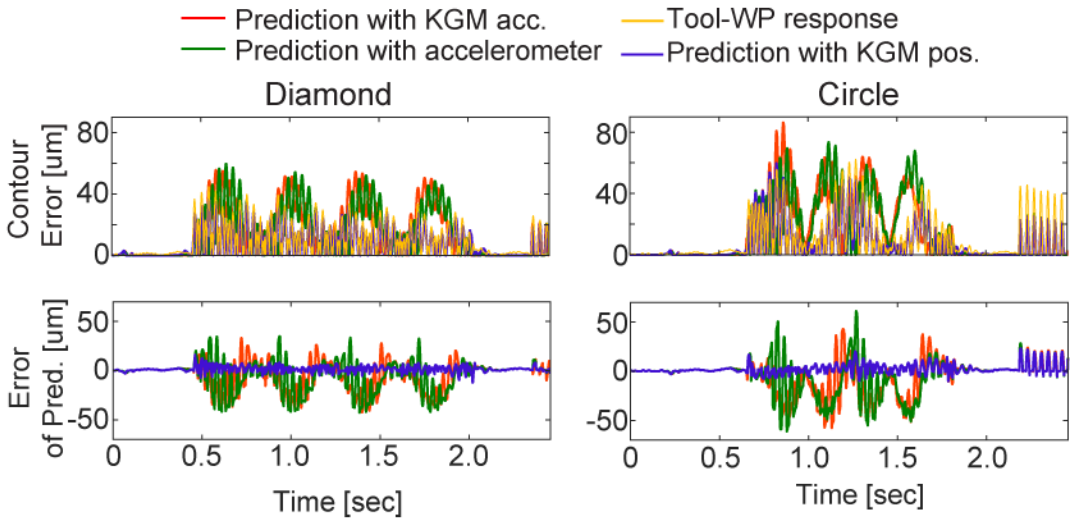


Figure 7-12 Contour errors for diamond and circle toolpaths on Deckel Maho 80P and flexure setup

7.3 Conclusions

In this chapter, the use of accelerometer sensors for identifying MIMO models, presented in Chapter 5, was discussed. To validate that the MIMO model is able to capture vibration modes, a 2D KGM grid encoder was also used to identify a model. The flexure design in Chapter 6 was mounted on the five-axis Deckel Maho machine tool, to induce a flexible mode on the rigid machine tool. The data for this experimental setup was collected on three different data acquisition systems, therefore angle and time calibration was required before the experimental data could be used for model identification. A pseudo-random trajectory was executed on the five-axis machining center and was used for identifying the parameters for all the models: position measurements from LGM, acceleration measurements from KGM and accelerometer models. The MIMO algorithm was able to capture a vibratory mode at 25.76 Hz for all the models in the y-axis and were able to reconstruct the position response accurately from the both the position and acceleration data. However, the x-axis reconstruction has a higher discrepancy for the models identified using acceleration data, which could be caused by time alignment issues from the pre-processing step. Overall, the accelerometer model was able to estimate the RMS tracking prediction error with 16-33% improvement, and an improvement of 11-51% for the maximum tracking prediction error. There was also a 4-5% improvement in RMS contour error prediction when using the accelerometer model, and there was an improvement in maximum contour error prediction, about 1 to 6%.

Chapter 8
Conclusion and Future Work

8.1 Conclusions

In this thesis three methods for identifying digital twins of feed drives for virtual process planning are presented with minimal intervention to the machine tool. The quasi-static method is a quick and simple model that can be used for machine tools that are known to have decoupled and rigid body dynamics. The feed drive's natural response is not taken into account in this model and it is able to predict the tracking response from the time derivatives of the position commands collected from the CNC controller. This method was experimentally validated using data collected from a Siemens 840D controller on a laser drilling machine tool. This method was able to predict the RMS tracking error within 2.62 to 11.91 μm and the maximim tracking error within 11.40 to 69.05 μm.

The second method presented in this thesis is the SISO rigid body model, which provided improvements from the previous rapid identification method proposed in literature, such as better convergence characteristics and the elimination of the use of derivatives of measured signals during estimation. The two methods were compared using experimentally gathered data on a five-axis Deckel Maho machining center from the Heidenhain TNC 430N controller. The RMS tracking prediction error has improved 50-76% and prediction of the maximum values has also improved by 16-50%. The contour prediction error has also improved by using the new method, between 22-35%. A set of GUIs was developed for this algorithm to be delivered to P&WC to allow the process planners and engineers to use this method to simulate the tracking and contouring capabilities of new trajectories.

The final algorithm discussed in this thesis is the MIMO model, which is an extension of the new SISO method. This method provides the user the capability of determining transfer functions between several different inputs and outputs, which is of interest to create higher fidelity models of machine tools. Using this algorithm, flexible modes of the machine tool or process can also be identified. This method was tested on two different testbeds, a five axis milling machine with a FANUC controller and the Deckel Maho with a flexure to induce a flexible mode. For the first experimental setup, three different models were considered, a decoupled *3*rd order model and two coupled *6*th order model, one with a smaller search space. It was observed that overall, the *6*th order model with the more general search space was able to more closely predict the tracking and contouring response of the machine tool, between 2-19% RMS prediction error and maximum prediction error.

A flexure was designed and constructed to induce a repeatable and safe flexible mode on a rigid machine tool. The flexure was designed to be mountable on two different feed drives and was also designed to be reconfigurable (one-mode, two-mode same direction, and two-mode opposite direction). Simulations were done in Solidworks to determine the height of the sideplates to get the desired flexible modes. Tap testing was done after the flexure was built to verify that the simulations were correct. The simulations were able to predict the correct mode shapes and were able to predict the frequencies with 95-98% accuracy.

This flexure was installed on the five-axis Deckel Maho along with a 2D KGM grid encoder and two 3-axis accelerometers. Three models were tested on this machine tool, a model identified from the position measurements from the KGM, a model identified from the acceleration estimated from the KGM and a model identified from the accelerometer signals. Since data was collected using three different acquisition systems, time and angle alignment was done before the data was used for identification purposes. All three models were able to identify a flexible mode at 25.75Hz, however, overall the model identified using the KGM position measurements had the best prediction characteristics. The accelerometer identified model was able to more closely predict the axis positioning of this experimental setup, with a 16-33% improvement in predicting the RMS tracking error and 11-51% improvement in predicting the maximum tracking error over the model identified with KGM acceleration measurements. The RMS contouring error prediction has also improved by 4-5% and the maximum contour error prediction improved by 1-6% between the accelerometer and the KGM acceleration models.

8.2 Future Work

Future improvements to this work include conducting more experiments using the flexure, KGM, and accelerometer setup to determine if the large discrepancy in the x-axis identification can be reduced. This may be done by introducing a quick motion at the beginning of the NC code, as to provide a marker for timing alignment purposes. Further research into using different types of friction models with the MIMO algorithm should also be considered, this can improve the overall accuracy of the tracking error prediction, specifically during velocity reversals in the trajectory. This research has been commenced in the lab [65].

Due to time limitations, no part has been machined on the experimental setup with the induced flexible mode. In future work, a part will be made while the flexure is attached to the Deckel Maho, which will then be measured on a coordinate measuring machine (CMM) to verify the contouring error of the part, to validate the simulated true tool-workpiece motion.

Bibliography

[1] Siemens, "Benefit from the advantages of the digital transformation," [Online]. Available: https://www.siemens.com/global/en/home/company/topic-areas/future-of-manufacturing/digital-enterprise.html. [Accessed 24 July 2018].

[2] K. Erkorkmaz, Y. Altintas and C. Yeung, "Virtual Computer Numerical Control System," *Annals of CIRP*, vol. 55, pp. 399-402, 2006.

[3] K. Erkorkmaz and W. Wong, "Rapid identification technique for virtual CNC drives," *International Journal of Machine Tools & Manufacture*, vol. 47, pp. 1381-1392, 2007.

[4] MasterCAM, "CAD Solutions," 2018. [Online]. Available: https://www.mastercam.com/en-us/Solutions/CAD-Solutions. [Accessed 10 May 2018].

[5] 3D Systems Inc., "Why choose GibbsCAM," 2018. [Online]. Available: https://www.3dsystems.com/software/gibbscam/why-gibbscam. [Accessed 10 May 2018].

[6] Autodesk, "Compare Fusion 360 Features," 2018. [Online]. Available: https://www.autodesk.com/products/fusion-360/compare#engineer. [Accessed 10 May 2018].

[7] CGTech, "CNC Machine Probing: Vericut Module," 2018. [Online]. Available: http://www.cgtech.com/products/about-vericut/module-cnc-machine-probing/. [Accessed 10 May 2018].

[8] CGTech, "CNC Machine Simulation: Vericut Module," 2018. [Online]. Available: http://www.cgtech.com/products/about-vericut/machine-simulation/. [Accessed 10 May 2018].

[9] C.-H. Yeung, Y. Altintas and K. Erkorkmaz, "Virtual CNC system. Part I. System architecture," *International Journal of Machine Tools & Manufacture*, vol. 46, no. 10, pp. 1107-1123, 2006.

[10] Manufacturing Automation Labratories, "Virtual CNC," 2018. [Online]. Available: https://www.malinc.com/products/virtual-cnc/#overview. [Accessed 10 May 2018].

[11] Siemens, "NX CAM - Siemens Sinumerik Virtual NC Controller Kernel," 2011. [Online]. Available: https://www.plm.automation.siemens.com/en_us/Images/14541_tcm1023-67384.pdf. [Accessed 11 May 2018].

[12] K. Erkorkmaz and Y. Altintas, "High speed CNC system design. Part III: high speed tracking and contouring of feed drives," *International Journal of Machine Tools and Manufacture*, vol. 41, pp. 1637-1658, 2001.

[13] P. Boucher, D. Dumur and P. Rodriguez, "Robustification of CNC Controllers for Machine Tools Motor Drives," *CIRP Annals - Manufacturing Technolofy*, vol. 29, no. 6, pp. 1686-1690, 1995.

[14] Y. Altintas, K. Erkorkmaz and W. Zhu, "Sliding Mode Controller Design for High Speed Feed Drives," *CIRP Annals - Manufacturing Technology*, vol. 49, no. 1, pp. 265-270, 2000.

[15] Y. Altintas and C. Okwudire, "Dynamic stiffness enhancement of direct-driven machine tools using sliding mode control with disturbance recovery," *CIRP Annals - Manufacturing Techology*, vol. 58, no. 1, pp. 335-338, 2009.

[16] B. Sencer and E. Shamoto, "Effective torque ripple compensation in feed drive systems based on the adaptive sliding-mode controller," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 6, pp. 1764-1772, 2014.

[17] R. Ramesh, M. Mannan and A. Poo, "Tracking and contour error control in CNC servo systems," *International Journal of Machine Tools and Manufacture,* vol. 45, no. 3, pp. 301-326, 2005.

[18] A.-N. Poo, J. G. Bollinger and G. W. Younkin, "Dynamic Errors in Type 1 Contouring Systems," *IEEE Transactions on Industry Applications,* vol. 8, no. 4, pp. 477-484, 1972.

[19] Y. Koren, Computer Control of Manufacturing Systems, New York: McGraw-Hill, 1983.

[20] K. Erkorkmaz and Y. Altintas, "High speed CNC system design. Part II: modeling and identification of feed drives," *International Journal of Machine Tools and Manufacture,* vol. 41, no. 10, pp. 1487-1509, 2001.

[21] D. A. Vicente, R. L. Hecker, F. J. Villegas and G. M. Flores, "Modeling and vibration mode analysis of a ball screw drive," *International Journal of Advanced Manufacturing Technology,* vol. 58, no. 1, pp. 257-265, 2012.

[22] C. Okwudire and Y. Altintas, "Minimum Tracking Error Control of Flexible Ball Screw Drives Using a Discrete-Time Sliding Mode Controller," *Journal of Dynamic Systems, Measurements, and Control,* vol. 131, no. 5, 2009.

[23] S. Frey, A. Dadalau and A. Verl, "Expedient modeling of ball screw feed drives," *Production Engineering,* vol. 6, no. 2, pp. 205-211, 2012.

[24] K. Erkorkmaz and A. Kamalzadeh, "High Bandwidth Control of Ball Screw Drives," *CIRP Annals,* vol. 55, no. 1, pp. 393-398, 2006.

[25] U. Forssell and L. Ljung, "Closed-loop identification revisited," *Automatica,* vol. 35, no. 7, pp. 1215-1241, 1999.

[26] S. J. Qin, "An overview of subspace identification," *Computers and Chemical Engineering,* vol. 30, pp. 1502-1513, 2006.

[27] I. Gustavsson, L. Ljung and T. Soderstrom, "Identification of Prcoesses in Closed Loop - Identifiability and Accuracy Aspects," *Automatica,* vol. 13, no. 1, pp. 59-79, 1977.

[28] Y. Altintas and B. Sencer, "Identification of 5-Axis Machine Tools Feed Drive Systems for Contour Simulation," in *Proceedings of 4th CIRP International Conference on High Performance Cutting*, 2010.

[29] B. Armstrong-Helouvry, P. Dupont and C. Wit, "A survey of models, analysis tools and compensation methods for the control of machines with friction," *Automatica,* vol. 30, no. 7, pp. 1083-1138, 1994.

[30] Z. Jamaludin, H. Van Brussel and J. Swevers, "Friction Compensation of an XY Feed Table Using Friction-Model-Based Feedforward and an Inverse-Model-Based Disturbance Observer," *IEEE Transactions on Industrial Electronics,* vol. 56, no. 10, pp. 3848-3853, 2009.

[31] C. Canudas de Wit, H. Olsson and K. Astrom, "A New Model for Control of Systems with Friction," *IEEE Transactions on Automatic Control,* vol. 40, no. 3, pp. 419-425, 1995.

[32] J. Swevers, F. Al-Bender, C. G. Ganseman and T. Prajogo, "An Integrated Friction Model Structure with Improved Presliding Behavior for Accurate Friction Compensation," *IEEE Transactions on Automatic Control,* vol. 45, no. 4, pp. 675-686, 2000.

[33] F. Al-Bender, V. Lampaert and J. Swevers, "The Generalized Maxwell-Slip Model: A Novel Model for Friction Simulation and Compensation," *IEEE Transactions on Automatic Control,* vol. 50, no. 11, pp. 1883-1887, 2005.

[34] V. Lampaert, F. Al-Bender and J. Swevers, "A Generalized Maxwell-Slip Friction Model appropriate for Control Purposes," in *International Conference of Physics and Control*, Saint Petersburg, 2003.

[35] C. Canudas de Wit and P. Lischinsky, "Adaptive Friction Compensation with Dynamic Friction Model," in *13th World Congress of IFAC*, San Francisco, 1996.

[36] K. Worden, C. X. Wong, U. Parlitz, A. Hornstein, D. Engster, T. Tjahjowidodo, F. Al-Bender, D. D. Rizsos and S. D. Fassois, "Identification of pre-sliding and sliding friction dynamics: Grey box and black-box models," *Mechanical Systems and Signal Processing*, vol. 21, no. 1, pp. 514-534, 2007.

[37] S.-J. Kim and I.-J. Ha, "A Frequency-Domain Approach to Identification of Mechanical Systems with Friction," *IEEE Transactions on Automatic Control*, vol. 46, no. 6, pp. 888-893, 2001.

[38] R. H. A. Hensen, M. J. G. van de Molengraft and M. Steinbuch, "Frequency domain identification of dynamic friction model parameters," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 2, pp. 191-196, 2002.

[39] J. Y. Yoon and D. L. Trumper, "Friction modeling, identification, and compensation based on friction hysteresis and Dahl resonance," *Mechatronics*, vol. 24, no. 6, pp. 734-741, 2014.

[40] H. S. Lee and M. Tomizuka, "Robust Motion Controller Design for High-Accuracy Positioning Systems," *IEEE Transactions on Industrial Electronics*, vol. 43, no. 1, pp. 48-55, 1996.

[41] D. Sun and M. C. Tong, "A Synchronization Approach for the Minimization of Contouring Errors of CNC Machine Tools," *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 4, pp. 720-729, 2009.

[42] S.-L. Chen and C.-Y. Chou, "Contouring Control of Multi-Axis Motion Systems for NURBS Paths," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 1062-1071, 2016.

[43] 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, pp. 308-322, 2013.

[44] W. W.-S. Wong and K. Erkorkmaz, "Constrained identification of virtual CNC drives using a genetic algorithm," *International Journal of Advanced Manufacturing Technology*, 2010.

[45] DMG MORI, "LASERTEC 50 Shape," DMG MORI, 2018. [Online]. Available: https://en.dmgmori.com/products/machines/advanced-technology/lasertec/lasertec-shape/lasertec-50-shape. [Accessed 30 May 2018].

[46] M. H. Turhan, G. W. G. Tseng, Y. Hosseinkhani, K. Erkorkmaz and B. Fidan, "Identification of virtual machine tool feed drives from in-process gathered data," in *ASPE 2016 Spring Topical Meeting*, Boston, 2016.

[47] G. W. G. Tseng, K. Erkorkmaz and S. Engin, "Constructing Virtual Models of CNC Drives from In-Process Data," in *6th International Conference on Virtual Machining Process Technology*, Montreal, 2017.

[48] MathWorks, "GlobalSearch," [Online]. Available: https://www.mathworks.com/help/gads/globalsearch.html. [Accessed 10 June 2018].

[49] MathWorks, "patternsearch," [Online]. Available: https://www.mathworks.com/help/gads/patternsearch.html. [Accessed 10 June 2018].

[50] MathWorks, "MultiStart," [Online]. Available: https://www.mathworks.com/help/gads/multistart.html. [Accessed 10 June 2018].

[51] Haas, "Inverse-Time and Feed-Per-Minute Modes," [Online]. Available: https://diy.haascnc.com/reference-docs/inverse-time-and-feed-minute-modes. [Accessed 10 June 2018].

[52] C. Q. G. Chen, "Time-Optimal Feedrate Planning for Freeform Toolpaths for Manufacturing Applications," University of Waterloo, Waterloo, 2018.

[53] MathWorks, "ss," [Online]. Available: https://www.mathworks.com/help/control/ref/ss.html#f4-446923. [Accessed 12 February 2018].

[54] MathWorks, "balance," [Online]. Available: https://www.mathworks.com/help/matlab/ref/balance.html. [Accessed 12 February 2018].

[55] MathWorks, "tf2ss," [Online]. Available: https://www.mathworks.com/help/signal/ref/tf2ss.html. [Accessed 30 Jult 2018].

[56] MathWorks, "Continuous-Discrete Conversion Methods," [Online]. Available: https://www.mathworks.com/help/control/ug/continuous-discrete-conversion-methods.html#bs78nig-8. [Accessed 8 February 2018].

[57] M. Egan, "This Machine Can Cut Titanium Like A Hot Knife Slicing Through Butter," GE Reports, 14 March 2016. [Online]. Available: https://www.ge.com/reports/this-machine-can-cut-titanium-like-a-hot-knife-through-butter/. [Accessed 30 May 2018].

[58] K. Wong, "Design of a Chatter Inducing Spindle Flexure," Waterloo, 2016.

[59] Heidenhain , Mounting Instructions, 2006.

[60] Deckel Maho, *Technical Fundamentals Machine Manual: DMU 80P/FD High-Dyn TNC 430.*

[61] ETEL, "ILM Linear Motors - Specifications," [Online]. Available: http://www.etel.ch/en/linear-motors/ilm/specifications/.

[62] R. G. Budynas and J. K. Nisbett, Shigley's Mechanical Engineering Design, New York: McGraw Hill, 2015.

[63] Heidenhain, User's Manual ACCOM, 2015.

[64] D. Ewins, Modal Testing: Theory, Practice and Application, Baldock, England: Research Studies Press, 2000.

[65] M. H. Turhan, G. W. G. Tseng, K. Erkorkmaz and B. Fidan, "Reduced Parameter GMS Friction Model," in *7th International Conference on Virtual Machining Process Technology*, Hamilton, 2018.

Appendix A: Peak Picking Method Data

Table A-1 Single mode top base plate flexure peak picking data

Mode	Point	wn [Hz]	wr,1 [Hz]	wr,2 [Hz]	Zeta []	Q
1	1	28	27.5	28.5	0.0179	126.7
	2	28	27.5	28.5	0.0179	123.9
	3	28	27.5	28.5	0.0179	123.6
	4	28	27.5	28.5	0.0179	126.1
	5	28	27.5	28.5	0.0179	126.9
	6	28	27.5	28.5	0.0179	123

Table A-2 Single mode bottom base plate flexure peak picking data

Mode	Point	wn [Hz]	wr,1 [Hz]	wr,2 [Hz]	Zeta []	Q
1	1	28	27.5	28	0.0092	1.199
	2	28	27.5	28	0.0092	0.6416
	3	28	27.5	28.5	0.0179	1.384
	4	28	27.5	28	0.0092	1.526
	5	28	27.5	28.5	0.0179	1.353
	6	28	27.5	28	0.0092	1.707

Table A-3 Two mode same direction top base plate flexure peak picking data

Mode	Point	wn [Hz]	wr,1 [Hz]	wr,2 [Hz]	Zeta []	Q
1	1	23.5	23.5	24	0.0103	46.08
	2	23.5	23.5	24	0.0103	45.65
	3	23.5	23.5	24	0.0103	44.57
	4	23.5	23.5	24	0.0103	46.92
	5	23.5	23.5	24	0.0103	45.28
	6	23.5	23.5	24	0.0103	44.48
2	1	61	61	61.5	0.0040	13.5
	2	61	61	61.5	0.0040	13.58
	3	61	61	61.5	0.0040	13.52
	4	61	61	61.5	0.0040	13.91
	5	61	61	61.5	0.0040	14.21
	6	61	61	61.5	0.0040	14.12

Table A-4 Two mode same direction middle base plate flexure peak picking data

Mode	Point	wn [Hz]	wr,1 [Hz]	wr,2 [Hz]	Zeta []	Q
1	1	23.5	23.5	24	0.0103	46.08

	2	23.5	23.5	24	0.0103	45.65
	3	23.5	23.5	24	0.0103	44.57
	4	23.5	23.5	24	0.0103	46.92
	5	23.5	23.5	24	0.0103	45.28
	6	23.5	23.5	24	0.0103	44.48
2	1	61	61	61.5	0.0040	13.5
	2	61	61	61.5	0.0040	13.58
	3	61	61	61.5	0.0040	13.52
	4	61	61	61.5	0.0040	13.91
	5	61	61	61.5	0.0040	14.21
	6	61	61	61.5	0.0040	14.12

Table A-5 Two mode same direction bottom base plate flexure peak picking data

Mode	Point	wn [Hz]	wr,1 [Hz]	wr,2 [Hz]	Zeta []	Q
1	1	23.5	23.5	24	0.0103	1.485
	2	23.5	23.5	24	0.0103	1.369
	3	23.5	23.5	24	0.0103	1.428
	4	23.5	23.5	24	0.0103	1.447
	5	23.5	23.5	24	0.0103	1.362
	6	23.5	23.5	24	0.0103	1.44
2	1	61	60.5	61.5	0.0082	-2.378
	2	61	60.5	61.5	0.0082	-2.348
	3	61	60.5	61.5	0.0082	-2.347
	4	61	60.5	61.5	0.0082	-2.309
	5	61	60.5	61.5	0.0082	-2.459
	6	61	60.5	61.5	0.0082	-2.561

Table A-6 Two mode opposite direction flexure peak picking data x-direction

Base Plate	Point	wn [Hz]	wr,1 [Hz]	wr,2 [Hz]	Zeta []	Q
Top	1	62	59	66	0.0555	7.044
	2	62	58.5	66	0.0602	7.32
	3	62	59	66	0.0555	6.948
	4	62	58.5	66	0.0602	7.015
	5	62	58.5	66	0.0602	7.059
	6	62	59	66	0.0555	6.811
Middle	1	62	59	66	0.0555	6.516
	2	62	58.5	66.5	0.0635	6.312
	3	62	58.5	65	0.0534	7.482

	4	62	59	65	0.0486	8.051
	5	62	59	65.5	0.0521	7.376
	6	62	58.5	65.5	0.0568	7.802

Table A-7 Two mode opposite direction flexure peak picking data y-direction

Base Plate	Point	wn [Hz]	wr,1 [Hz]	wr,2 [Hz]	Zeta []	Q
Top	1	29	29	29.5	0.0084	82.63
	2	29	29	29.5	0.0084	89.904
	3	29	28.5	29.5	0.0173	81.134
	4	29	28.5	29.5	0.0173	94.223
	5	29	28.5	29.5	0.0173	82.087
	6	29	28.5	29.5	0.0173	82.087
Middle	1	29	28.5	29	0.0088	-0.2691
	2	29	28.5	29	0.0088	-0.6245
	3	29	28.5	29	0.0088	-0.2206
	4	29	28.5	29	0.0088	-0.069
	5	29	28.5	29	0.0088	1.1
	6	29	28.5	29	0.0088	0.8791