2010

Identification of Unmodeled Dynamics in Rotor Systems Using Mu-Synthesis Approach

Ryan J. Madden
Cleveland State University

IDENTIFICATION OF UNMODELED DYNAMICS IN ROTOR SYSTEMS USING MU-SYNTHESIS APPROACH

RYAN J. MADDEN

Bachelor of Mechanical Engineering

Cleveland State University

August, 2009

Submitted in partial fulfillment of requirements for the degree

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

at the

CLEVELAND STATE UNIVERSITY

December, 2010

This thesis has been approved

for the Department of MECHANICAL ENGINEERING

and the College of Graduate Studies by

_____

Dr. Jerzy T. Sawicki, Thesis Committee Chairperson
Department of Mechanical Engineering, CSU

_____

Dr. Hanz Richter
Department of Mechanical Engineering, CSU

_____

Dr. Dan Simon
Department of Electrical and Computer Engineering, CSU

ACKNOWLEDGEMENTS

I would like to thank Dr. Jerzy T. Sawicki for serving as my advisor and providing his guidance and expertise throughout this project. Without his insistence, I would never have pursued my master's degree and without his support, I would never have been able to successfully complete it. I would also like to thank Dr. Dan Simon and Dr. Hanz Richter for serving on my thesis committee and donating their time and expertise to this thesis. I would also like to express my gratitude to all of my colleagues in the Center for Rotating Machinery Dynamics and Control at Cleveland State University. Special thanks go to Alexander Pesch for being a constant resource in the laboratory.

I am forever in debt to my family for their unwavering love and support throughout my academic career, which has made all of my accomplishments possible.

This work was partially funded by the NASA Aviation Safety and Security Program, "Research Opportunities in Aeronautics", Grant No. NNX08AC31A. My gratitude goes to NASA for making this work possible.

IDENTIFICATION OF UNMODELED DYNAMICS IN

ROTOR SYSTEMS USING MU-SYNTHESIS APPROACH

RYAN J. MADDEN

ABSTRACT

It is well recognized that analytical models only approximate the true dynamics of analyzed rotating machines, due to the presence of components that are inherently difficult to model. Such models of rotating machines are driven by the best engineering knowledge and experience, and very often are updated based on experimental results. The problem of unmodeled or missing dynamics can be exacerbated in the presence of rotor structural damage such as a transverse crack on a shaft. This thesis will present an effective approach for model updating using advanced tools developed in robust control theory, specifically mu-synthesis. The methodology will be introduced based on a simple three-mass system and then applied to identification of the minute changes in the dynamics of the rotor test rig due to the presence of a transverse crack. Experimental data collected from the cracked rotor rig will be utilized to validate the developed approach.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

1.1 Introduction

Uncertainty is a persistent problem in the creation of engineering models. This uncertainty may come from parameters which are difficult to quantify, assumptions made in the modeling process, wear of the system components over time, or inaccuracies in experimental testing. These uncertainties are present in even the most rigorously developed engineering models, even those tuned to experimental results. These uncertainties may lead to a model which does not accurately predict the true experimental response of the system. Model updating has been developed to fix this problem by utilizing various schemes to drive the analytical model to match the experimental results. One such strategy, model-based identification, utilizes the assumption that part of the true

system's dynamics have been left out because either they are difficult to model or that a model simply does not exist. These missing dynamics become known as the unmodeled dynamics. This work aims to apply a model updating method, specifically model-based identification, to two systems with a known difference in an attempt to model this difference in dynamics. The two systems that will be studied are a healthy rotor and a rotor with a transverse crack at its mid-span. The result of this study will be a new model for the change in dynamics induced by the presence of a transverse crack.

This first chapter presents some examples of modeling and the appearance of unmodeled dynamics, a review of works which have led to this point and the objectives of the thesis.

1.2 Analytical Models in Real Engineering

In general, analytical models in engineering are developed to predict the behavior of the system under consideration. One of the most widely used techniques in engineering modeling is finite element modeling. These finite element models are usually tuned to experimental responses of the system. This technique can be seen in models of bridges (Farrar and James III 1997), airplanes (Ruotolo and Surace 1998) and nuclear power plants (Sinha and Friswell 2003), to name a few. Rotor systems are also modeled using finite elements, as is done in this work. For example, the finite element method was used to model an airplane wing in (Ruotolo and Surace 1998) and the resulting model was used to develop on-line health monitoring, similarly for a rotor system (Sawicki and Friswell 2010).

The field of system identification takes a different approach to modeling. System identification uses various curve-fitting techniques to create a model from a system response to a specific input. Farrar and James III (1997) present the system identification of a bridge using the cross-coupling of ambient vibration measurements.

Analytical models have even been created for faults in mechanical systems. Gasch (1976), (1993), Mayes and Davies (1980), (1984) were pioneers in the field of crack modeling in rotating machinery. Fault models are used for fault detection and isolation, so having an accurate model is important because it allows for the earliest possible detection. "It is necessary to have the earliest fault-warning system to avoid having a fault develop into a catastrophic event, a failure" (Esteban 2004).

Modeling difficulties and unmodeled dynamics appear in each modeling approach. Modeling difficulties are inherent, they appear in the finite element modeling of every system, and in the case of rotors in "large, abrupt changes in rotor diameter; fits between rotor and disks; seals" (Vazquez, et al. 2003). Unmodeled dynamics in system identification is contained in the model uncertainty. This uncertainty may be evaluated at the end of the process or during the identification algorithm. Hsu et al. (2006) present a new adaptation in system identification which places model uncertainty into the identification process.

1.3 Literature Review

1.3.1 Model Updating

In this section, a brief introduction to the field of model updating is presented. Model updating developed as a specific application of system identification which "Seek[s] to correct the inaccurate parameters in the model so that the agreement between predictions and test results are improved" (Mottershead and Friswell 1993). The Mottershead and Friswell survey paper (1993) offers a more comprehensive look at the development of the field.

The first methods for model updating became known as the direct methods, named for the direct manner in which the mass and stiffness matrices were updated to achieve the desired performance. The first method for direct model updating is introduced by Baruch and Bar Itzhack (1978). This method assumes that the analytical mass matrix is known. This assumption is used along with the measured eigenvalues to derive a new optimal stiffness matrix. The method becomes known as the first method of reference basis, in which the mass matrix is the reference basis. Berman (1979) adds to Baruch's work by developing the second method of reference basis. Berman uses the measured modes as the reference basis to correct the mass matrix. Baruch (1984) completes this method by introducing the third method of reference bias which uses the stiffness matrix as the reference basis in order to correct the mass matrix.

Srinathkumar (1978) and Andry et al. (1983) develop the method of eigenstructure assignment for model correction. This technique drives a system to a set of desired eigenvalues and eigenvectors by using a controller in a feedback loop. Minas and Inman (1990) improve on this method by applying the controller to the stiffness and damping

matrices in a finite element model. The work is successful in making the model match the experimental modal data but Minas and Inman explain a weakness in that, the method "…does not, however, guarantee that the resulting modified stiffness and damping matrix have the same physical significance they had from the original modeling."

Model updating was later enhanced by the method of inverse eigenvalue techniques, first developed by Gladwell (1986). Bucher and Braun (1993) presented an application of the method of inverse eigenvalues. In this paper, structural modifications were successfully found for beam structures which would drive them to achieve a desired mode shape.

The second set of model updating methods became known as the penalty methods. The first set of these penalty methods were used on modal data. These methods use a sensitivity function which is based on how changes in the parameters affect the measured output. Friswell (1989) gives an example of such a penalty method. This paper presents an algorithm for updating selected stiffness or mass parameters using a minimum variance estimator. A few pitfalls in this approach are that it is iterative, so may stop on local minima or maxima and that once the parameters are updated they may lose their physical meaning.

Friswell and Penny (1990) improve upon the penalty methods by updating analytical models with comparison to the test frequency response functions directly. This application is most useful in situations where the derivation of the test data's modal model is very difficult. An advantage in this approach is that the data has endured one less processing step, which may remove a source of error in the final result. Friswell and

Penny (1992) go on to apply this method to models with close or even repeated eigenvalues, showing that the algorithm works for systems that are not ideal.

Xiong et al. (2008) offer a more recent examination of various penalty methods which include those that are parameter-based and directly use the experimental frequency response function. The work looks at the effectiveness of both the parametric deterministic calibration approach and the non-parametric bias correction methods. The Bayesian bias-correction model, that is evaluated is credited to Chen et al. (2006), uses a bias function that directly uses the difference between the experimental and analytical model. The deterministic calibration approach separates the analytical model between controllable inputs and uncontrollable parameters. These parameters are then tuned in a nonlinear regression analysis to minimize the error between the model and the experimental data.

The third method was first developed by Maslen et al. (2002) and has come to be known as "model reconciliation." Model reconciliation uses the basic approach of model identification i.e., controlling the nominal model in order to minimize the error between its response and that of the experimentally identified model. The assumption in model reconciliation is that the basic model structure is correct, but there is missing dynamics that is left out because of modeling errors or unknown model phenomena of specific parts of the system. This missing dynamics become known as the "unmodeled dynamics" which is absent in the nominal engineering model. This method is innovative in that it distinguishes between the parts of the system which are not certain in modeling and the easily modeled portions. Accordingly, an H_∞ controller is applied only at the uncertain locations and the systems are driven to minimize the modeling error. When the two

responses match, the controller, or its dynamics, represents the unmodeled dynamics which was missing in the nominal model.

Vazquez et al. (2003) applied the model reconciliation method to identify the dynamics of magnetic bearing journals. In order to identify the magnetic journal bearings, they are recognized as the portion of the system which would be difficult to model. This realization leads to the bearings being left out of the nominal model. Conversely, the experimental model is derived from the frequency response of the entire system with the magnetic rotors in action. Finally, the model reconciliation process is used. The unmodeled dynamics is discovered, which is known to be the behavior of the magnetic bearing journals. The paper concludes with this model for the dynamics of the magnetic bearing journals which reconciled the nominal and experimental models.

Wang and Maslen (2006) continue the work on model reconciliation by making two important contributions to the method. First is the direct use of the frequency response functions (FRF) for the nominal and engineering systems. Use of the FRFs removes the first step of creating models for each of the systems. This takes away one modeling step, removing an error source which should lead to a more accurate final model. The second advancement is the use of a μ-controller to drive the model correction. An application of a μ-controller opens up μ-analysis machinery, which contributes uncertainty bounds for the model and unmodeled dynamics. This is the first estimate of the quality of the presented unmodeled dynamics and allows for the examination of modeling and experimental uncertainty effects on the unmodeled dynamics. Wang's dissertation (2008) presents a more exhaustive look at the concept covered in the Wang and Maslen paper (2006).

The book written by Zhou et al. (1995) is a valuable resource for a background on μ theory; also the work of Maslen and Sawicki (2007) explains advantages inherent in the use μ-synthesis for control applications. Maslen and Sawicki note that while μ-synthesis may not always produce significantly better results than simpler methods, it provides the creator with more physical insight into the system behavior. In comparing the method to traditional PID controllers the authors note, "In contrast to hand synthesis, the parameters of the μ-synthesis design process are precisely the specifications themselves." The transparency of the design parameters provides a simple physical interpretation of how the controller is working, which is the primary strength of μ-synthesis as a control method.

The next work from Wang et al. (Identification in Rotordynamics: Model-Based vs. Direct Measurements 2009a) provides a new derivation of the model reconciliation method, now referring to it as "model-based identification." This derivation solves the error minimization using the transfer functions of the nominal engineering model and the true, experimentally identified model. These transfer functions are then measured experimentally and the results are compared to those found from using the engineering model. Similar to the previous work of Wang and Maslen (2006), this new method removes one modeling step from the process, which would remove one instance of modeling error.

The most recent paper authored by Wang et al. (Identification in Rotordynamics: Uncertainty Analysis and Quality Estimation 2009b) on the subject on model reconciliation or model-based identification gives a detailed analysis on the quality estimation in the unmodeled dynamics. The paper utilizes μ-analysis and the linear

fractional transformation to derive the upper and lower μ bounds for the unmodeled dynamics. This provides an illustrative quality estimation of the unmodeled dynamics, showing where the modeling process has been the most or least accurate.

An application of model identification methods that is related to this work is in the area of fault detection. A concise background on fault detection is given in (Esteban 2004). Fritzen et al. (1998) use an inverse sensitivity approach, a specific penalty method, on both modal and frequency response function data for damage localization. Sinha and Friswell (2003) used a gradient based sensitivity approach in the detection of cracks in nuclear power plant components. Jaishi and Ren (2006) use a similar sensitivity model updating method to identify the damage in a reinforced concrete beam, both in simulation and an experiment.

1.3.2 Crack Modeling in Rotors

Crack modeling in rotors can be divided into two subsets: breathing and non-breathing cracks. The modeling of breathing cracks is of particular interest to this thesis because the data to be used later is of a spinning rotor which under the assumption of weight dominance, should be breathing. Penny et al. (2006) explain weight dominance as the condition of the crack opening and closing depending only on the rotor angle because the static deflection is significantly larger than the amplitude of vibration. The first breathing crack model was presented by Gasch (1976), (1993). This hinge model has the crack abruptly change between the open and closed state as a function of the rotation angle. Mayes and Davies (1980), (1984) added to the work of Gasch by deriving a crack model which opened and closed based on a cosine function. This innovation allowed for

a smooth transition between the open and closed crack states. Finally, Jun et al. (1992) created a crack model which was derived from the theory of fracture mechanics. Penny and Friswell (2002) offer a theoretical comparison of the three crack models and simulate the quality of each in representing a crack in a Jeffcott rotor.

Penny et al. (2006) go on to apply a Mayes model for a breathing crack in a rotor to simulate the response of a cracked rotor under excitation of an active magnetic bearing. The aim of this study was to predict these behaviors in order to establish a crack detection strategy. Pesch (2008), Wroblewski (2008) and Storozhev (2009) experimentally prove this technique.

1.4 Objectives of Thesis

The primary objective of this thesis is to detect the changes in system dynamics induced by the presence of a transverse crack in a rotor. The crack dynamics will be found by an application of model-based identification, in which the crack is a known difference between the "nominal" healthy rotor and the "true" cracked rotor. The process of model-based identification will find the unmodeled dynamics, or the necessary changes to the nominal model that drives its response to match the true system. This unmodeled dynamics will become the change in dynamics induced by the transverse crack. The purpose for developing a new approach for estimating the transverse crack dynamics is to create a more reliable strategy of detecting this fault in the future. To that end, the scope of the thesis is as follows.

Chapter 2 provides a theoretical background on the powerful controls tools which are used in the method of model-based identification. First, a brief introduction and literature

review of robust control will be given. Next, the basic concepts that are involved in H_∞ and μ-synthesis controls, and the linear fractional transformation are developed. Finally, an illustrative example of H_∞ and μ-synthesis is given.

Chapter 3 develops the method of finding a known difference in dynamics by utilizing model-based identification. First, the application of the method for finding missing dynamics is developed. Next, a simple three-mass example is given to prove the method. This involves presentations of the analytical results, the three-mass experimental test rig and the experimental results. Finally, conclusions are made about the three-mass example of unmodeled dynamics and how it may be applied to more interesting applications.

Chapter 4 presents the application to structural damage detection in rotating machinery. First, the crack detection test rig will be overviewed. Next, the formulation of the approach will be developed. Finally, the example of detecting the crack dynamics will be presented.

Chapter 5 concludes the paper with a discussion of the impact of the results and future work which may be needed.

CHAPTER II

ROBUST CONTROL TOOLS FOR MODEL UPDATING

2.1 Introduction

This chapter presents an introduction to the robust control tools that are utilized in this thesis for model-based identification. First, a brief history of H_∞ and μ-synthesis controls will be presented. Next, the control methods will be derived along with a description of how to use these controllers in a MATLAB® environment. Finally, a two-mass example of H_∞ and μ-synthesis controls will be presented. The goal of this chapter is to provide a sufficient background on the tools which will be used in model-based identification along with demonstrating their utility in an example.

Zames (1981) is credited for authoring the first work on the H_∞ control problem. In this paper he uses state feedback control which is optimized using an analysis of the H_∞

norm. Next, the two papers authored by Glover and Doyle (1988) and (1989) generalize the H_∞ control problem. First, Glover and Doyle (1988) lay out the state-space partitioning of the system plant which provides the methods ease of use in MATLAB. Next, Glover and Doyle (1989) place the controller in a linear fractional transformation with the plant. Doyle et al. (1989) go on to create a more complete tutorial for H_∞ controller synthesis. Alternatively, Sampei et al. (1990) derive a purely algebraic method for synthesizing the controller. Meanwhile, Ran and Vreugdenhil (1988) explicitly proved the math behind solving the two Riccati equations which leads to the development of the H_∞ controller. Boyd et al. (1989) added to the state of the art by creating a bisection algorithm which allows for faster location of the frequency containing the H_∞ norm. Nagpal and Khargonekar (1991) present methods for improving the controller by utilizing filtering and smoothing.

Doyle (1982) developed the method of μ-synthesis by applying the addition of structured uncertainty to an H_∞ control problem. At the time, H_∞ controllers were able to handle the problem of uncertainty, but were not able to analyze structured uncertainty in the form of linear fractional transformations. Once Doyle applied the structured singular value, μ, to measuring the performance of an H_∞ controller with structured uncertainty, the D-K iteration was born. Later, Balas (1990) presented experimental examples to prove that the D-K iteration works.

2.2 Concepts

2.2.1 H_∞ Control

An H_∞ controller is an optimal linear time invariant state-space controller. This controller seeks to minimize the infinity norm of the closed-loop system, *clp*, resulting in the cost function:

$$\gamma = \|clp\|_\infty \tag{2.1}$$

The infinity norm of the closed-loop system is the maximum gain achieved across all frequencies. The infinity norm is closely related to the 2-norm, which is a measure of the size of a signal. It follows that the gain of the closed-loop system is the ratio of the 2-norm of the output over the 2-norm of the input. Thus the infinity norm identifies the largest value of this ratio for all frequencies (Doyle, Francis and Tannenbaum 1990). The closed-loop system is shown (Balas, Chiang, et al. 2009):

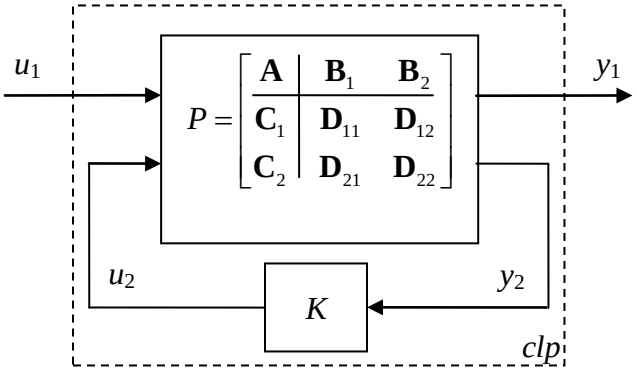


Figure II.1 H_∞ Control Schematic

where:

P is the system plant

K is the controller

u_1 is the external input

u_2 is the control input

y_1 is the closed-loop output

y_2 is the output to the controller

$\mathbf{A}$ is the system dynamics

$\mathbf{B}_1$ is the selector matrix for the external input

$\mathbf{B}_2$ is the selector matrix for the control input

$\mathbf{C}_1$ is the selector matrix for the closed-loop output

$\mathbf{C}_2$ is the selector matrix for the output to the controller

The goal of H_∞ control is to design a controller, K, which minimizes the infinity-norm of the transfer function from u_1 to y_1. Control of the plant is achieved by utilizing the application of a control force, u_2, based on a measured y_2 signal.

MATLAB's default method for solving the H_∞ control problem is shown in the Robust Toolbox Guide (Balas, Chiang, et al. 2009) and uses a Riccati method. This method utilizes Hamiltonian matrices, with the values from the plant, P (Zhou and Doyle 1998):

$$H = \begin{bmatrix} \mathbf{A} & \gamma^{-2}\mathbf{B}_1\mathbf{B}_1^* - \mathbf{B}_2\mathbf{B}_2^* \\ -\mathbf{C}_1^*\mathbf{C}_1 & -\mathbf{A}^* \end{bmatrix} \tag{2.2}$$

$$J = \begin{bmatrix} \mathbf{A}^* & \gamma^{-2}\mathbf{C}_1^*\mathbf{C}_1 - \mathbf{C}_2^*\mathbf{C}_2 \\ -\mathbf{B}_1\mathbf{B}_1^* & -\mathbf{A} \end{bmatrix} \tag{2.3}$$

where "*" denotes a complex conjugate transpose. An H_∞ controller, K, which meets the condition, $\|clp\|_\infty < \gamma$ exists when all three of the following conditions are met (Balas, Chiang, et al. 2009):

i. "*H* and *J* Hamiltonian matrices must have no imaginary-axis eigenvalues."

ii. "The stabilizing Riccati solutions X_∞ and Y_∞ associated with the Hamiltonian matrices must exist and be positive, semi-definite."

iii. "Spectral radius of (X_∞, Y_∞) must be less than or equal to γ^2."

The spectral radius is the magnitude of the maximum eigenvalue. This utilizes the solutions to the algebraic Riccati equations:

$$\mathbf{A}^* X_\infty + X_\infty \mathbf{A} + X_\infty \left(\gamma^{-2} \mathbf{B}_1 \mathbf{B}_1^* - \mathbf{B}_2 \mathbf{B}_2^* \right) X_\infty + \mathbf{C}_1^* \mathbf{C}_1 = 0 \tag{2.4}$$

$$\mathbf{A} Y_\infty + Y_\infty \mathbf{A}^* + Y_\infty \left(\gamma^{-2} \mathbf{C}_1^* \mathbf{C}_1 - \mathbf{C}_2^* \mathbf{C}_2 \right) Y_\infty + \mathbf{B}_1 \mathbf{B}_1^* = 0 \tag{2.5}$$

The algorithm works by first assuming a value for γ. Next, X_∞ and Y_∞ are solved for using Equations 2.4 and 2.5, and the three controller conditions are checked. Finally, the method employs a bisection search to find new γ values, and the process is repeated, eventually locating the minimum γ value.

Conveniently, the MATLAB function *hinfsyn* has been created to solve equations 2.1-2.5 and develop the H_∞ controller. With an automated solution to solving the math, the primary difficulty in developing the H_∞ controller becomes reconciling the system to be controlled to the closed-loop control schematic in Figure 2.1. The plant, *P*, is created from a state-space system divided with **A** as the system dynamics, $\mathbf{B}_1$ as the disturbance inputs, or external excitation, $\mathbf{B}_2$ is the control inputs, $\mathbf{C}_1$ are the outputs to be minimized, and $\mathbf{C}_2$ are the outputs to the controller (Balas, Chiang, et al. 2009).

Another important design consideration is weighting. Both the closed-loop inputs, $\mathbf{B}_1$, and outputs, $\mathbf{C}_1$, need to be weighted in order to keep the input and output signal magnitudes less than one. These weights are utilized to meet performance specifications

of the system. Consequently, the goal of the H_∞ controller becomes keeping the cost function $\gamma < 1$ while meeting the performance specifications. Both weighting and the resulting cost function value will be further discussed in the illustrative example of a two-mass system at the end of this chapter.

2.2.2 Mu-Synthesis

Mu-synthesis is a controller design process which expands on the methods of H_∞ control design. While a linear time invariant state-space controller is also created in the μ-synthesis approach, the advancement made in the method is that the μ-controllers are designed to handle structured uncertainty in the system plant. The μ-synthesis control schematic is shown in Figure 2.2:

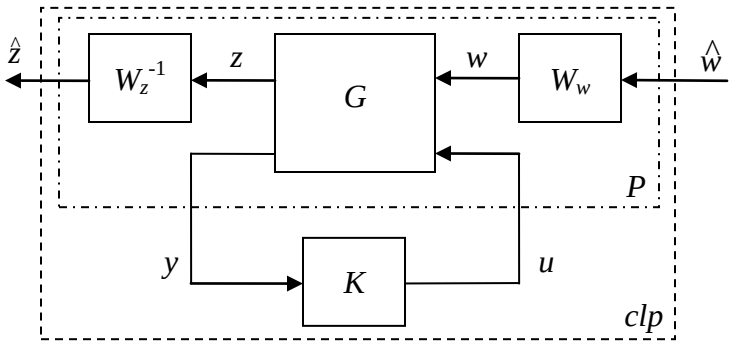


Figure II.2 μ-Synthesis Schematic

The plant P is created with the same partitioning as the H_∞ plant. Following Figure 2.1, $\mathbf{A}$ is the system dynamics, $\mathbf{B}_1$ is the disturbance inputs, etc. In the μ-synthesis diagram, the plant P has been divided between the weighting functions W_z^{-1} and W_w, and the rest of the state-space plant G for increased clarity. These weights, W_z^{-1} and W_w, are used to keep the magnitudes of the dimensionless output and input signals, $\hat{z}$ and $\hat{w}$, less

than or equal to one. W_w is used to scale the system's input to the magnitude of the excitation, which will later be referred to as the reasonable operating conditions. W_z^{-1} is used to place a range of permissible values on the system's output, which will be later referred to as a performance specification (Maslen and Sawicki 2007). The use of these weighting functions is the same for both H_∞ and μ-synthesis control.

The properties of the μ-synthesis approach are derived from the structured singular value, μ. The necessary and sufficient condition for a successful controller is that the maximum singular value of *clp* is less than 1 (Maslen and Sawicki 2007):

$$\sigma_{\max}(clp(j\omega)) \leq 1.0 \quad \forall \omega \in \mathbb{R} \tag{2.6}$$

This condition requires that the closed-loop input and output are weighted such that:

$$\left|\hat{z}_i\right| < 1.0 \quad : \quad \left|\hat{w}_i\right| < 1.0 \tag{2.7}$$

Equation 2.6 is equivalent to (Maslen and Sawicki 2007):

$$\sup_{\omega \in \mathbb{R}} \sigma_{\max}(clp(j\omega)) \leq 1.0 \tag{2.8}$$

Equation 2.8 is equivalent to the H_∞ norm of the closed-loop transfer function. The important distinction is that the infinity norm "…ignores the known block diagonal structure of the uncertainties" (Zhou and Doyle 1998). Meanwhile, the structured singular value takes the uncertainty structure into account.

Similar to designing an H_∞ controller, MATLAB has a built-in function, *dksyn*, which completes the derivation of the μ-controller (Balas, Chiang, et al. 2009). Like the *hinfsyn* command, *dksyn* only requires the input of a properly defined plant. The algorithm used in *dksyn* is the D-K iteration. First, a controller, *K*, is created for the open loop system using H_∞ synthesis, as described in the previous section. Then a frequency-dependent

scaling matrix $\mathbf{D}(\omega)$ is produced by solving the following optimization problem (Smith and Packard 1997):

$$\inf_{\mathbf{D}\in D} \sigma_{\max}\left[\mathbf{D}(\omega)F_{\ell}(P,K)\mathbf{D}(\omega)^{-1}\right] \tag{2.9}$$

where inf denotes the infimum of the set and F_l is a lower linear fractional transformation. This scaling matrix $\mathbf{D}(\omega)$ is found such that Equation 2.9 is an accurate upper bound to the μ-value of P. Initially, $\mathbf{D}(\omega)$ is a point-wise frequency function. Therefore, $\mathbf{D}(\omega)$ is rationalized to a transfer function $\hat{\mathbf{D}}(s)$. This scaling matrix $\hat{\mathbf{D}}(s)$ is then placed into the open-loop system as shown in Figure 2.3 and a new H_∞ controller is created.

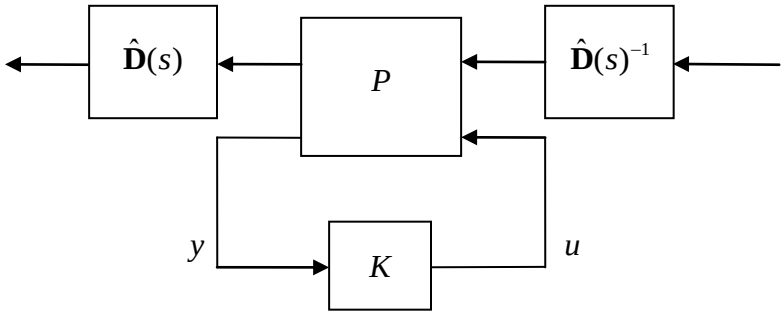


Figure II.3 Scaled μ -Synthesis Plant

Finally, the closed-loop μ-bound is calculated. If a value of $\mu < 1$ is returned, the controller successfully meets the design specifications and the algorithm is complete. If this condition is not met, the procedure is repeated starting at solving for a new $\mathbf{D}(\omega)$.

2.2.3 LFT Formulation

The linear fractional transformation is an important tool in robust controls. LFTs are seen in both H_∞ and μ-synthesis control, along with structured uncertainties. First, assume a plant, P, which can be partitioned into:

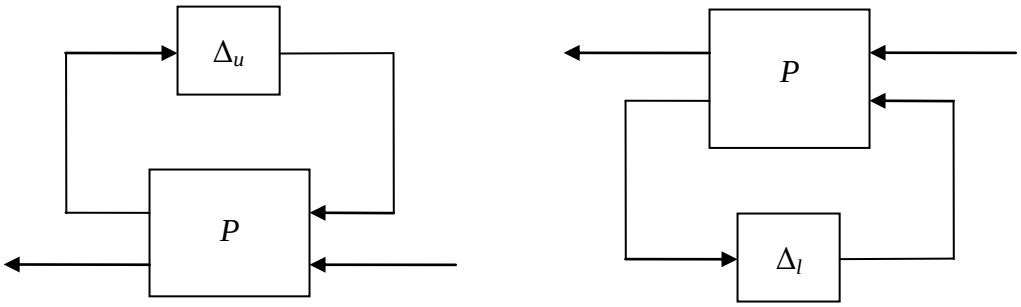
$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$$

It follows that the upper and lower LFTs, $\mathcal{F}_u$ and $\mathcal{F}_\ell$, respectively, are defined by the following equations (Zhou and Doyle 1998):

$$\mathcal{F}_u(P, \Delta_u) = P_{22} + P_{21}\Delta_u(I - P_{11}\Delta_u)^{-1}P_{12} \tag{2.10}$$

$$\mathcal{F}_\ell(P, \Delta_\ell) = P_{11} + P_{12}\Delta_\ell(I - P_{22}\Delta_\ell)^{-1}P_{12} \tag{2.11}$$

A diagram representation for the linear fractional transformations is shown in Figure 2.4:



Upper LFT Framework **Lower LFT Framework**

Figure II.4 Linear Fractional Transformation Framework

The figure shows how a plant, P, would interact with a structured uncertainty matrix, Δ. It can be seen that an LFT also describes the manner in which the controller interacts

with the plant in both the H_∞ and μ-synthesis control schemes as shown in Figures 2.1 and 2.2, respectively.

2.3 A Two-Mass Example of H_∞ Control and μ-Synthesis

To illustrate the use of H_∞ and μ-synthesis control, a simple two-mass example is presented. The intent of this example is to illustrate how easily performance criteria and reasonable operating conditions can be applied to a physical plant using both H_∞ and μ-synthesis control methods. In other words, this example is used to show how intuitive these control techniques are once the system is properly modeled.

The system to be controlled is made up of two masses, the upper mass, m_1, and the lower mass, m_2. Both masses are connected by a spring, k_1, while an additional spring, k_2, connects m_2 to ground. The objective of the controller is to manage the response of m_1 in the presence of an external disturbance force, f_e, through a control force, u, acting on m_2. The control sensor measures the displacement of m_2 and is polluted by the noise signal, η_e. A diagram of the system is shown in Figure 2.5.

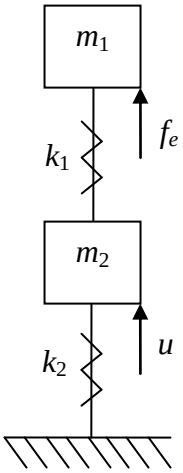


Figure II.5 Two-Mass System

Parameter values of the system are:

$$m_1 = 1 \text{ kg} \tag{2.12}$$

$$m_2 = 2 \text{ kg} \tag{2.13}$$

$$k_1 = 450 \text{ N/m} \tag{2.14}$$

$$k_2 = 175 \text{ N/m} \tag{2.15}$$

Equations of motion are:

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix} + \begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 + k_2 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix} f_e + \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} u \tag{2.16}$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + \eta_e \tag{2.17}$$

It is important to note that while the control problem can be classified as single-input single-output, the system still retains its multi-input multi-output behavior because the behavior of the system is dependent on four input/output pairs.

2.3.1 H_∞ Control Solution

Before the process of synthesizing an H_∞ controller may begin, the control objective needs to be explicitly defined. This objective is to keep x_1 less than 0.035 m for disturbance forces less than 10 N and noise less than 0.1 mm while not exceeding 50 N of control force. These conditions lead to a performance output of:

$$z = \begin{bmatrix} 28.6 & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + \begin{bmatrix} 0 \\ 0.02 \end{bmatrix} u \tag{2.18}$$

Values in Equation 2.18 are derived from the control objectives. The weighting on x_1 and u come from $x_{\max}^{-1}$ and $u_{\max}^{-1}$, respectively. Under "reasonable circumstances" the goal is:

$$\|z\|_2 < 1 \tag{2.19}$$

The "reasonable circumstances" are defined in the objective to mean that $|f_e| < 10\,\text{N}$ and $|\eta_e| < 0.0001\,\text{m}$ or, more simply interpreted:

$$\begin{Bmatrix} f_e \\ \eta_e \end{Bmatrix} = \begin{bmatrix} 10 & 0 \\ 0 & 0.0001 \end{bmatrix} \mathbf{f} \tag{2.20}$$

with the requirement that:

$$\|f\|_2 < 1 \tag{2.21}$$

Thus, the complete problem definition is to find a controller $u(s) = K(s)y(s)$ for:

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix} = -\begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 + k_2 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{f} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{2.22}$$

$$z = \begin{bmatrix} 28.6 & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + \begin{bmatrix} 0 \\ 0.02 \end{bmatrix} u \tag{2.23}$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + \begin{bmatrix} 0 & 0.0001 \end{bmatrix} f \tag{2.24}$$

subject to:

$$\|z\|_2 < 1$$

when:

$$\|f\|_2 < 1$$

This is the standard definition for an H_∞ control problem.

The resulting γ is a measure of the H_∞ norm of the closed-loop system, or more explicitly, the gain from normalized sensor noise and exogenous force to normalized displacement and control effort. When the output returns a value of $\gamma < 1$, then a controller was synthesized which is able to meet all of the performance objectives. Alternatively, an output of $\gamma > 1$ implies that the problem is ill-posed. The condition of being ill-posed may have two consequences. An example of an ill-posed problem would be setting the maximum displacement of m_1 to a value that is lower than the static displacement. This first option leads to tuning the control objectives and reasonable operating conditions in order to make a working controller. Alternatively, if these objectives are based on a real, physical model, then the output of $\gamma > 1$ proves that the

controller is not physically possible. MATLAB returns $\gamma = 0.9671$, meaning that the controller was successful.

The following figures give a more detailed analysis of the results. Figure 2.6 shows the frequency responses of the transfer functions of the closed-loop system. The inputs are the excitation force acting on m_1 and the sensor noise, respectively. The outputs are the displacement of m_1 and the control effort, respectively. The magnitude plot from the sensor noise input to the control effort output attenuates at two frequencies, 7.5 and 26.6 rad/s, corresponding to two natural frequencies in the open loop plant. Figure 2.7 shows the frequency response of the designed controller. Interestingly, the controller is unstable, with two of its four eigenvalues in the right half plane. However, this instability is allowed because the controller drives all of the closed-loop system's eigenvalues to stability. Next, Figure 2.8 shows the maximum singular value of the closed-loop system plotted against the frequency. An important observation is that the highest value shown in the maximum singular value plot is 0.9671, which matches the reported value for γ.

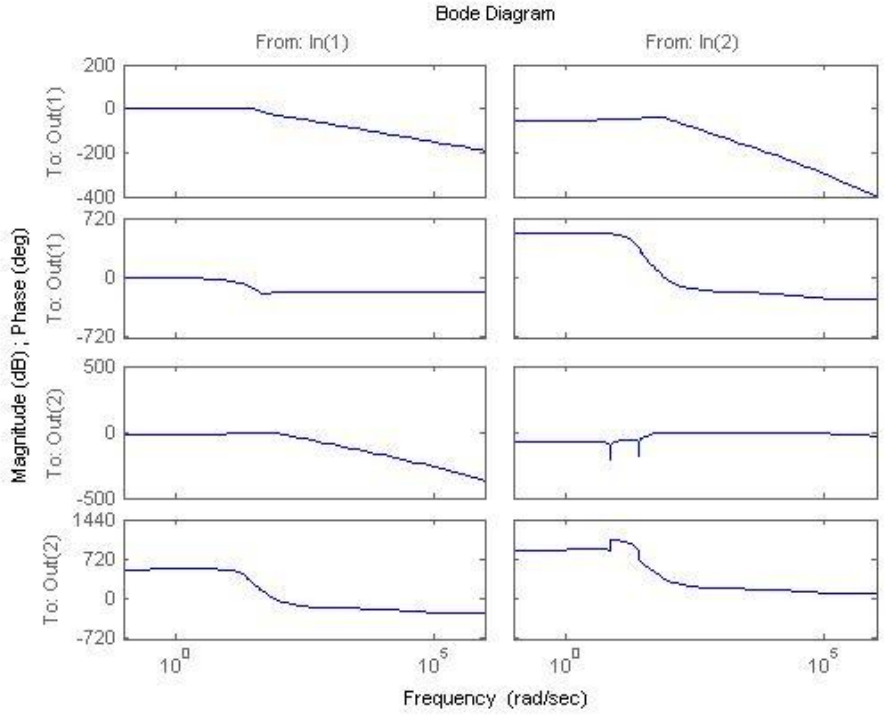


Figure II.6 H_∞ Closed-loop Transfer Functions
Input 1: f_e Input 2: η_e Output 1: Displacement of m_1 Output 2: u

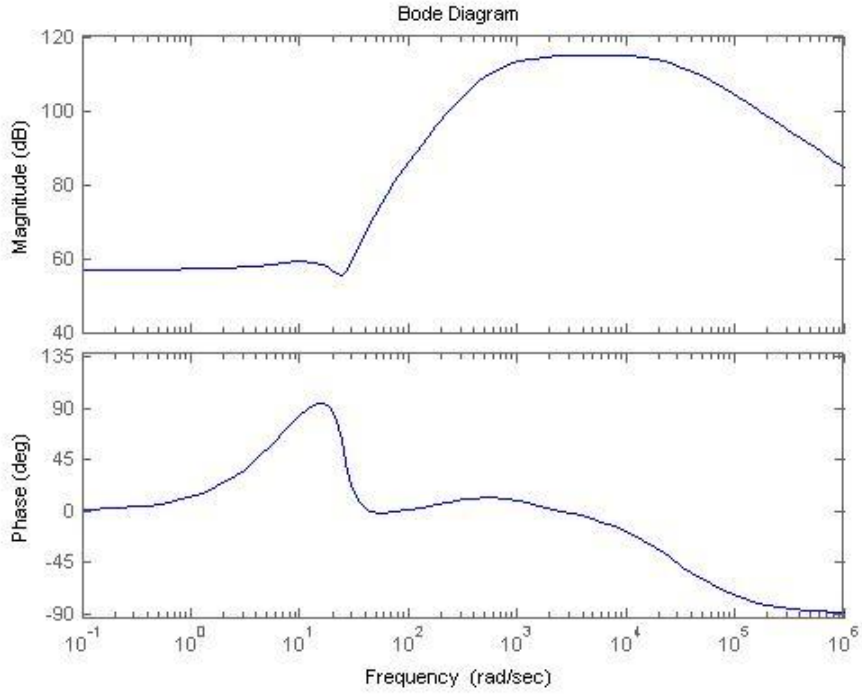


Figure II.7 H_∞ Controller Frequency Response

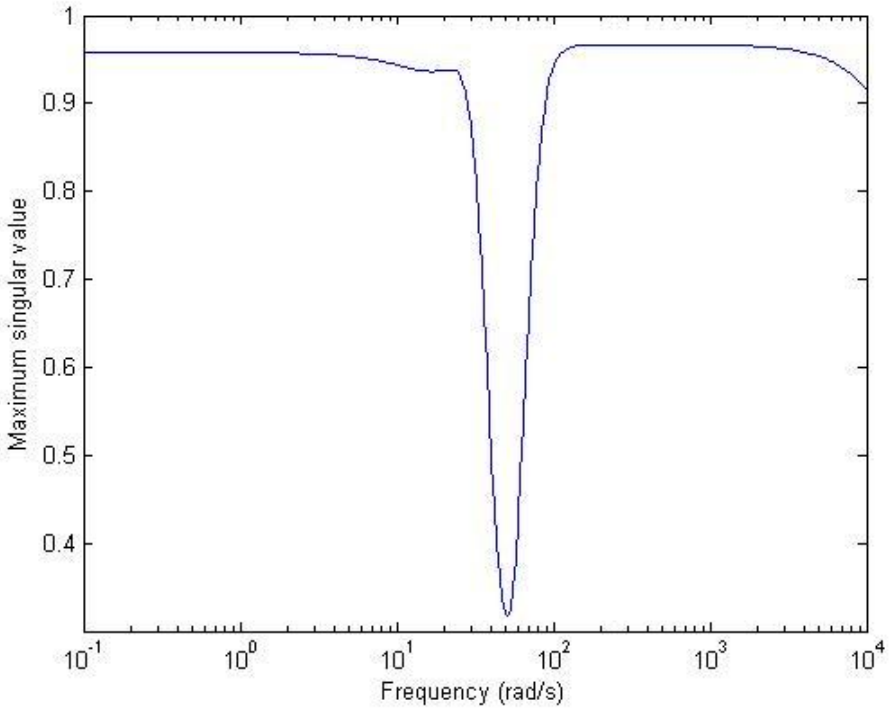


Figure II.8 H_∞ Closed-Loop Maximum Singular Value

2.3.2 Mu-Synthesis Solution

In order for this problem to be extended to μ-synthesis, uncertainty needs to be added to the system parameters. A nominal uncertainty value of two percent was chosen for all of the parameters:

$$m_1 = 1 \pm 0.02\,\text{kg} \tag{2.25}$$

$$m_2 = 2 \pm 0.04\,\text{kg} \tag{2.26}$$

$$k_1 = 450 \pm 9\,\text{N/m} \tag{2.27}$$

$$k_2 = 175 \pm 3.5\,\text{N/m} \tag{2.28}$$

The weighting functions W_w and W_z^{-1} are derived from the performance criteria. Both of these weighting functions are required to normalize the input and output of the closed-

loop system to absolute values < 1. Recall that the maximum external disturbance force is 10 N and the displacement of m_1 needs to be kept under 0.035 m. This performance criteria leads to the weighting functions:

$$W_w = 10\,\text{N} \tag{2.29}$$

$$W_z = 1/0.035 = 28.6\,\text{m}^{-1} \tag{2.30}$$

MATLAB returns $\mu = 1.11$, showing that the controller is no longer able to meet the performance specifications once a two percent uncertainty is introduced in the system. Clearly, the addition of uncertainty to the system leads to a more difficult system to control. The graphical output shows a similar behavior to that of the H_∞ controlled system. Figure 2.9 shows the bode plot of the transfer functions of the closed-loop system. Similar to Figure 2.6, the inputs are the excitation force acting on m_1 and the sensor noise, respectively. The outputs are again the displacement of m_1 and the control effort, respectively. Figure 2.10 presents the frequency response of the μ-controller.

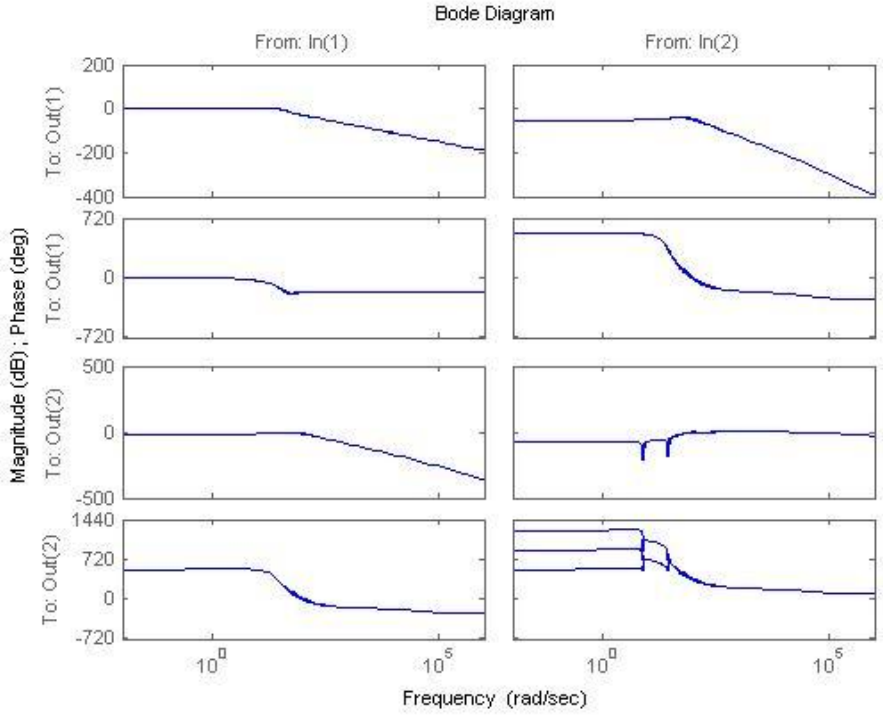


Figure II.9 μ-Synthesis 1 Closed-Loop Transfer Functions
Input 1: f_e Input 2: η_e Output 1: Displacement of m_1 Output 2: u

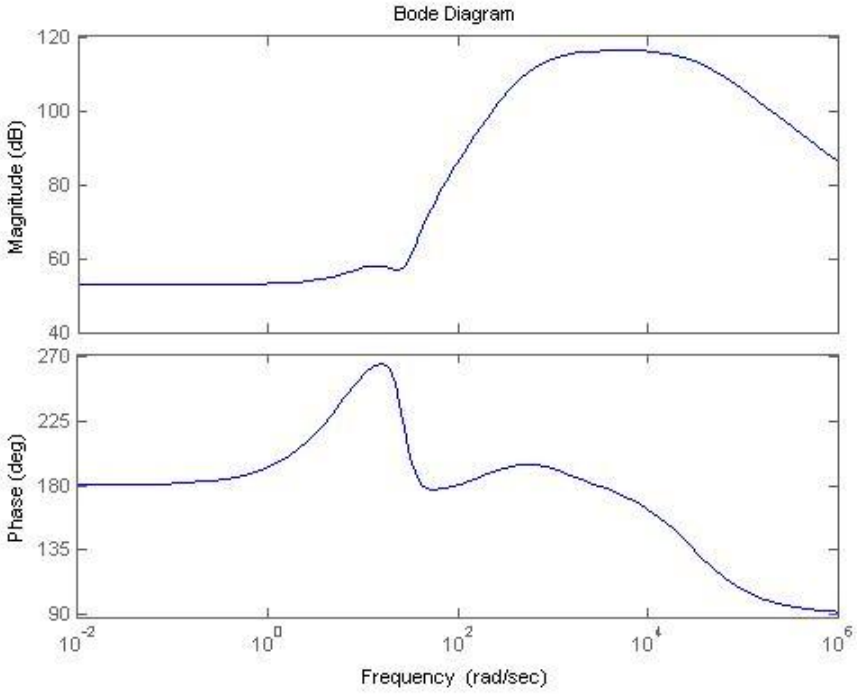


Figure II.10 μ-Synthesis 1 Controller Frequency Response

In order to obtain the proper $\mu < 1$, the performance criteria are relaxed. Specifically, the maximum displacement allowed on m_1 is raised to 0.041 m. The new performance criteria led to MATLAB outputting μ=0.9890. Figures 2.11 and 2.12 show the frequency responses of the closed-loop transfer function and the μ-controller, respectively. Similar to the H_∞ controller, the μ-controller has multiple unstable eigenvalues. Two of the controller's eighteen eigenvalues are unstable. This instability is acceptable because it leads to a stable closed-loop system.

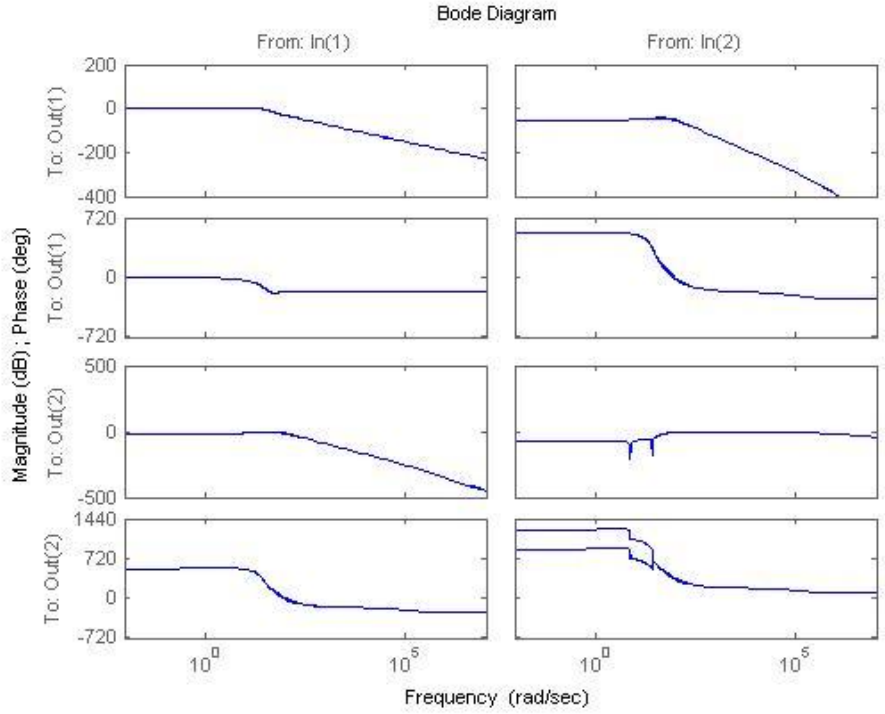


Figure II.11 μ-Synthesis 2 Closed-Loop Transfer Functions
Input 1: f_e Input 2: η_e Output 1: Displacement of m_1 Output 2: u

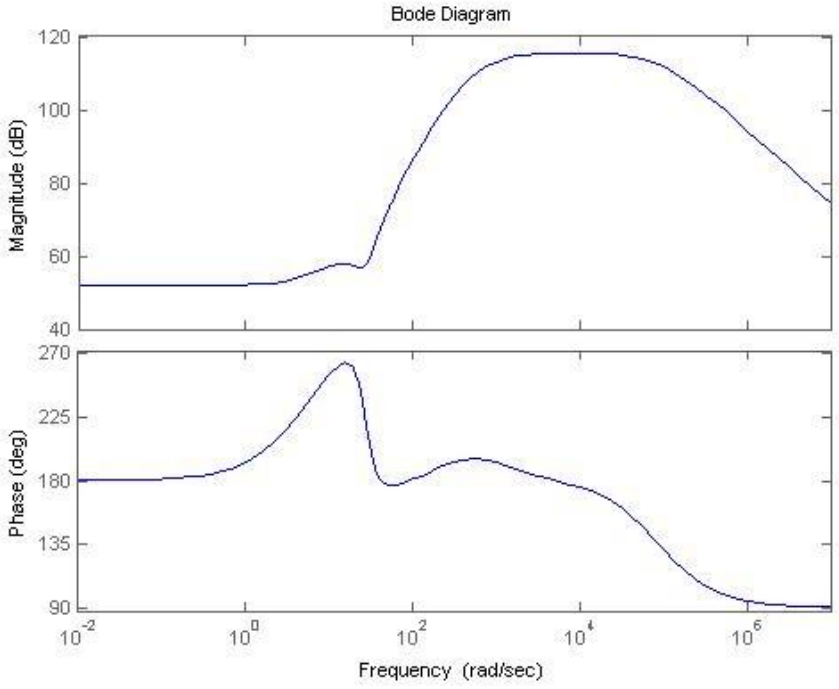


Figure II.12 μ-Synthesis 2 Controller Frequency Response

CHAPTER III

METHOD FOR EXTRACTING UNMODELED DYNAMICS

3.1 Introduction

This chapter will cover the development of the concepts which will be used in the application of model-based identification to find a system's unmodeled dynamics. First, the method of model-based identification will be derived for use with a μ-synthesis controller. Next, the three-mass model to be studied will be introduced. This includes a description of the test rig that will be used, a system identification experiment to define the parameters for the true three-mass system, and experimental verification of the true system. Finally, an example of unmodeled dynamics will be presented to utilize the three-mass model.

3.2 Application of Model-Based Identification for Missing Dynamics

This section presents the manner in which a μ-controller is applied to model-based identification in order to extract a system's unmodeled dynamics. To begin, a quick review of model-based identification is presented. The first control schematic, initially known as model reconciliation is shown below (Maslen, Vazquez and Sortore 2002):

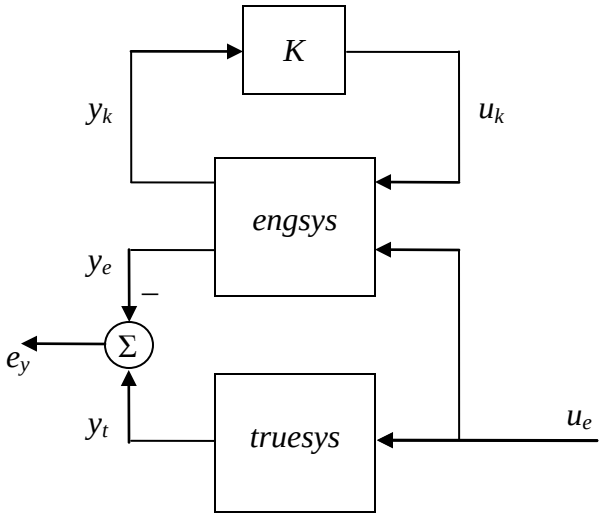


Figure III.1 Model Reconciliation Control Schematic

where:

truesys is the true system found from experimental data

engsys is the nominal engineering model derived from analytical techniques

u_e is the external input on the system

u_k is the control force

y_t is the measured output of the true system

y_e is the output of the engineering system under the control of K

y_k is the output to the controller

e_y is the error, or difference between the true and engineering system responses

K is the controller which drives the error to zero

Model-based identification reconciles the engineering model to the experimental true system by creating a controller, K, which minimizes the difference between the system responses. Model-based identification is unique in that it assumes that the structure of the engineering model is correct, while inherent inaccuracy is caused by unmodeled dynamics. This unmodeled dynamics results from portions of the system which do not have an accurate model, such as bearing seals, shrink fits, etc. Model-based identification seeks to correct the model by discovering this dynamics. First, the unmodeled dynamics is separated from the rest of the model. Next, the unmodeled dynamics is identified using the control schematic (Wang, Pettinato and Maslen 2009a):

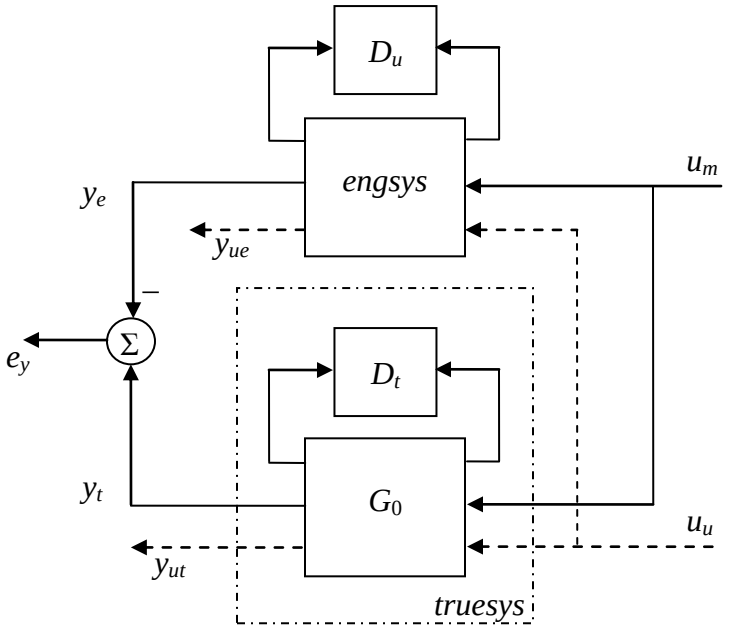


Figure III.2 Model-based Identification Control Schematic

With the addition of:

D_u is the unmodeled dynamics and the controller

D_t is the true unmodeled dynamics

u_u is the unmeasured input

u_m is the measured input

y_{ut} is the unmeasured output of the true system

y_{ue} is the unmeasured output of the engineering system

G_0 is the portion of the true system which is easy to model

Similar to the model reconciliation method, a controller is found which minimizes the error function e_y. The developed controller, D_u, becomes the unmodeled dynamics. The assumptions that the structure of the true system is known and $G_0 = truesys$, guarantee that $D_u = D_t$. Now, the controlled engineering system may use its unmeasured output y_{ue} to predict the true unmeasured output y_{ut} (Wang, Pettinato and Maslen 2009a).

This thesis utilizes a μ-controller and MATLAB's *dksyn* command in model-based identification. The reason for this choice is that once the control scheme is developed, it may be easily changed to reflect any changes in the performance criteria or other system requirements. In order to use the *dksyn* command, the control scheme in Figure 3.2 needs to be reconciled to the μ-control schematic shown in Figure 2.2. The following figure successfully blends the two diagrams into the control schematic for a method which will be known as μ-controlled model-based identification:

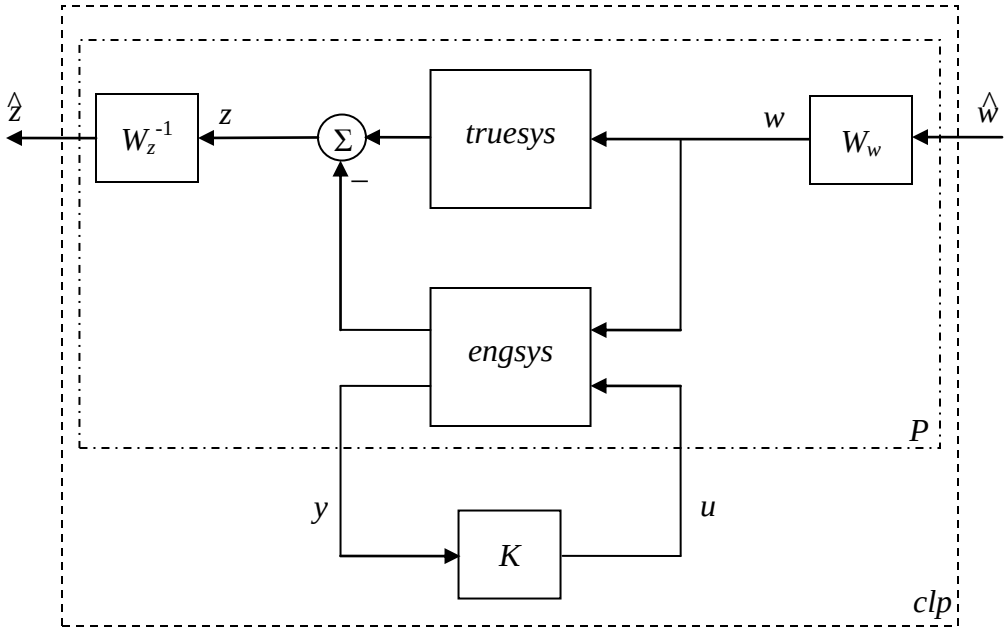


Figure III.3 Mu-Controlled Model-Based Identification Control Schematic

The controller plant *P* is partitioned in the following way:

$$\mathbf{A}=\begin{bmatrix}\mathbf{A}_t & \mathbf{0}\\ \mathbf{0} & \mathbf{A}_e\end{bmatrix}\quad \mathbf{B}_1=\begin{bmatrix}\mathbf{B}_t\\ \mathbf{B}_{e_1}\end{bmatrix}W_w\quad \mathbf{B}_2=\begin{bmatrix}\mathbf{0}\\ \mathbf{B}_{e_2}\end{bmatrix}\quad \mathbf{C}_1=W_z^{-1}\begin{bmatrix}\mathbf{C}_t & -\mathbf{C}_{e_1}\end{bmatrix}\quad \mathbf{C}_2=\begin{bmatrix}\mathbf{0} & \mathbf{C}_{e_2}\end{bmatrix}$$

where:

$\mathbf{A}_t$ is the dynamic matrix of the true system

$\mathbf{A}_e$ is the dynamic matrix of the engineering system

$\mathbf{B}_t$ is the state-space input matrix of the true system

$\mathbf{B}_{e_1}$ is the segment of the state-space input matrix of the engineering system corresponding to the disturbance input

$\mathbf{B}_{e_2}$ is the segment of the state-space matrix of the engineering system corresponding to the controller input

$\mathbf{C}_t$ is the state-space output matrix of the true system

$\mathbf{C}_{e_1}$ is the segment of the state-space output matrix of the engineering system corresponding to the disturbance output

$\mathbf{C}_{e_2}$ is the segment of the state-space output matrix of the engineering system corresponding to the controller output

Analogous to the previous control schematics of Figures 3.1 and 3.2, the primary objective of the μ-controlled model-based identification control schematic is to minimize the error between the true system and engineering system responses, *z*. This value is the difference between the responses of the true and engineering systems to the disturbance input, *w*. The response of the engineering system is driven by the controller, *K*, to minimize *z*. As a result, the combination of the controller and the engineering system should generate a response that matches that of the true system. To generate this behavior, the controller must match the unmodeled dynamics, or the difference between the dynamics of the two systems.

An important parameter in the control scheme is the output weight, W_z^{-1}. Explicitly, the weighting term W_z^{-1} is the inverse of the maximum allowable value of the error *z*. In the normal μ-synthesis control problem, the weighting is taken from a required performance of the system output. Conversely, model-based identification does not have a required performance. As a consequence, W_z^{-1} can be assigned any value; therefore it is utilized to drive the results. But first, this weighting needs to be started at some nominal value. The first application of the code leads to three cases of μ values: $\mu << 1.0$, $\mu > 1.0$, or $\mu \leq 1.0$but close to 1.0 within an acceptable error.

The first condition, $\mu << 1.0$, implies that W_z^{-1} is not stringent enough. This will result in a controlled engineering system whose response may not match that of the true system within an acceptable limit, left to be determined by the user. Naturally, μ-synthesis accepts this difference in the responses, because the controller meets all of its design requirements, performing within the weighting functions while keeping $\mu \leq 1.0$. A value of $\mu << 1.0$ also implies that the control effort can be significantly increased before violating the laws of H_∞ control. In order to increase the control effort, the output to be minimized, $\mathbf{C}_1$ can be made smaller by decreasing W_z. The value of W_z is decreased until the results become acceptably close, but it is more accurate to say that W_z is decreased until the μ-value closes in on one. This condition will be further illustrated in the example which follows at the end of chapter.

The second condition, $\mu > 1.0$, reveals that W_z^{-1} is too large. This μ-value tells the user that the control scheme is unable to keep the error, *z*, under the defined limit W_z. The code needs to be run again after increasing W_z. It is possible that no weighting value gives an acceptable error while keeping $\mu \leq 1.0$, meaning the systems may be ill-defined. This can occur when at least one of the engineering or true system models is inaccurate.

The third condition, $\mu \leq 1.0$ but close to 1.0 within an acceptable error, shows that the μ-controlled model-based identification scheme has been completed correctly and the controller represents the unmodeled dynamics, D_u.

3.3 The Three-Mass Model Study

3.3.1 Experimental Test Rig

The three-mass study in this chapter utilizes the Educational Control Products Model 210a Rectilinear Control System. Figure 3.4 shows the system in the test configuration and Figure 3.5 (Parks 1999) shows a labeled schematic of the system. The plant is a three-mass and spring setup, with the addition of an excitation motor, encoders, and a damper. The three masses are adjustable from 0 kg to 2 kg, nominally. Each mass rides on a carriage which slides on ball bearings. This carriage adds some weight and damping to the masses, the exact values of which will be determined in the system identification experiment in the following section. The maximum displacement of each mass is regulated by travel limit stops in each direction which contain electronic circuit breakers that shut down the system when triggered. Mass position data is read by the three optical encoders connected to the carriages. Two springs connect the masses, while a third connects the first mass (left-most in Figure 3.4) to ground. A dashpot is attached to the third mass (right-most in Figure 3.4). This dashpot is an air damper whose damping value may be changed by adjusting the damper's air flow valve.

The rectilinear system is excited by a brushless DC servo motor, which drives a rack and pinion. This pinion has a rigid connection to the first mass. A digital signal processor based real-time controller manages the operation of the motor. The signal processor also serves as a data acquisition. The controller board is installed into a personal computer which runs the executive program. This program gives the user access to real-time system data as well as various controllers and excitation. The experiment in this chapter utilizes both the step and sine sweep input trajectories.

Figure III.4 Educational Control Products Model 210a Rectilinear Control System

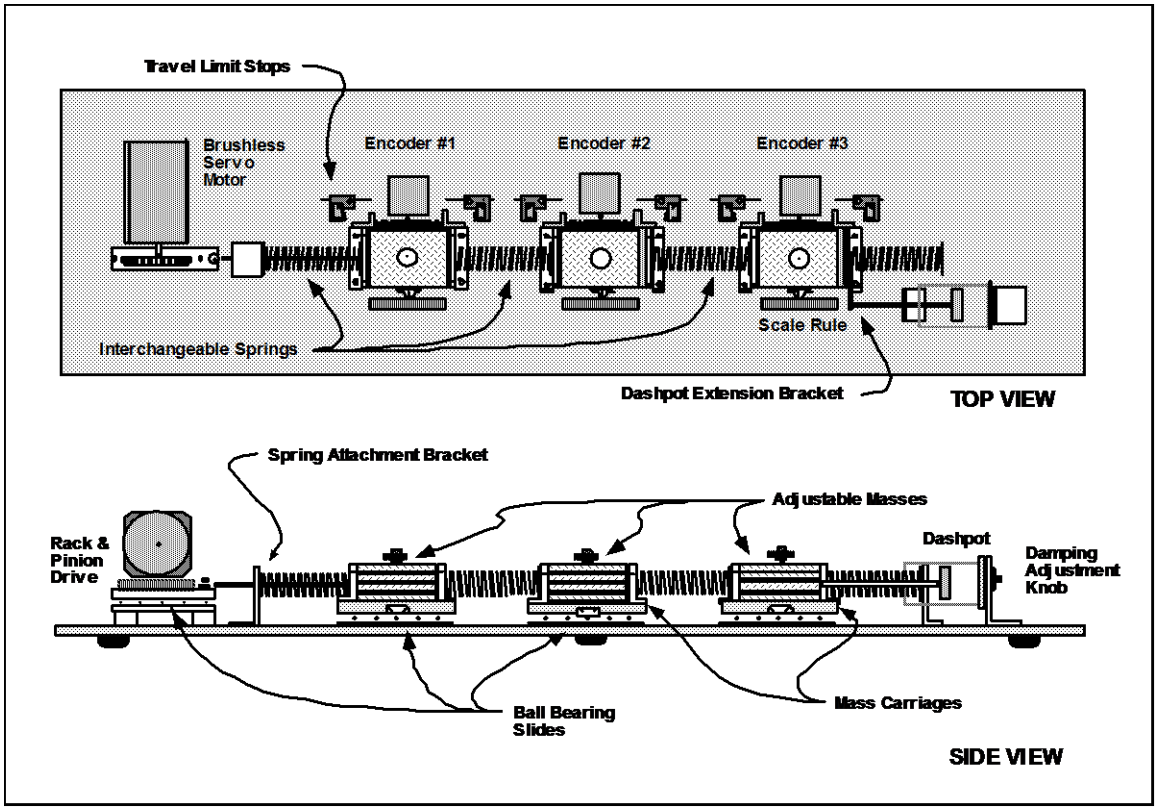


Figure III.5 Labeled ECP Model 210a Rectilinear Control System

3.3.2 System Identification Experiment

A system identification experiment is carried out in order to identify all of the system parameters required to fully define the three-mass system shown in Figure 3.6.

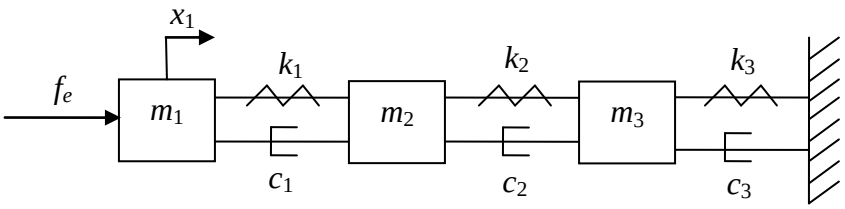


Figure III.6 Three-Mass System

First, m_1 is isolated from the rest of the system by using the travel limit stops to inhibit the motion of m_2 and m_3. The controller is set to input an open loop step function with a magnitude of 0 and duration of 3 seconds. This trajectory forces the data acquisition software to record 3 seconds of data without applying an input force. A displacement of roughly 2.5 cm is manually imparted on m_1, and the data acquisition records the response when the mass is released. This process is completed twice, first with the cart alone and next with four 500 g masses on the cart. The results of these two trials are shown below:

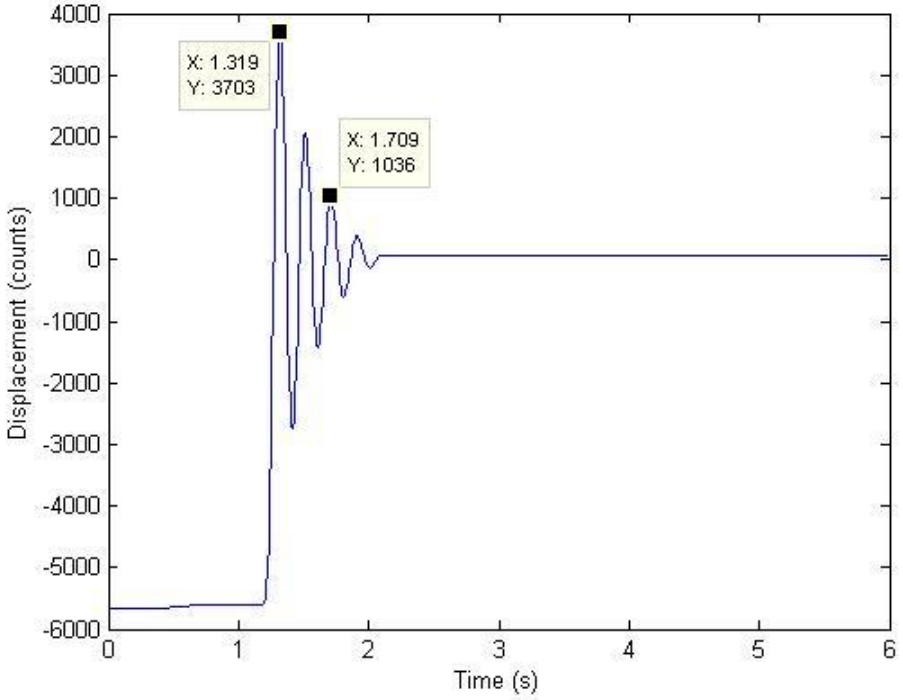


Figure III.7 Mass 1 Cart Response

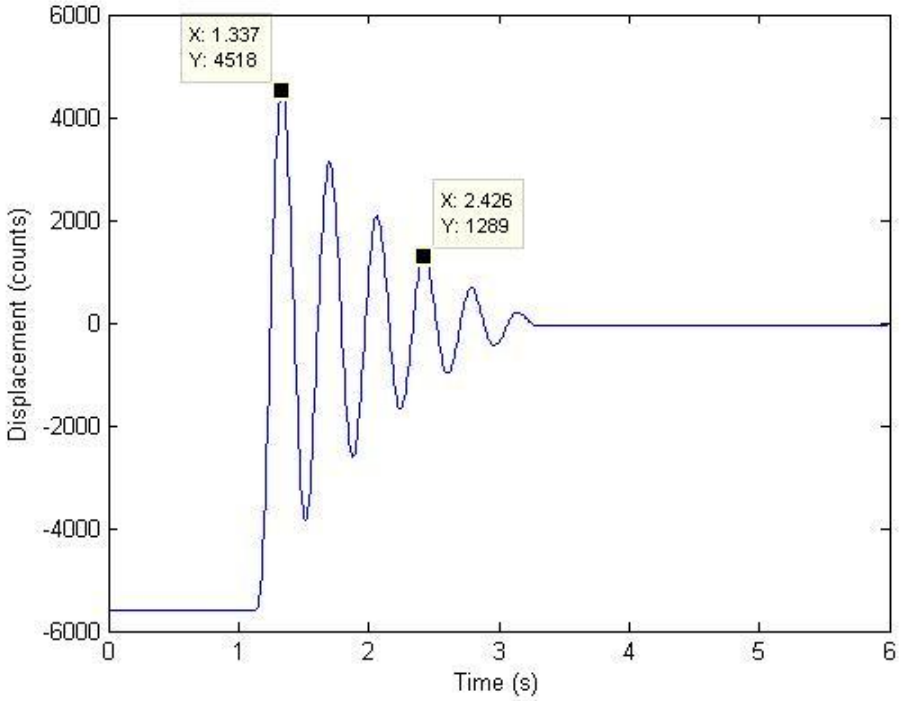


Figure III.8 Mass 1 Weighted Response

The natural frequencies of the two m_1 configurations are calculated using the peak data from the plots and the following equation:

$$\omega_n = \frac{N}{t_2 - t_1} 2\pi \tag{3.1}$$

Where N is the number of cycles between the two selected peaks and t_1 and t_2 are the times of the first and second peaks, respectively. Next, the damping ratio is calculated from the cart response with the following equation:

$$\zeta = \frac{1}{2\pi N} \log\left(\frac{y_1}{y_2}\right) \tag{3.2}$$

where y_1 and y_2 are the amplitudes of the first and second peaks, respectively. Then the mass of the cart is found with the following formula:

$$m_c = \frac{m_w \omega_{nc}^2}{\omega_{nc}^2 - \omega_{nw}^2} \tag{3.3}$$

Where m_w is the amount of weight added to the cart, ω_{nc} and ω_{nw} are the natural frequencies of the cart trial and the weighted trial, respectively. Next, equivalent stiffnesses on each mass are found using:

$$k_{1eq} = \omega_{nc}^2 m_c \tag{3.4}$$

Finally, the decoupled damping values may be found using the standard equivalence

$$c = 2\zeta m_c \omega_{nc} \tag{3.5}$$

This process is repeated for the second mass, third mass, and damper. Identification of the damper requires an additional set of data with the damper attached to the weighted

third cart. Solving for the air damper's damping ratio requires a modified decoupled damping equation:

$$c_d = 2(m_{c3} + m_w)\zeta_d \omega_{nd} - c_{m3} \tag{3.6}$$

Once all of the trials have been completed, off-diagonal stiffness values are found by comparing the equivalent stiffnesses. Results of the system identification experiment are summarized below in Table 3.1.

Table I. System Identification Results

System Parameters	Equivalence	Identified Values	Parameter Value
m_1	$m_1 = m_{c1} + m_w$	$m_w = 2.0\,\text{kg}$ $m_{c1} = 0.8113\,\text{kg}$	$m_1 = 2.8113\,\text{kg}$
m_2	$m_2 = m_{c2} + m_w$	$m_{c2} = 0.6396\,\text{kg}$	$m_2 = 2.6396\,\text{kg}$
m_3	$m_3 = m_{c3} + m_w$	$m_{c1} = 0.6074\,\text{kg}$	$m_3 = 2.6074\,\text{kg}$
k_1	$k_1 = k_{1eq}$	$k_{1eq} = 842.3\,\text{N/m}$	$k_1 = 842.3\,\text{N/m}$
k_2	$k_2 = k_{2eq} - k_1$	$k_{2eq} = 1648.8\,\text{N/m}$	$k_2 = 806.5\,\text{N/m}$
k_3	$k_3 = k_{3eq} - k_2$	$k_{3eq} = 1192.5\,\text{N/m}$	$k_3 = 386\,\text{N/m}$
c_1	$c_1 = c_{m1}$	$c_{m1} = 5.2992\,\text{Ns/m}$	$c_1 = 5.2992\,\text{Ns/m}$
c_2	$c_2 = c_{m2}$	$c_{m2} = 1.9467\,\text{Ns/m}$	$c_2 = 1.9467\,\text{Ns/m}$
c_3	$c_3 = c_{m3} + c_d$	$c_{m3} = 1.9297\,\text{Ns/m}$ $c_d = 8.8078\,\text{Ns/m}$	$c_3 = 10.7375\,\text{Ns/m}$

3.3.3 Model Validation

In order to confirm the parameter values found in the system identification experiment, two sine sweeps of the system are completed for comparison. The first sweep is of the simulated model. This model is created by inserting the values from the system identification experiment of the previous section into a state-space representation of Figure 3.6. The second sine sweep is found from direct measurement of the Rectilinear

Control System utilizing the included Executive Software. An open-loop, logarithmic sine sweep is used as the trajectory input with a frequency range of 0.1-10 Hz, an input force amplitude of 0.3 V, and a running time of 29.5 s. Next, the two sine sweeps are plotted together, as shown in Figure 3.9. The agreement of the experimental and simulated sine sweeps confirms the accuracy of the state-space three-mass model. Consequently, this state-space model will be used as the experimental true system during the three-mass example of unmodeled dynamics in the following section.

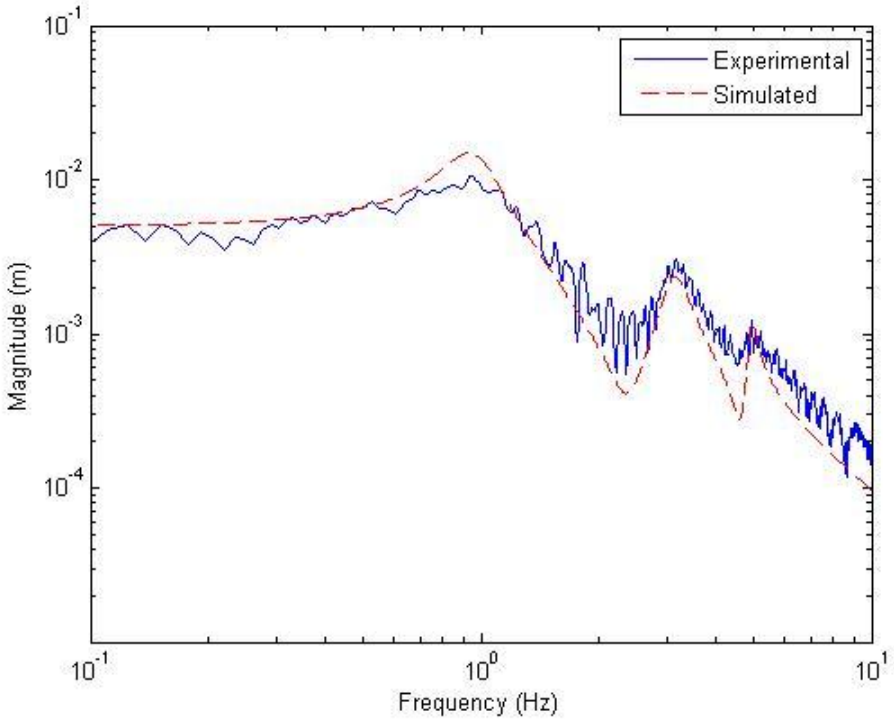


Figure III.9 Three-Mass Sine Sweeps

3.4 Three-Mass Example of Unmodeled Dynamics

A three-mass example is now presented to illustrate the method of using model-based identification to discover unmodeled dynamics. The purpose of this example is to show how the method may be applied to a physical system to discover dynamics which have been left out of the engineering model. In all of the trials in this section, the unmodeled dynamics will be known, so the accuracy of the method may be confirmed. An additional goal of this section is to refine the model-based algorithm first laid out by Vazquez et al. (2003) in order to make the method easier to apply to systems in which the unmodeled dynamics is not already known.

3.4.1 Problem Description

In this problem, the third mass will be associated with various forms of unknown dynamics. Either mass, stiffness, damping, or some combination of the three parameters of the third mass will be unknown. The true system will be the known from experimental results and the engineering system will be associated with a "guessed" nominal model. Therefore, the new variables m_t, k_t, c_t, m_g, k_g, and c_g are introduced to describe the mass, stiffness, and damping of the third mass in the true and engineering systems, respectively. All shared parameters and true values are found in Table 3.1. True parameters are also given an initial uncertainty of one percent which represents possible inaccuracy in test measurements. For example, the true system and engineering system models for the first trial are shown below:

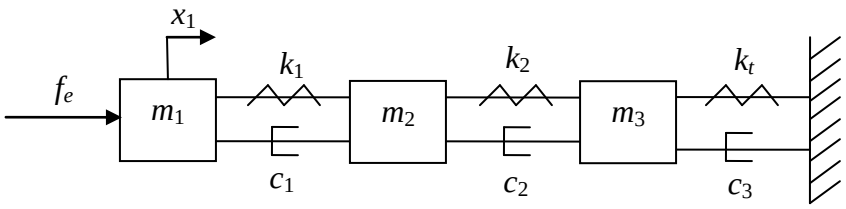


Figure III.10 Trial 1 True System

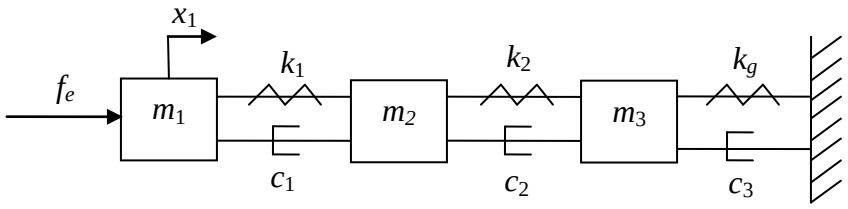


Figure III.11 Trial 1 Engineering System

In the first trial, the stiffness of the third spring is the only unknown value. As shown in the figures, each system has an external input and output acting on m_1. The unmodeled dynamics appears on the third mass of the engineering system, so this is where the controller K is placed. As a result, the control objective becomes: minimize the error between the m_1 responses of the true and engineering systems by modifying the behavior of m_3 in the engineering system. The resulting control schematic for the three-mass model-based identification is shown below:

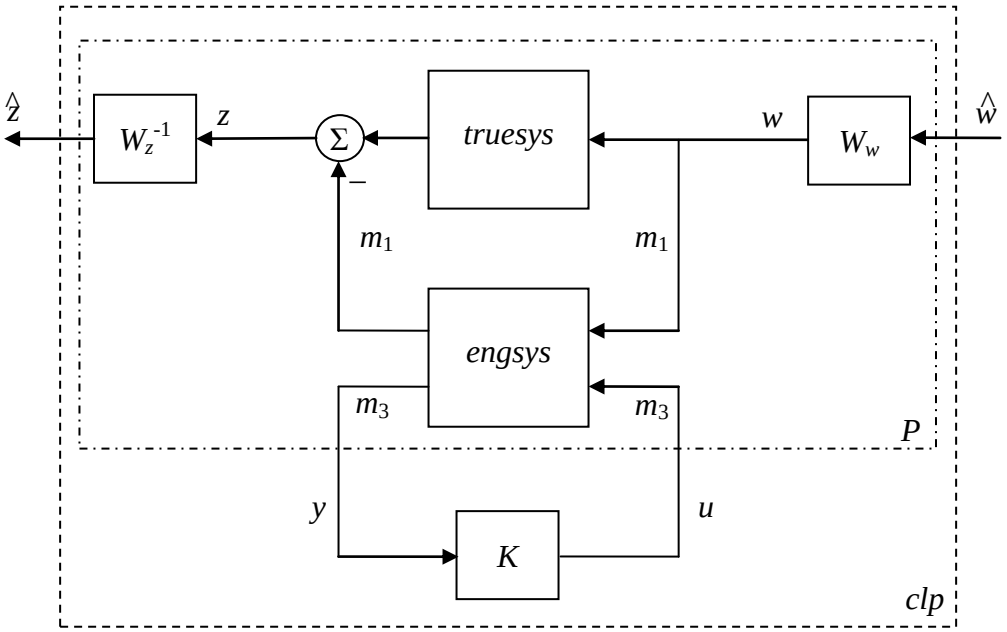


Figure III.12 Three-Mass Example Mu-Controlled Model-Based Identification Schematic

The resulting controller plant is:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_t & 0 \\ 0 & \mathbf{A}_e \end{bmatrix} \quad \mathbf{B}_1 = \begin{bmatrix} \mathbf{B}_{m_1} \\ \mathbf{B}_{m_1} \end{bmatrix} W_w \quad \mathbf{B}_2 = \begin{bmatrix} 0 \\ \mathbf{B}_{m_3} \end{bmatrix} \quad \mathbf{C}_1 = W_z^{-1} \begin{bmatrix} \mathbf{C}_{m_1} & -\mathbf{C}_{m_1} \end{bmatrix} \quad \mathbf{C}_2 = \begin{bmatrix} 0 & \mathbf{C}_{m_3} \end{bmatrix}$$

where:

$\mathbf{A}_t$ is the dynamic matrix of the true system

$\mathbf{A}_e$ is the dynamic matrix of the engineering system

$\mathbf{B}_{m_1}$ is the m_1 input selector matrix

$\mathbf{B}_{m_2}$ is the m_2 input selector matrix

$\mathbf{C}_{m_1}$ is the m_1 output selector matrix

$\mathbf{C}_{m_3}$ is the m_3 output selector matrix

It is important to note that all of the selector matrices are weighted by their associated mass.

3.4.2 Results

Trial 1a

The first trial models are shown above in Figures 3.10 and 3.11. The only difference between the two systems is the stiffness of the spring attached to the third mass. The engineering system contains the guessed spring stiffness, $k_g = 200\,\text{N/m}$, while the true system has the known spring stiffness, $k_t = 386\,\text{N/m}$ with an uncertainty of one percent. Once the code is run, the controller should resemble a spring with a stiffness of 186 N/m, the difference between the engineering and true systems.

The code is run and the final results are that when W_z=0.0004, $\mu \le 0.9337$. Results of the first trial are shown below in Figures 3.13-3.15. Figure 3.13 shows the responses of the *truesys*, *engsys*, and the controlled engineering system, *controlled engsys*, in which *engsys* is connected to *K* by a lower LFT. Figure 3.14 shows the frequency responses of the controller *K* and the known unmodeled dynamics, D_u. The known unmodeled dynamics is a transfer function found from the difference in the dynamics of the third mass:

$$D_u = D_t - D_e \tag{3.7}$$

With:

$$D_t = m_n s^2 + c_n s + k_n \tag{3.8}$$

$$D_e = m_g s^2 + c_g s + k_g \tag{3.9}$$

Where D_t and D_e are the dynamics of the true and engineering systems, respectively, and the subscript "$_n$" represents the true parameter values with no uncertainty, or the nominal true values. For Trial 1a, Equation 3.7 simplifies to:

$$D_u = k_n - k_g = 186\,\text{N/m} \tag{3.10}$$

Finally, Figure 3.15 shows the closed-loop response of the control schematic shown in Figure 3.12. This figure offers visual proof that the controller is working, and has met the control objective of keeping the weighted output $\hat{z} < 1.0$ for all frequencies. An important feature to note is the appearance of multiple plots for the value of $\hat{z}$. The cause of this behavior is uncertainty in the true system. Multiple plots will also appear in system frequency response plots in later trials for the same reason.

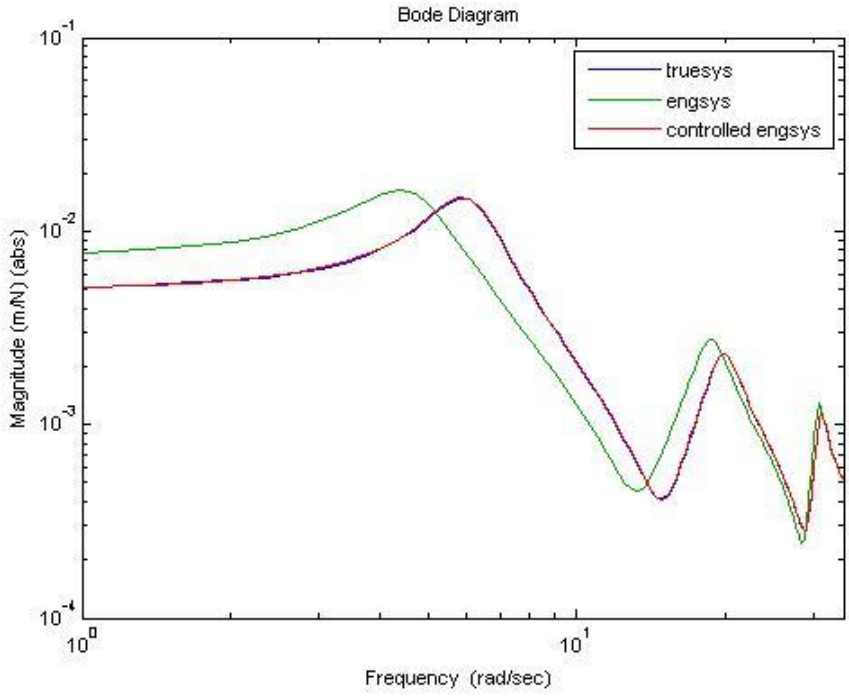


Figure III.13 Trial 1a System Frequency Responses

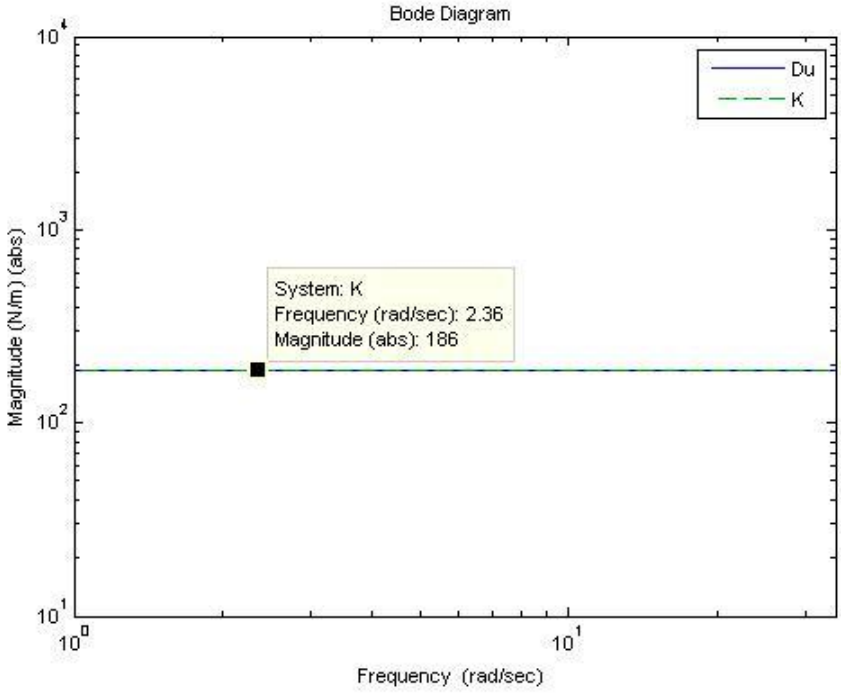


Figure III.14 Trial 1a Controller Response

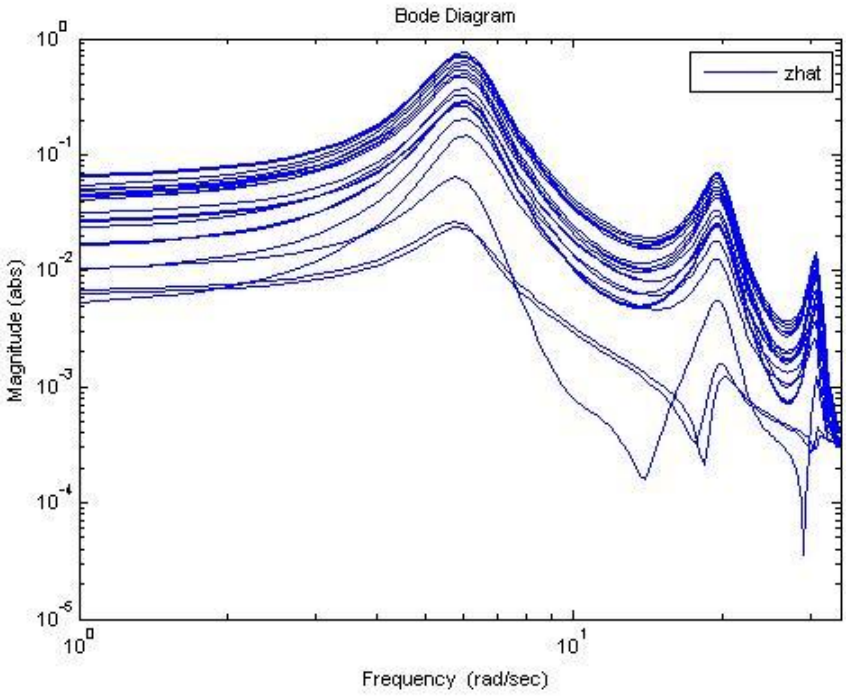


Figure III.15 Trial 1a Closed-Loop Response

The preceding three figures show that μ-controlled model-based identification was successful in determining the unmodeled spring dynamics. First, Figure 3.13 shows that the *controlled engsys* response matches the *truesys* response. Next, Figure 3.14 shows that *K* matches D_u across the entire frequency range, with the controller acting as a spring with a stiffness of 186 N/m, matching the nominal difference. Finally, Figure 3.15 shows that $\hat{z} < 1.0$ for all frequencies.

Trial 1b

Similar to Trial 1a, Trial 1b also seeks to identify a difference in stiffness between the true and engineering systems. The difference is that this trial models a decrease in stiffness between the engineering and true system. Parameter values of $k_g = 386\,\text{N/m}$ and $k_n = 200\,\text{N/m}$ are used in Figures 3.10 and 3.11, leading to the simplification of Equation 3.7:

$$D_u = k_n - k_g = -186\,\text{N/m} \tag{3.11}$$

The algorithm returns a value of $\mu \leq 0.8930$ when W_z=0.0004. Resulting system responses are shown in Figure 3.16. Interestingly, the controller frequency response matches that in Figure 3.14. In order to discover a difference between these controllers, the phase responses are included in the Bode plots shown in Figures 3.17 and 3.18. These Bode plots show that the only difference between an increase in stiffness and a reduction is a 180 degree phase shift.

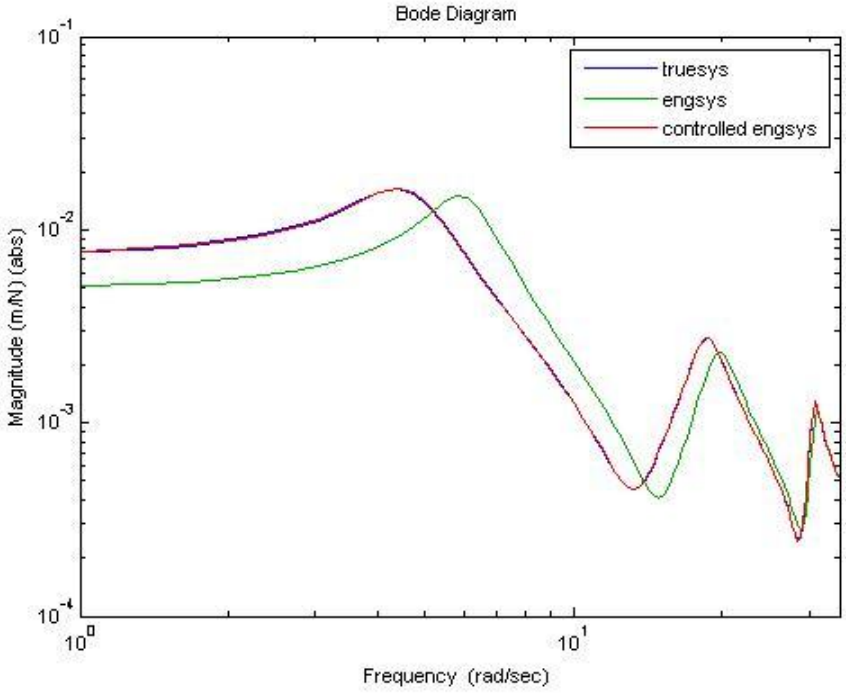


Figure III.16 Trial 1b System Frequency Responses

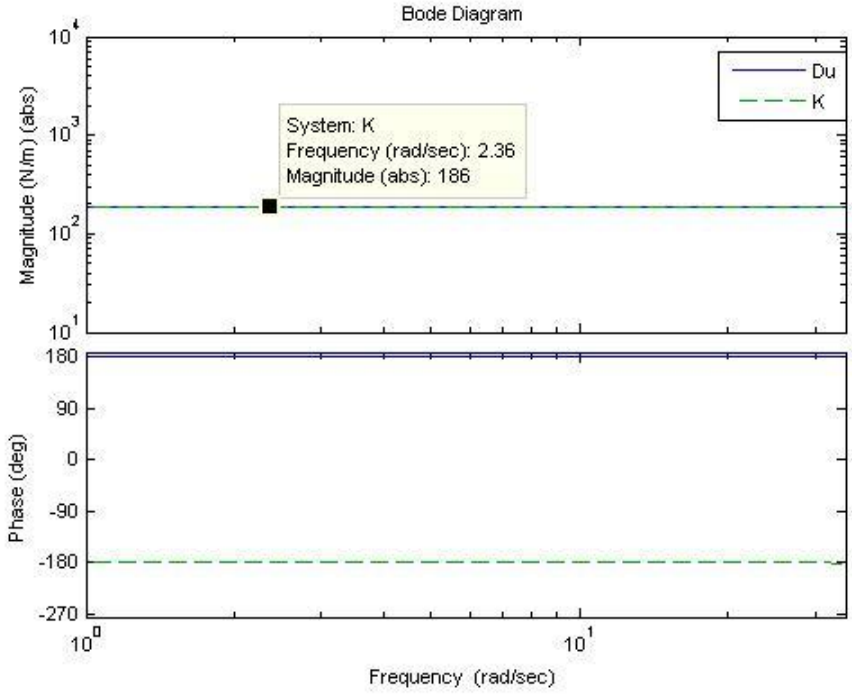


Figure III.17 Trial 1a Controller Bode Plot

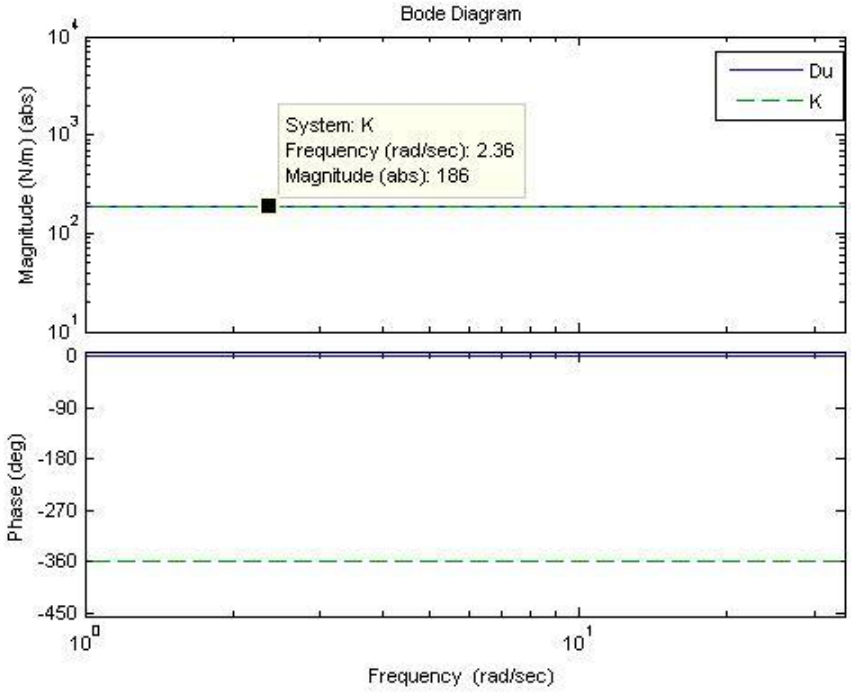


Figure III.18 Trial 1b Controller Bode Plot

Trial 2

The second trial seeks to identify a difference in mass between the true and engineering systems. The guessed mass, m_g, has a value of 1.6074 kg, while the true mass, m_t, has a nominal value of 2.6074 kg with an uncertainty of one percent. The system models are shown in Figures 3.19 and 3.20. For this trial, Equation 3.7 simplifies to:

$$D_u = (m_n - m_g)s^2 = s^2 \text{ N/m} \tag{3.12}$$

A result of $\mu \leq 0.9492$ is obtained when W_z=0.000091. The controller acts as predicted, giving a transfer function that is equal to the mass difference times the

frequency squared. Results generated by this version of the code are of similar quality to Trial 1 and are shown in Figures 3.21-3.23.

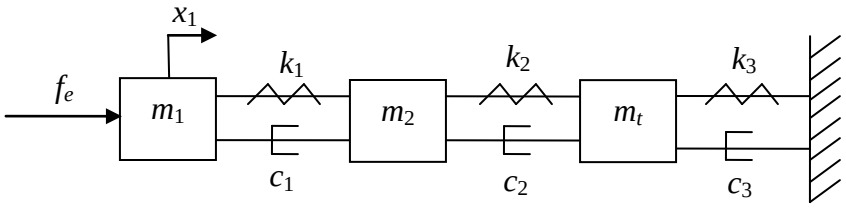


Figure III.19 Trial 2 True System

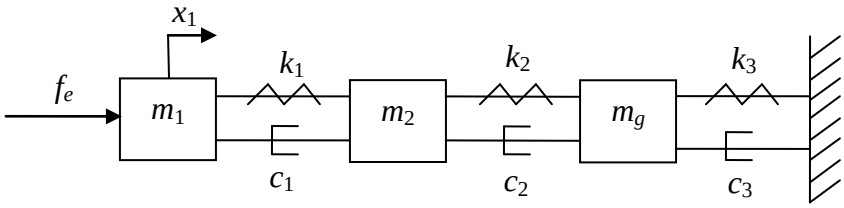


Figure III.20 Trial 2 Engineering System

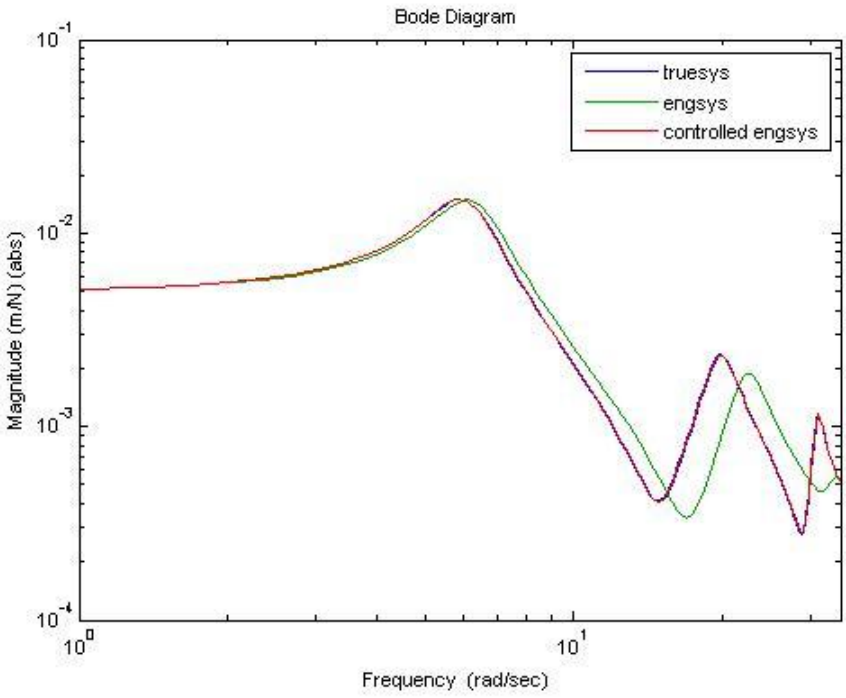


Figure III.21 Trial 2 System Frequency Responses

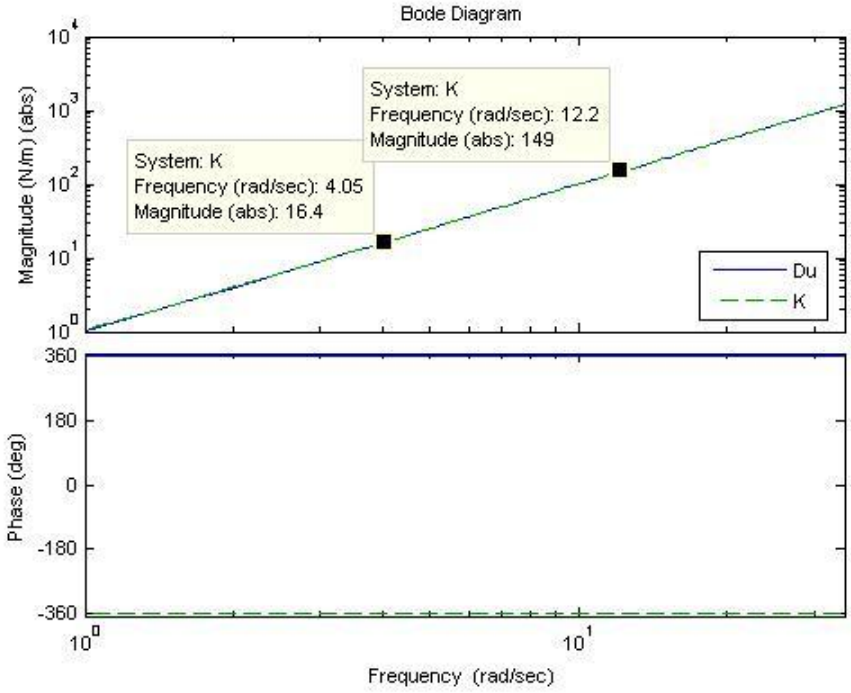


Figure III.22 Trial 2 Controller Response

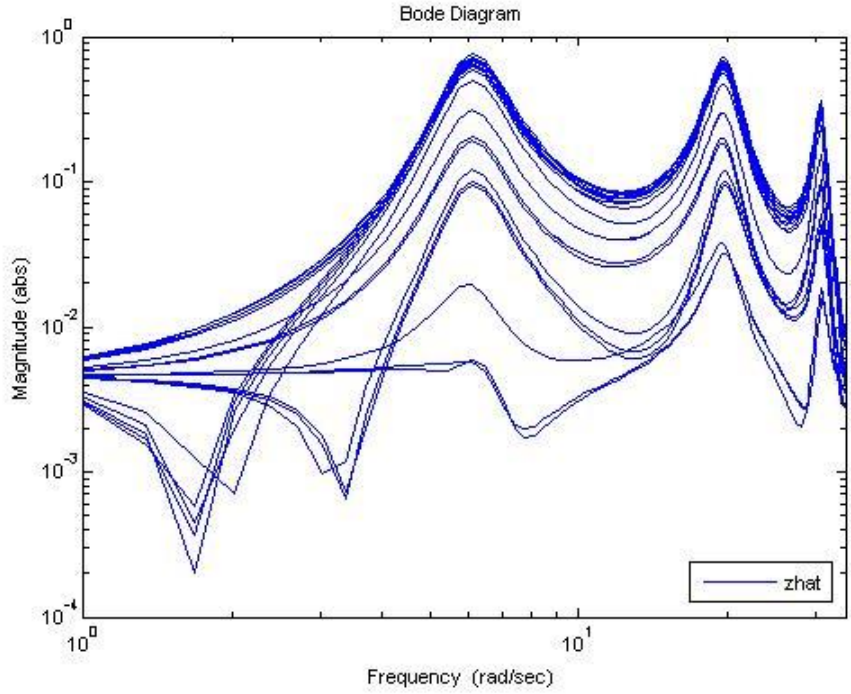


Figure III.23 Trial 2 Closed-Loop Response

Trial 3

The third trial seeks to identify a difference in damping between the true and engineering systems. The guessed and true damping values are c_g=5.7375 N*s/m and c_t=10.7375 N*s/m, respectively, with the true damping having a one percent uncertainty. For this trial, Equation 3.7 simplifies to:

$$D_u = (c_n - c_g)s = 5s \text{ N/m} \tag{3.13}$$

The final iteration gives $\mu \leq 0.9816$ with the maximum error, W_z=0.00006. Once again, the results indicate that the method is successful. Figures 3.24 and 3.25 show the system models and the graphical output is shown in Figures 3.26-3.28.

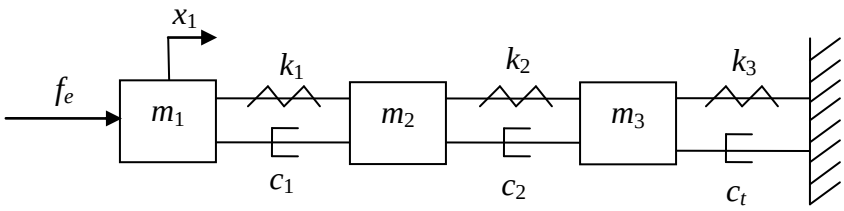


Figure III.24 Trial 3 True System

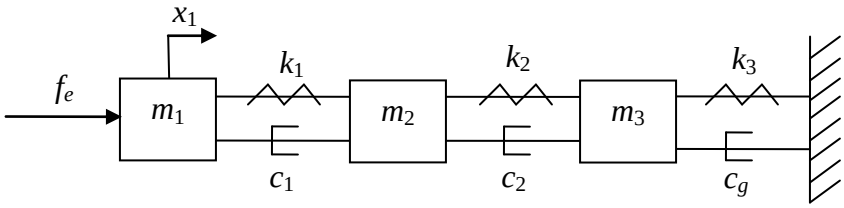


Figure III.25 Trial 3 Engineering System

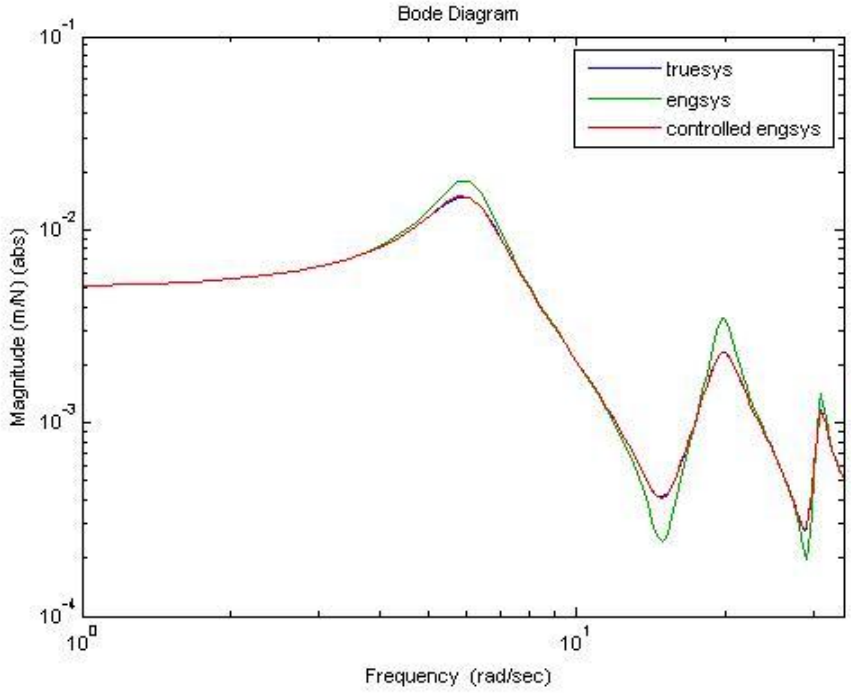


Figure III.26 Trial 3 System Frequency Responses

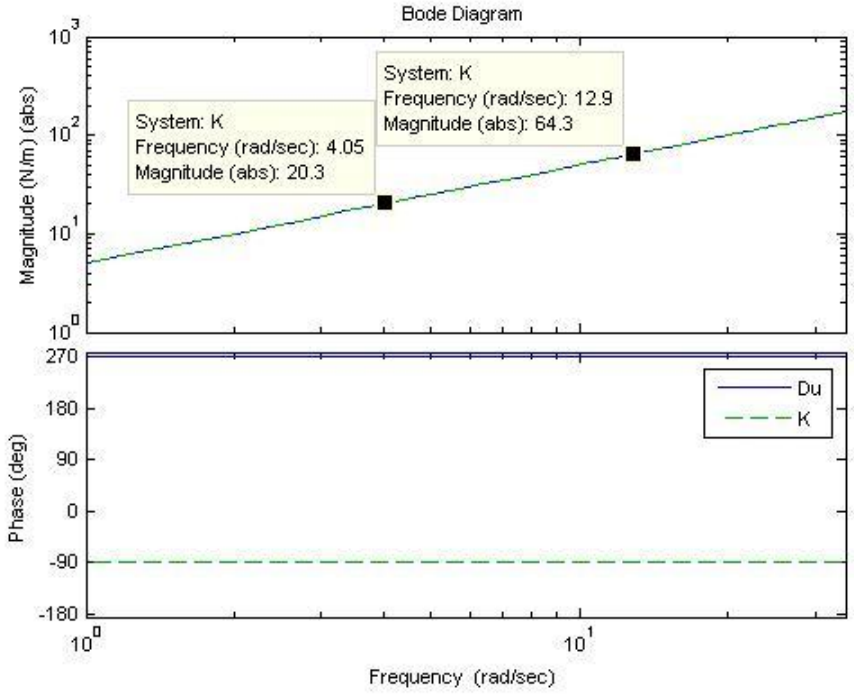


Figure III.27 Trial 3 Controller Response

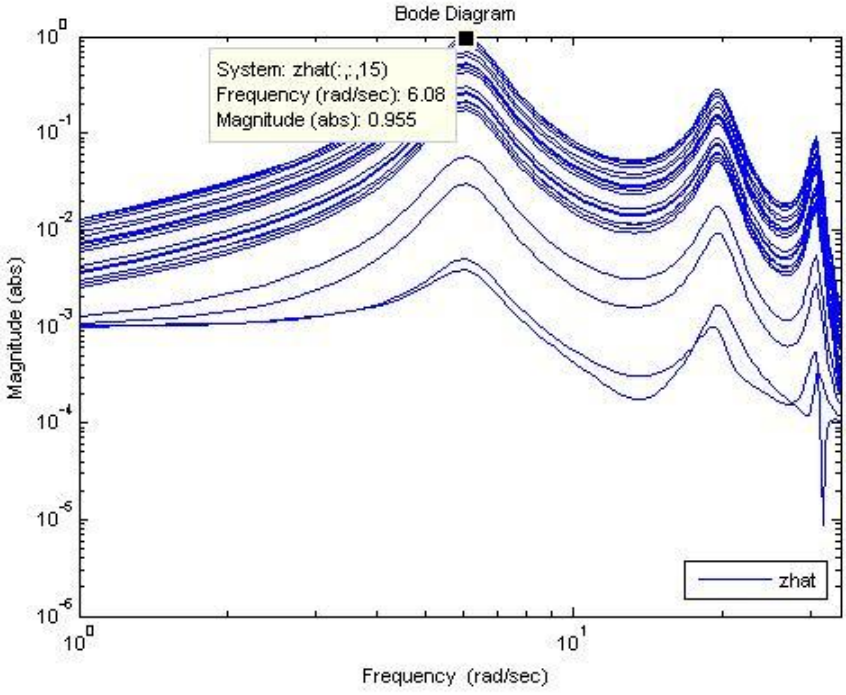


Figure III.28 Trial 3 Closed-Loop Response

Trial 4

Trials 1 through 3 proved that the method was able to handle a single-parameter difference in dynamics. Now, the method is tested with a system with an unmodeled mass, stiffness, and damping. Similar to the first three trials, the guessed parameters have values of m_g=1.6074 kg, k_g=200 N/m, and c_g=5.7375 N*s/m, while the true parameters have values of m_t=2.6074 kg, k_t=386 N/m, and c_t=10.7375 N*s/m with each parameter having a one percent uncertainty. Using these values, Equation 3.7 simplifies to:

$$D_u = (m_n - m_g)s^2 + (c_n - c_g)s + k_n - k_g = s^2 + 5s + 186 \tag{3.14}$$

The final iteration gives $\mu \leq 0.9881$ with the maximum error, W_z=0.0005. The system model and results are shown in the following figures:

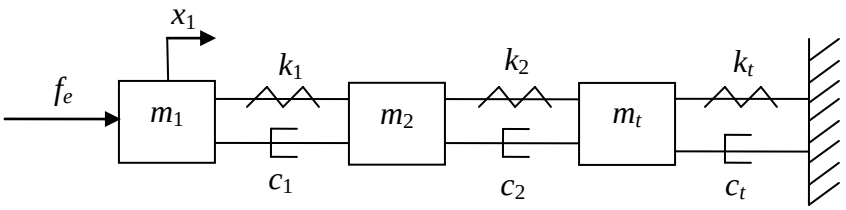


Figure III.29 Trial 4 True System

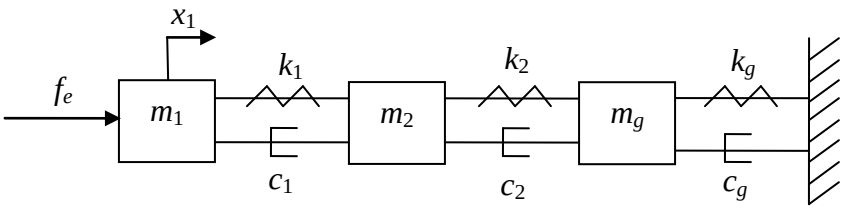


Figure III.30 Trial 4 Engineering System

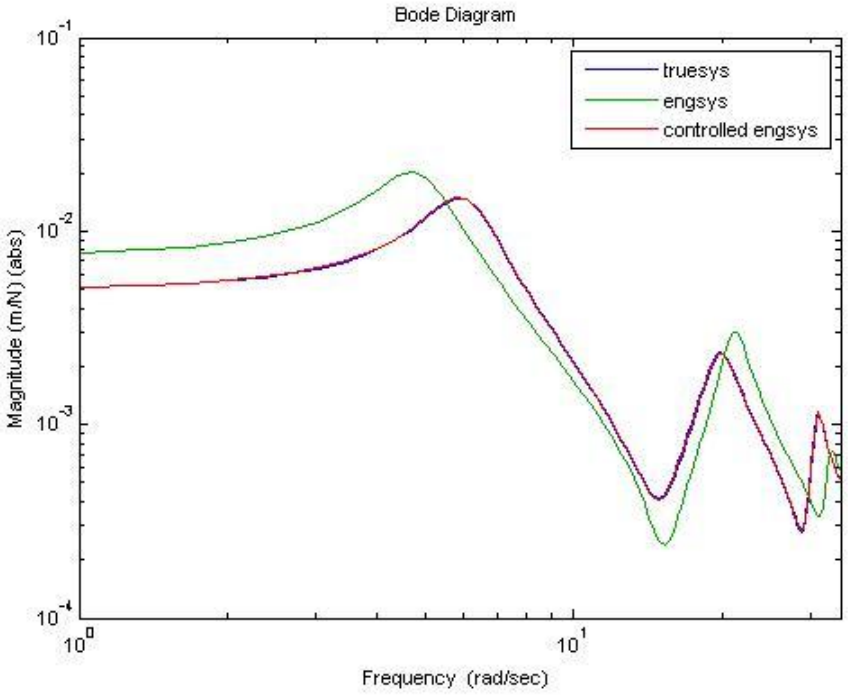


Figure III.31 Trial 4 System Frequency Responses

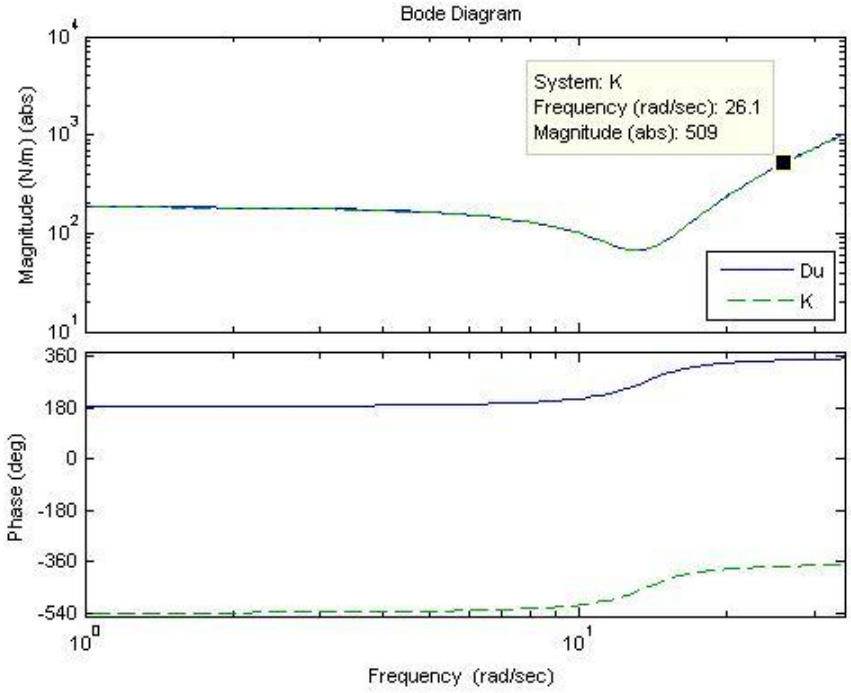


Figure III.32 Trial 4 Controller Response

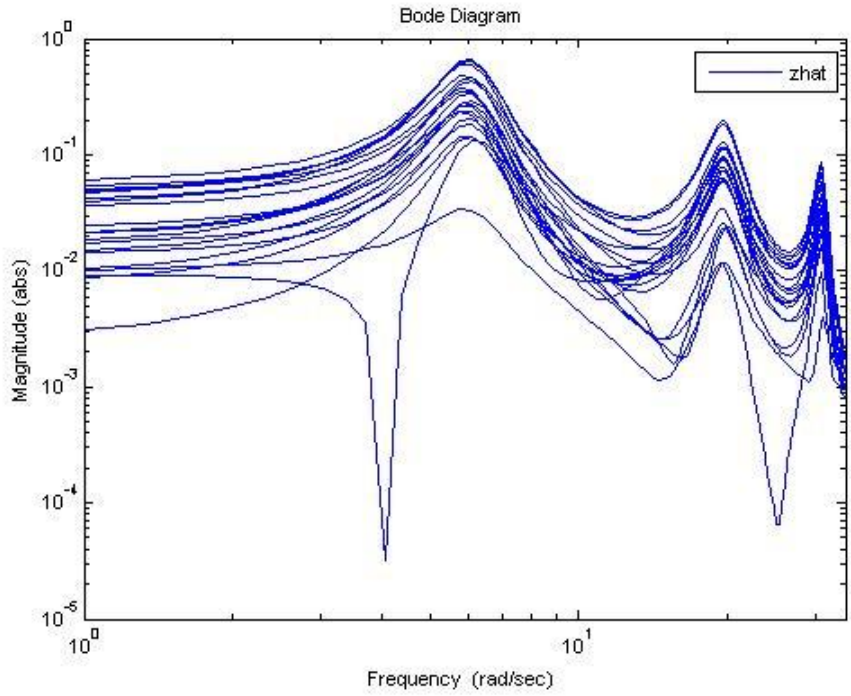


Figure III.33 Trial 4 Closed-Loop Response

Trial 5

Finally, the robustness of μ-controlled model-based identification is evaluated. This is accomplished by running the same experiment as in Trial 4 except with the uncertainty on each true parameter increased from one percent to ten percent. The method is able to handle this increased uncertainty, with the final iteration giving $\mu \leq 0.9846$ with an error weighting W_z=0.0026. Graphical results are shown below:

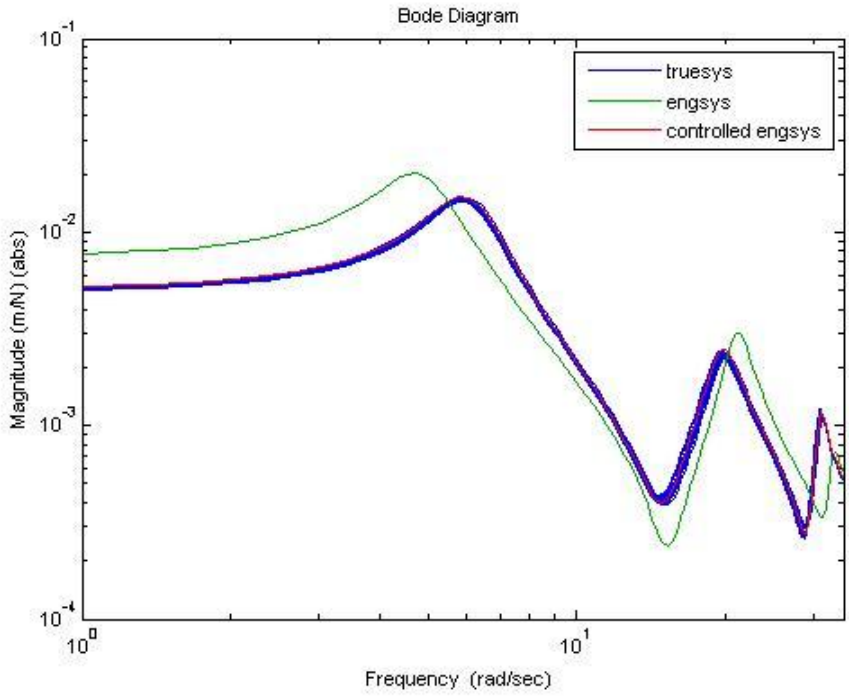


Figure III.34 Trial 5 System Frequency Responses

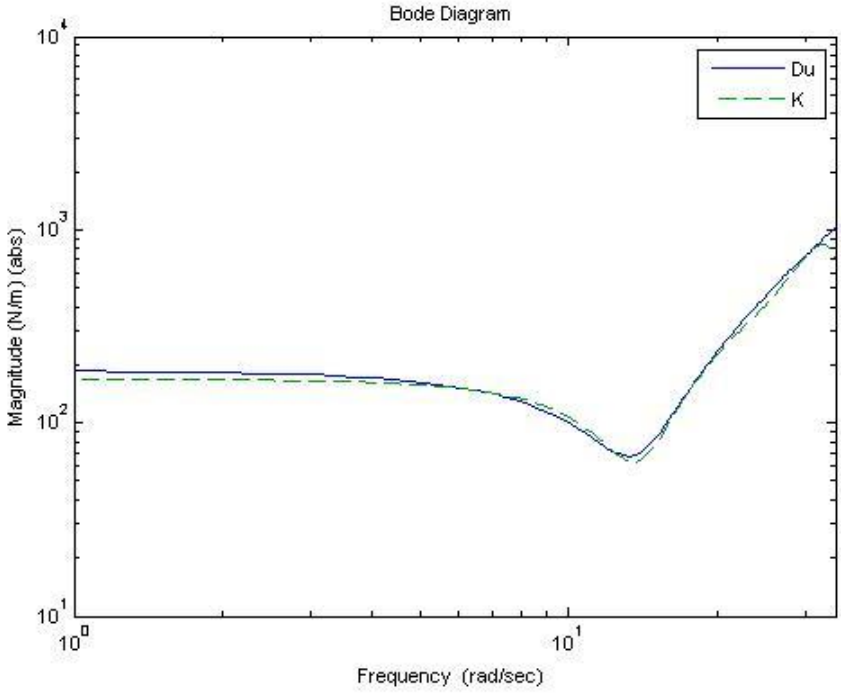


Figure III.35 Trial 5 Controller Response

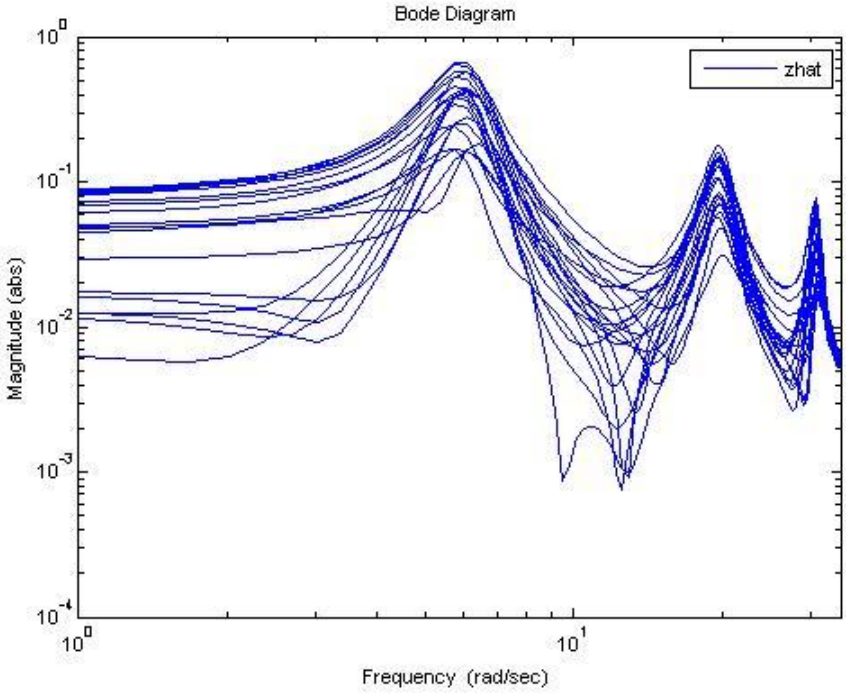


Figure III.36 Trial 5 Closed-Loop Response

3.5 Conclusions

Comparing the results of the five three-mass μ-controlled model-based identification trials leads to some interesting conclusions. First, and most importantly, the method works in creating a controlled engineering system which matches the response of the true system for various system configurations. Consequently, a controller is synthesized which accurately depicts the unmodeled dynamics. The next conclusion is that uncertainty has a noticeable effect on the accuracy of the method. Figures 3.32 and 3.35 may imply that this effect is negligible for this system, but closer analysis of the results provides more insight. A major discrepancy in the weighting values can be seen between Trials 4 and 5, with $W_{z_4} = 0.0005$ and $W_{z_5} = 0.0026$. These weights indicate that the controller in Trial 5 can only hold the systems to within 2.6 mm, while Trial 4's controller is able to keep the systems within 0.5 mm. This difference may not alter the three-mass results significantly, but it does lead to the final conclusion.

The final conclusion from the three-mass trials is that the weighting function and μ-value of the final iteration together can be used as a performance measure of model-based identification. As explained earlier, the value of W_z is decreased to drive the μ-value to one. This weighting value indicates the largest possible discrepancy between the controlled engineering system and the true system. The final value of W_z indicates very clearly how well the model-based identification algorithm has performed. This correlation is shown in the final error values of the fourth and fifth trials along with the accuracies of the controllers shown in Figures 3.29 and 3.32.

This new performance measure leads to an update to the model-based identification algorithm presented in the work by Vazquez et al. (2003), with the addition of the seventh step:

1. Measure the frequency response of the actual system.
2. Identify a state-space representation (*truesys*) from the experimental frequency response.
3. Create a nominal engineering model (*engsys*).
4. Identify uncertain portions of the model.
5. Assemble the control schematic shown in Figure 3.3.
6. Run the μ-synthesis, resulting in *K*.
7. Evaluate the μ-value:
   A. If $\mu << 1.0$, decrease W_z and repeat the μ-synthesis.
   B. If $\mu > 1.0$, increase W_z and repeat the μ-synthesis.
   C. If $\mu \leq 1.0$ within a user-defined acceptable error, then *K* represents the unmodeled dynamics.

The addition of step 7 to the algorithm allows for an easier transition to a system with unknown unmodeled dynamics. Once there is no defined D_u to evaluate *K* against, it is advantageous to have a measure of exactly how close the system responses are to each other. Now that the method has been proven and refined, it is ready to be applied to a system with unknown unmodeled dynamics, which will be presented in Chapter 4.

CHAPTER IV

APPLICATION TO STRUCTURAL DAMAGE DETECTION IN ROTATING MACHINERY

4.1 Introduction

In this chapter, the method of utilizing model-based identification for quantifying unmodeled dynamics will be applied to create a new model for the effects of a transverse crack on a rotor. This example will utilize the experimental responses of a healthy and a cracked rotor as measured and reported by Pesch (2008). First, the crack detection rig will be introduced. Next, an explanation of the technique used to derive the true and engineering system models from the experimental data. After the models are derived, the model-based identification approach is developed for discovering the dynamics brought

on by a crack in a rotor. Finally, the experiment is carried out and the results are presented.

4.2 Crack Detection Test Rig

The test rig utilized in this experiment was manufactured by SKF Magnetic Bearings, which is a unit of SKF Canada Limited. Figure 4.1 shows the experimental setup and Figure 4.2 shows the dimensions of points of interest on the rotor (Pesch 2008). This test rig consists of a carbon steel disk and three magnetic bearing rotors mounted on a 26 in long, 0.625 in diameter, 416 stainless steel shaft. The rotor is driven by a 48 volt DC brush-type motor, connected by a flexible coupling which allows axial and radial displacement. All of these components are mounted on an aluminum base plate which sits on a Technical Manufacturing Corporation 780 Series Vibration Isolation Optical Table.

Excitation force for the experimental sine sweeps is input through the middle magnetic force actuator. Both the motor and the exciter are controlled by SKF model MB340g4-ERX digital controllers. The motor is controlled by a PI controller, while the exciter has no active feedback control. The controllers are programmed using the MBScope2000 version 4.07 software package which is installed on a PC. This software package programmed the experimental sine sweeps and collected the data.

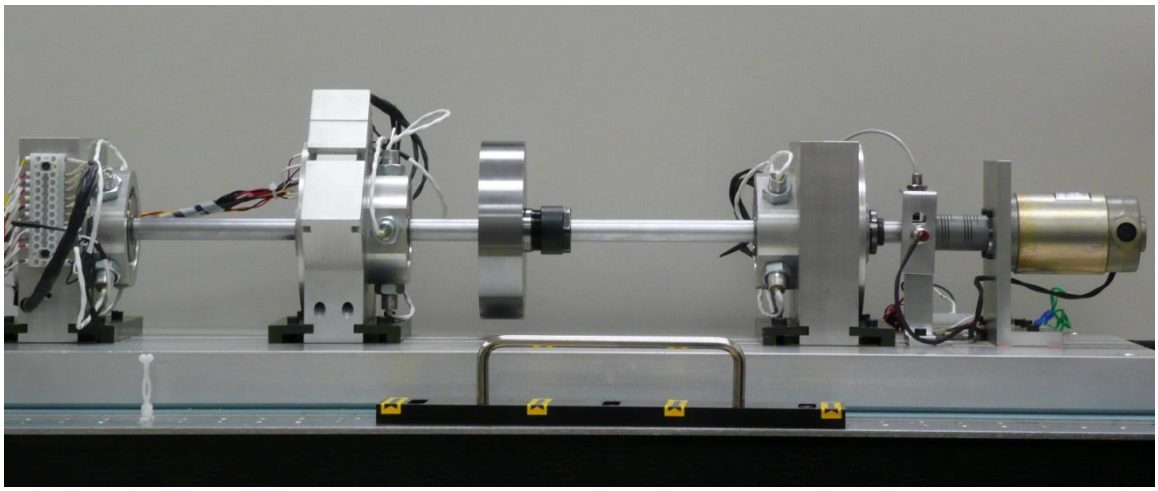


Figure IV.1 Crack Detection Test Rig in Experimental Configuration

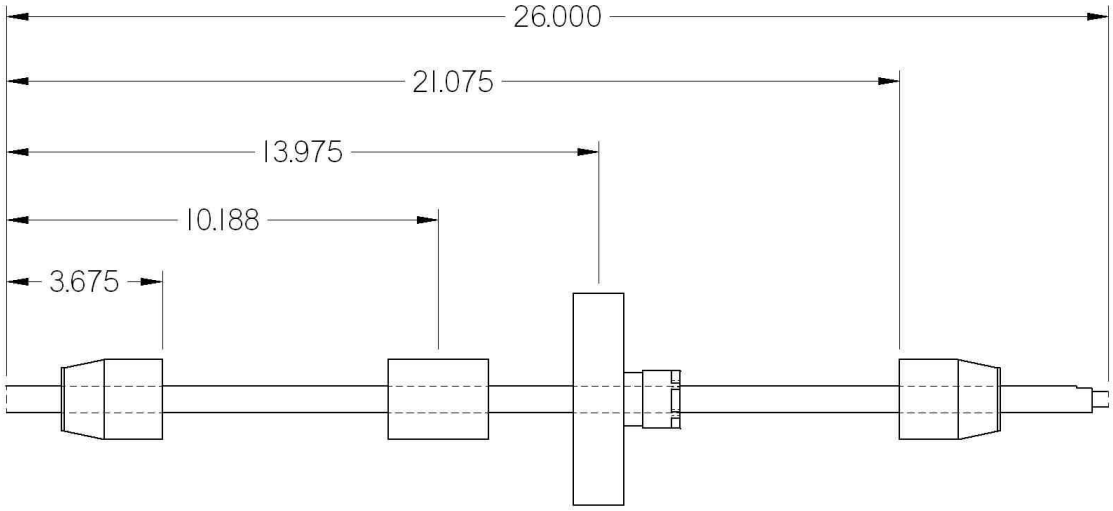


Figure IV.2 Crack Detection Rotor Configuration with Dimensions (inches)

To simulate a crack, a cut was put in the damaged rotor. This cut was made using a wire electrical discharge machine (EDM). The "crack" is approximately 115 μm wide, with a depth of 40% of the shaft diameter and is located at the bearing mid-span. A close-up of the crack is shown in Figure 4.3 (Pesch 2008).

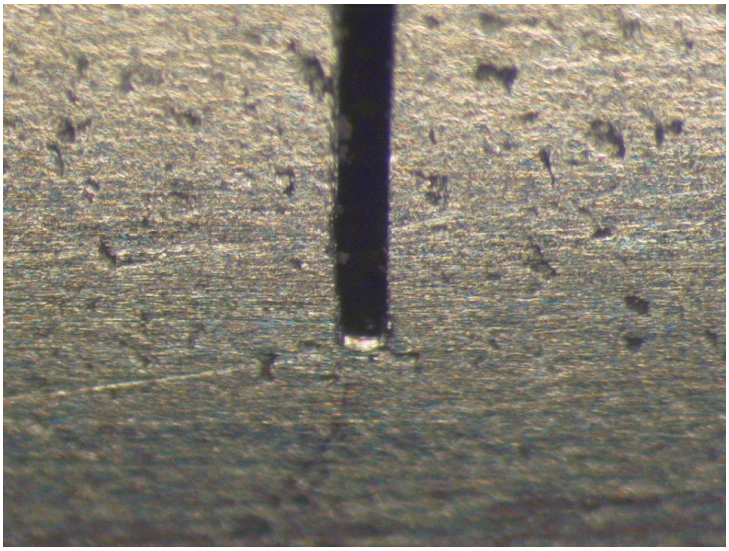


Figure IV.3 Bottom of Wire EDM Cut in Rotor

The trials from Pesch (2008) that are used in this model-based identification experiment do not use the two magnetic bearings near each end of the rotor. Instead, these trials have the rotor supported on deep-groove Conrad type ball bearings which sit in place of the magnetic bearings' touchdown bearings. Normally, the touchdown bearings are oversized and protect the magnetic bearing stators in the event of a levitation failure. In this case, the ball bearings are snug and used to support the rotor. The ball bearing trials were selected for analysis over the magnetic bearing trials because they contain sine sweeps conducted during rotation, where the magnetic bearing trials do not. This rotation during sine sweeps is important because it allows the crack to open and close, or breathes, giving a more accurate representation of the crack's behavior during normal operation. Additional information on the test rig may be found in Pesch's thesis (2008).

4.3 Modeling of the Rotor Systems

In order to utilize μ-controlled model-based identification to identify the changes in dynamics brought on by a transverse crack, state-space models of the healthy and cracked rotors need to be developed. The modeling is completed with a MATLAB program which generates the state-space representation of a rotor from a finite element model input file using Timoshenko beam elements. The parameters used to describe each element are: added weight, length, outer diameter, inner diameter, added polar moment of inertia, added transverse moment of inertia, Young's modulus, and mass density. Additionally, each node includes four binary values to identify if the location is a translational input, translational output, rotational input, and rotational output. The input files for the healthy and cracked rotors are shown in the Appendix.

For this experiment, additional elements needed to be added to the finite element shaft, including magnetic bearing rotors and a heavy disc. In modeling, lumped masses and shaft sections are used. The lumped mass uses added weight, polar moment of inertia, and transverse moment of inertia at the node of the standard element. In this experiment, beam elements will be used to model the magnetic bearing rotors and a lumped mass will be used to model the disc. Separate input files are generated for the healthy and cracked rotors.

The finite element input files are created to match experimental sine sweeps. These linear sine sweep trials have a frequency range of 20 to 1000 Hz, a step size of 0.5 Hz, a magnitude of 0.1 A, 40 revolutions per convolution, a settling time of 700 ms, and was done at 10 Hz rotation speed. The difference between the two sine sweep trials is the presence of a 40% transverse crack in the rotor, which differentiates the *healthy* and

cracked systems. Additionally, the results of these sweeps are truncated to a frequency range of 20 to 225 Hz because the output becomes dominated by noise at higher frequency values.

Both the *healthy* and *cracked* files begin from the same point- an input file created from the best known engineering parameters. The finite element program creates a state-space model of a free-free rotor using the input file. This generated state-space model has two differences from the sine sweep data which need to be reconciled. First, the finite element model is of a free-free rotor, where the sine sweep is of a rotor on ball bearings. Second, the sine sweep input is current while the free-free rotor input is force. These two discrepancies are corrected by placing the rotor on ball bearings as shown in Figure 4.4. In this figure, k_i and k_b represent the current stiffness of the exciter and the bearing stiffness, respectively. Interestingly, the crack input and crack output are included in the *healthy* system, but left out of the *cracked* system. This difference is a result of the controller being placed on the crack input and output of the *healthy* system for the model-based identification, a factor that will be explained further in the Formulation of Approach section.

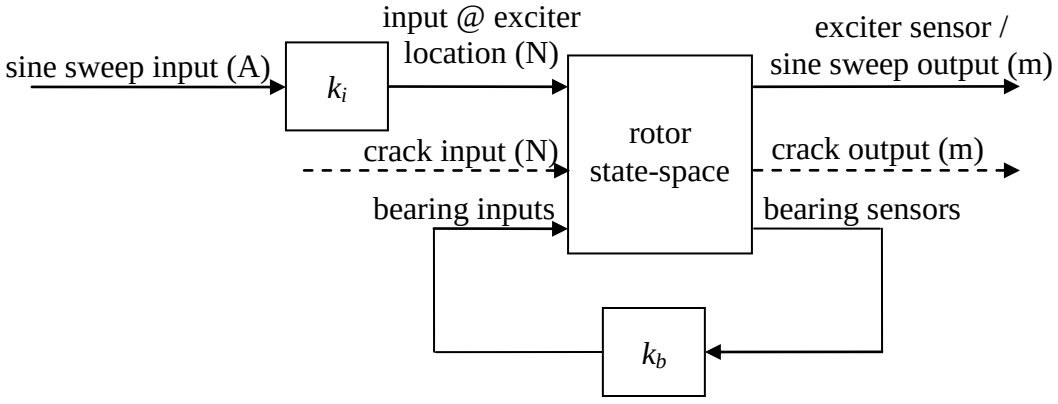


Figure IV.4 Rotor Model on Ball Bearings

Finally, the rotor models on ball bearings are tuned to match the experimental sine sweeps. The primary goal of this tuning was to match the first and second resonant peaks. Parameters that were tuned include: mass added by disc, exciter sensor location, density, and current stiffness. These were tuned manually through trial and error. Eventually, the peaks were matched very accurately. A deficiency in the modeling is shown in the anti-resonance. This was moved during tuning to be at the correct frequency but tuning was not able to bring the magnitude of the anti-resonance to the correct value. Once the tuning was complete, healthy and cracked systems of similar quality were generated. These final healthy and cracked systems are shown plotted against the experimental sine sweeps in Figures 4.6 and 4.7, respectively.

A graphical representation of the final input files is shown in Figure 4.5. This finite element model contains 30 nodes, with an overall length of 26 in. Points of interest include the ball bearing supports on stations 2 and 28, the exciter input on station 12, the exciter output on station 13, the disk on station 16, and the crack location at station 15. Note that the separation between the 12^{th} and 13^{th} node is only 0.05 in, so they appear as one heavy line on the plot. The only difference between the plot and the finite element input files is that the disk is modeled as a lumped mass in the input files.

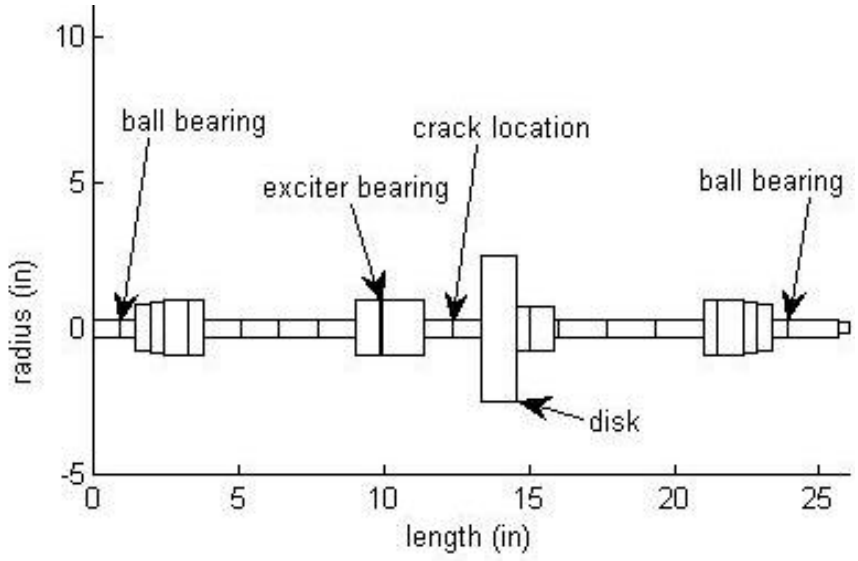


Figure IV.5 Model Configuration

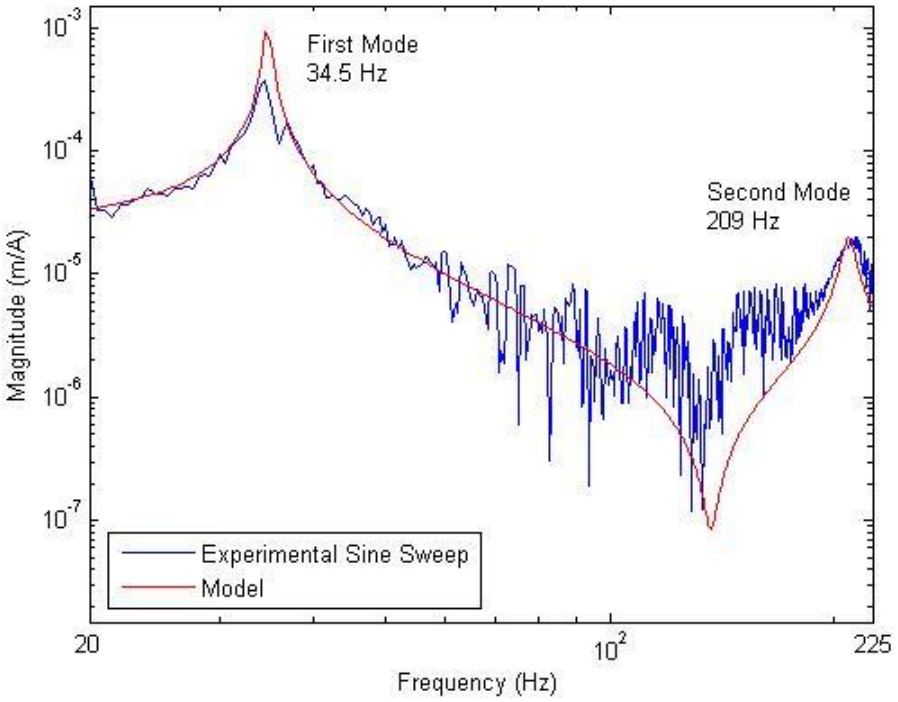


Figure IV.6 Healthy Rotor Model and Experimental Sine Sweep

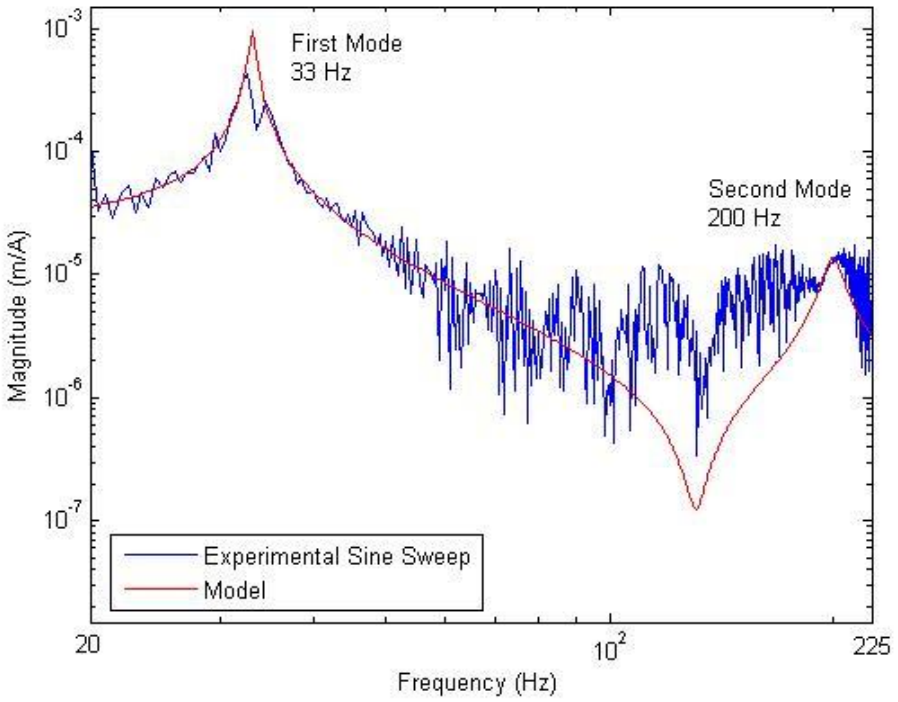


Figure IV.7 Cracked Rotor Model and Experimental Sine Sweep

4.4 Formulation of Approach

Before an approach is developed, it is important to identify the objective of the experiment. This objective is to identify the change in dynamics brought on by the presence of a transverse crack in a rotor system. Mu-controlled model-based identification is utilized to complete this goal. Accordingly, the algorithm laid out in Section 3.5 becomes the starting point for this experimental approach, which will be an adaptation of model-based identification for discovering the difference in dynamics between two systems.

First, measure the frequency response of the actual system. In this experiment, both systems are an "actual" physical model, which means that both the frequency responses of the cracked and healthy rotors are needed. This step was completed in the previous section.

Second, identify a state-space representation from the experimental frequency response which represents the true system, *truesys*. Next, create a nominal engineering model, *engsys*. These two steps show where using model-based identification for finding an unmodeled difference in dynamics differs from the standard procedure. Differentiating between the true and engineering systems is not as trivial as it normally would be because both systems are created from experimental frequency responses, so there is no "nominal" engineering model. In fact, the two systems can be described as "true" because of their derivation from experimental results. Consequently, there needs to be some other distinguishing factor. The two systems in this experiment are a healthy and cracked rotor. The objective is to identify the effects of adding a crack to the healthy rotor. This means that the crack will be modeled by the synthesized controller. The controller will replicate the dynamics induced by the presence of a crack, such that the extracted crack dynamics will force the healthy system response to match the cracked system response. Knowing that the controller works to drive the engineering system to the true system, it follows that the *healthy* and *cracked* systems will become the *engsys* and *truesys*, respectively. The modeling of the engineering and true systems was completed in the modeling section using the finite element program.

The fourth step is to identify difficult to model system components. For this experiment, the step is changed to: identify the dissimilar points between the two

models. Therefore the fifteenth node of the finite element model is chosen, or the node at the crack location. The chosen node becomes the location for the controller input, *y*, and output, *u*. This application of the controller to a particular point illustrates one of the major strengths in model-based identification. A controller applied to the cracked element allows for a local crack model which should be easier to interpret than one applied to the experimental system identification exciter input and output.

Next, assemble the μ-controlled model-based identification control schematic shown in Figure 3.3. Construction of the plant to be controlled begins with the identification of the parameters: *w*, *z*, *y*, and *u*. The excitation signal, *w*, and the closed-loop output, *z*, originate from the input and output of the sine sweep, leading to a current input at the exciter bearing coils and a position output at the exciter bearing sensor of each system. The controller input, *y*, and output, *u*, were already identified in the fourth step of the algorithm to be at the crack location of the healthy rotor. Additionally, the input weighting factor, W_w, needs to be assigned. The weighting factor is required to normalize the closed-loop input to a magnitude $|w| \leq 1.0$. Therefore, this equals the magnitude of the sine sweep trial inputs which were both 0.1 Amperes. The complete crack model identification schematic is shown in Figure 4.8. Figure 4.9 provides an alternative view of the control schematic. This is a more visual representation of the control scheme and blends the finite model of Figure 4.7 with the control schematic in Figure 4.8.

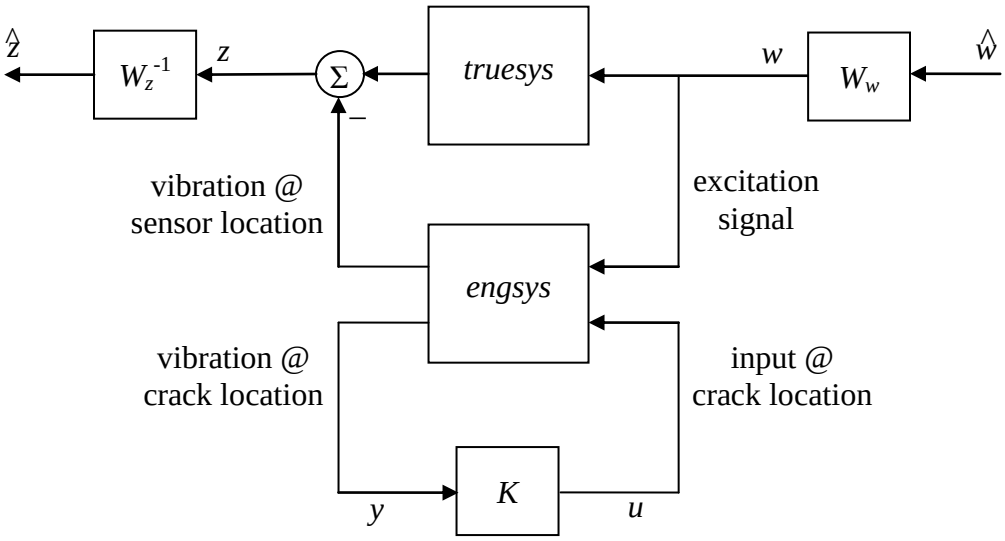


Figure IV.8 Crack Model Identification Schematic

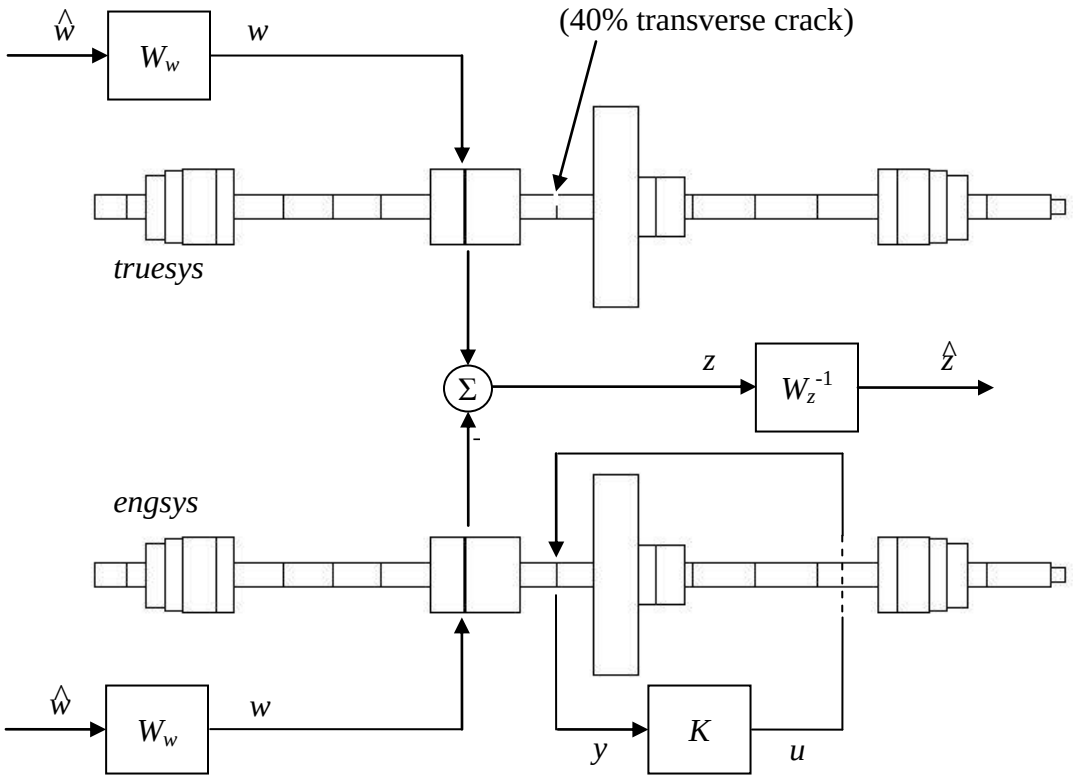


Figure IV.9 Graphical Representation of Crack Model-based Identification Schematic

The resulting controller plant is:

$$\mathbf{A}=\begin{bmatrix}\mathbf{A}_t & \mathbf{0}\\ \mathbf{0} & \mathbf{A}_e\end{bmatrix}\quad \mathbf{B}_1=\begin{bmatrix}\mathbf{B}_{ex,t}\\ \mathbf{B}_{ex,e}\end{bmatrix}W_w\quad \mathbf{B}_2=\begin{bmatrix}\mathbf{0}\\ \mathbf{B}_{cr}\end{bmatrix}\quad \mathbf{C}_1=W_z^{-1}\begin{bmatrix}\mathbf{C}_{ex,t} & -\mathbf{C}_{ex,e}\end{bmatrix}\quad \mathbf{C}_2=\begin{bmatrix}\mathbf{0} & \mathbf{C}_{cr}\end{bmatrix}$$

Where:

$\mathbf{A}_t$ is the dynamic matrix of the true system.

$\mathbf{A}_e$ is the dynamic matrix of the engineering system.

$\mathbf{B}_{ex,t}$ is the exciter input matrix of the true system.

$\mathbf{B}_{ex,e}$ is the exciter input matrix of the engineering system.

$\mathbf{B}_{cr}$ is the crack input matrix.

$\mathbf{C}_{ex,t}$ is the exciter output matrix of the true system.

$\mathbf{C}_{ex,e}$ is the exciter output matrix of the engineering system.

$\mathbf{C}_{cr}$ is the crack output matrix.

With all of the state matrices created by the finite element program and then placed on ball bearings as shown in Figure 4.4.

The final two steps of the μ-controlled model-based identification are to run the μ-synthesis and evaluate the μ-value to see if the controller synthesis step needs to be repeated. These steps are presented in the next section, which completes the experiment and presents the resulting crack model.

4.5 Application Utilizing Experimental Data

After the healthy and cracked rotor responses have been transformed into the *truesys* and *engsys*, and the control schematic has been developed, the μ-synthesis is run. The final iteration of the μ-synthesis returned $\mu \leq 0.9576$ when W_z=1.1×10^{-6}. Therefore, the controller is able to keep the unweighted closed-loop output, *z*, of the controlled engineering and true systems within 1.1 μm. Graphical outputs of the results are shown in Figures 4.10-4.13.

Figure 4.10 shows the frequency responses of the true system, engineering system, and controlled engineering system. The Bode magnitude plot shows the initial difference between the true and engineering systems, *truesys* and *engsys*, respectively. This initial difference illustrates the effect of adding a 40% transverse crack to the rotor model. The controller is then applied to drive engineering system to match the true system. As a result, the controlled engineering system or *controlled engsys* overlaps the response of the true system. Along with the output weighting, this confirms that the code has worked correctly and the controller gives an accurate model of the crack.

Next, Figure 4.11 shows the frequency response of the controller, which represents the crack model, or the dynamics brought on by introducing a 40% transverse crack to a rotor. Additional information is provided by the complete Bode plot shown in Figure 4.12 that includes the controller frequency response. For physical interpretation of the crack model, the response is compared to those from the three-mass trials. The crack response does not closely-resemble a change in stiffness as shown in Figure 3.17, a change in mass as in Figure 3.22, or a change in damping as in Figure 3.27. Clearly, the crack presents a more complicated change in dynamics than those three-mass trials.

Therefore, comparing the results to a trial with more changes in dynamics may lead to a match. Trial 4 included changes in mass, stiffness, and damping. Again, the crack model is more complicated than the controller from the three-mass trial. The primary difference is in the resonance peaks that occur 162 and 205 Hz. Additionally, these frequencies appear to have no physical meaning in the true or engineering systems.

An accepted interpretation of a crack is that it would bring a localized reduction in stiffness. As previously discussed, the magnitude response of the developed crack dynamics does not resemble the controller developed in Trial 1b. Interestingly, there is a similarity in the phase response of both of the models. Figure 3.18 shows that the phases of the nominal unmodeled dynamics, Du, and the developed controller, K, are 0° and $-360°$, respectively. The phase of the crack model begins at 720°, where it remains until it moves through the first resonant peak. Phase describes rotation around a circle so that the values of 0°, -360°, and 720° are at equivalent points about the circle.

Finally, Figure 4.13 provides further confirmation that the μ-synthesis worked correctly. The frequency response of $\hat{z}$ is plotted in order to confirm that the magnitude of the closed-loop output is kept under 1. Additionally, the plot is useful in illustrating the frequencies at which the controller has the most trouble meeting the performance criteria. The frequencies of the peaks in the $\hat{z}$ response closely relate to the resonant peaks of the engineering system shown in Figure 4.10. The implication associated with this result is that once the dynamics that drives the resonant peaks is corrected, the rest of the response will fall into matching the true system. Interestingly, the controller does not appear to struggle at the frequency of the anti-resonance in the engineering system.

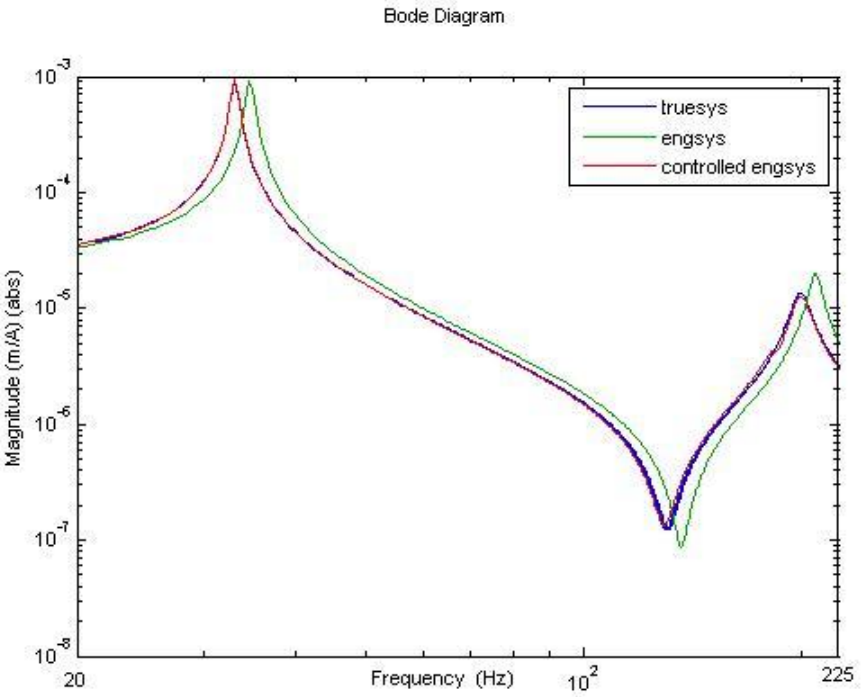


Figure IV.10 Crack Model Identification Frequency Responses

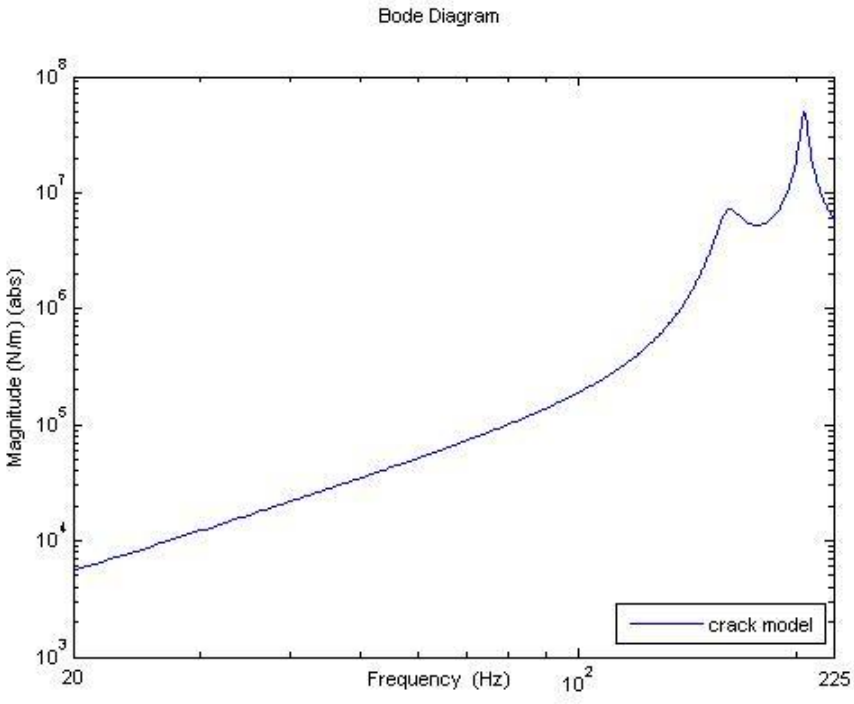


Figure IV.11 Identified Crack Model

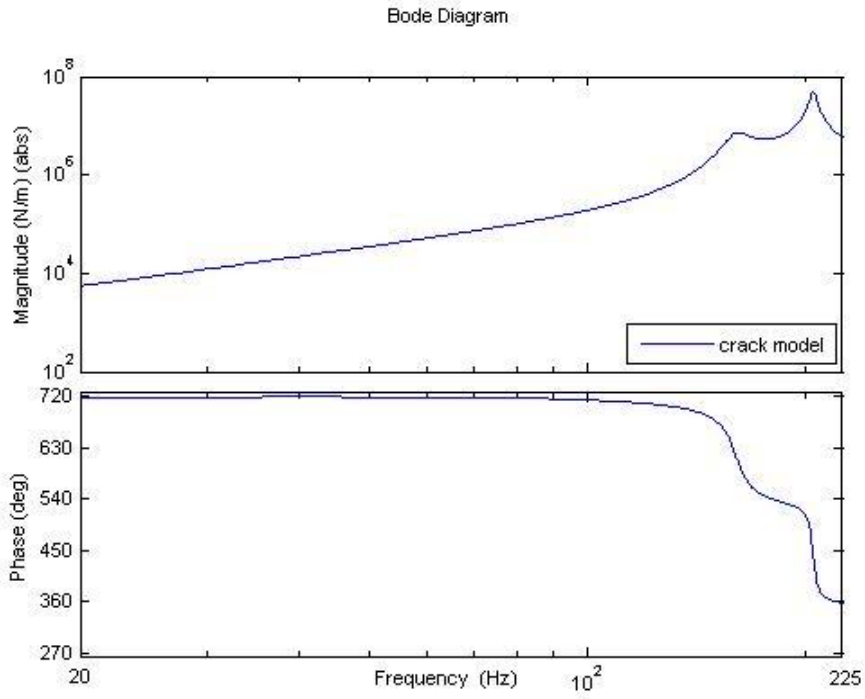


Figure IV.12 Identified Crack Model with Phase

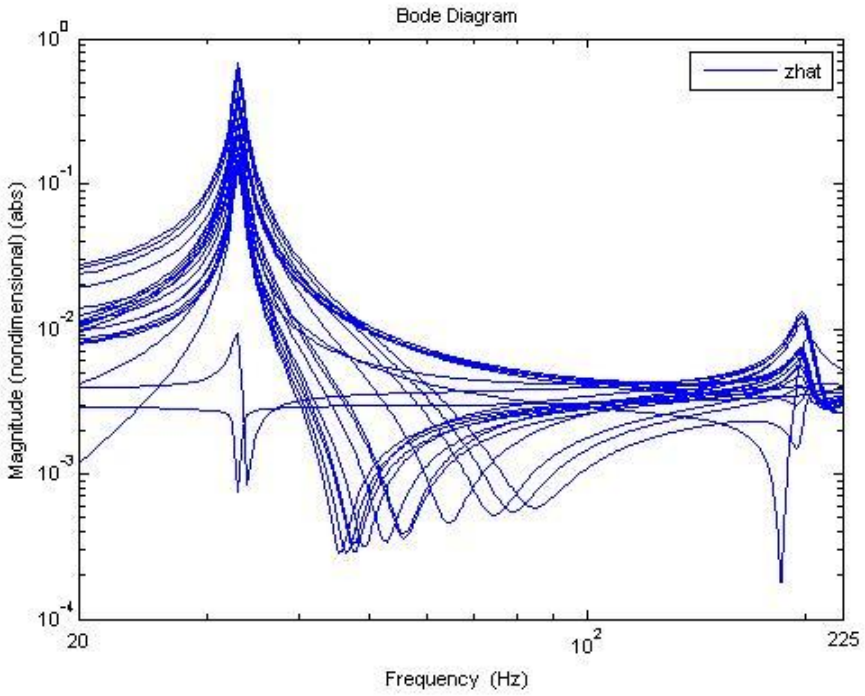


Figure IV.13 Crack Model Identification Closed-Loop Response

CHAPTER V

CONCLUSIONS

5.1 Summary

The primary objective of this thesis is to identify the changes in system dynamics induced by the presence of a transverse crack in a rotor; to this end the thesis was successful. Model-based identification was able to create an accurate model for the crack because it was derived from the experimental frequency responses of a healthy and a cracked rotor system. State-space system models were created from the frequency responses. A μ-synthesis control schematic is utilized to create a controller which minimizes the difference between the healthy and cracked system responses. Because the

difference is minimized by this controller, which is applied at the crack location, the controller is a new representation for the local crack dynamics.

Significant background work was required before the model for the change in dynamics brought on by the presence of transverse crack could be developed. First, the robust control tools used in model updating were introduced. Specifically, this included an introduction to the concepts of H_∞ control, μ-synthesis, and the linear fractional transformation. Most important was the background theory and explanation of how to use μ-synthesis, because this is the controller that is utilized by the model-based identification. Next, the combination of a μ-controller and model-based identification was developed for the application of missing dynamics. This is followed by a three-mass model study. The purpose of this study was to show how a difference in dynamics could be extracted with the application of μ-controlled model-based identification to a simple system, where the difference in dynamics is known, well understood in terms of modeling, and able to be measured directly. These trials led to several conclusions. First, the method is successful and intuitive. Next, the output weighting function W_z can be used as a performance measure of μ-controlled model-based identification. Finally, an update on the model-based identification algorithm was developed for extracting unmodeled dynamics using μ-synthesis and is shown in the Major Conclusions section.

5.2 Major Conclusions

The primary conclusion of this thesis is the new model of the change in dynamics induced by the presence of a transverse crack in a rotor. This model is shown in Figure

4.11. An equally important result of this paper is the development of an updated algorithm for model-based identification, which was first presented by Vazquez et al. (2003). This algorithm is modified for the identification of a known difference in dynamics using μ-synthesis, is a summary of the Formulation of Approach section of Chapter 4, and is presented below:

1. Measure the frequency response of the systems.
2. Identify a state-space representation (*truesys*) from the experimental frequency response of the system with the difference (the crack in this case).
3. Create an engineering model (*engsys*) from the experimental frequency response of the system without the difference.
4. Identify where the difference in dynamics occurs in the model.
5. Assemble the control schematic shown in Figure 3.3.
6. Run the μ-synthesis, resulting in controller, *K*.
7. Evaluate the μ-value:

    A. If $\mu << 1.0$, decrease W_z and repeat the μ-synthesis.

    B. If $\mu > 1.0$, increase W_z and repeat the μ-synthesis.

    C. If $\mu \leq 1.0$ within a user-defined acceptable error, then *K* represents the difference in the dynamics.

5.3 Future Work

The first addition to this study would be further examination of the identified crack model. This model could be used in simulation to see if it effectively predicts the

behavior of the cracked rotor at normal operating conditions, leading possibly to applications in damage detection. Also, further study to be done in the modeling of the dynamics brought on by the addition of a transverse crack in a rotor to see if a unified model could be developed. Such a study could include a more exhaustive look at the effects of crack depth, loading, and rotational speed on the dynamics brought on by the crack.

BIBLIOGRAPHY

Andry, A. N., Jr., E. Y. Shapiro, and J. C. Chung. "Eigenstructure Assignment for Linear Systems." *IEEE Transactions on Aerospace and Electronic Systems*, 1983: 711-729.

Balas, G. *Robust Control of Flexible Structures: Theory and Experiments.* PhD. Thesis, California Institute of Technology, 1990.

Balas, G., R. Chiang, A. Packard, and M. Safonov. *Robust Control Toolbox 3 User's Guide.* Natick: The MathWorks, 2009.

Baruch, M. "Methods of Reference Basis for Identification of Linear Dynamic Structures." *American Institute of Aeronautics and Astronautics Journal*, 1984: 561-564.

Baruch, M.., and Y. Bar Itzhack. "Optimal Weighted Orthogonalization of Measured Modes." *American Institute of Aeronautics and Astronautics Journal*, 1978: 346-351.

Berman, A. "Mass Matrix Correction Using an Incomplete Set of Measured Modes." *American Institute of Aeronautics and Astronautics Journal*, 1979: 1147-1148.

Boyd, S, V. Balakrishnan, and P Kabamba. "A bisection method for computing the Hinfinity norm of a transfer matrix and related problems." *Math. Control, Signals, and Systems*, 1989: 207-220.

Bucher, I., and S. Braun. "The Structural Modification Inverse Problem: An Exact Solution." *Mechanical Systems and Signal Processing*, 1993: 217-238.

Chen, W., Y. Xiong, K.-L. Tsui, and S. Wang. "Some Metrics and a Bayesian Procedure for Validating Predictive Models in Engineering Design." *Proceedings of IDETC/CIE 2006.* Philadelphia: ASME, 2006. 1-10.

Doyle, J. C. "Analysis of feedback systems with structured uncertainties." *IEEE Proceedings*, 1982: 45-56.

Doyle, J. C., A. Packard, and K. Zhou. "Review of LFTs, LMIs and mu." *Proc. IEEE Conf. Dec. Contr*. England, 1991. 1227-1232.

Doyle, J. C., K. Glover, P.P. Khargonekar, and B. A. Francis. "State-space solutions to standard H2 and Hinfinity control problems." *IEEE Trans. Automat. Contr.*, 1989: 831-847.

Doyle, J., B. Francis, and A. Tannenbaum. *Feedback Control Theory.* Macmillan Publishing Co., 1990.

Esteban, A. M. "Aircraft Applications of Fault Detection and Isolation Techniques." *University of Minnesota Dissertation.* 2004.

Farrar, C. R., and G. H. James III. "System Identification from Ambient Vibration Measurements on a Bridge." *Journal of Sound and Vibration,* 1997: 1-18.

Friswell, M. I. "The Adjustment of Structural Parameters Using a Minimum Variance Estimator." *Mechanical Systems and Signal Processing,* 1989: 143-155.

Friswell, M. I., and J. E. T. Penny. "The Effect of Close of Repeated Eigenvalues on the Updating of Model Parameters from FRF Data." *Transactions of the American Society of Mechanical Engineers,* 1992: 514-520.

Friswell, M. I., and J. E. T. Penny. "Updating Model Parameters from Frequency Domain Data via Reduced Order Models." *Mechanical Systems and Signal Processing,* 1990: 377-391.

Fritzen, C.-P., D. Jennewein, and T. Kiefer. "Damage Detection Based on Model Updating Methods." *Mechanical Systems and Signal Processing,* 1998: 163-186.

Gasch, R. "A Survey of the Dynamic Behaviour of a Simple Rotating Shaft with a Transverse Crack." *Journal of Sound and Vibration,* 1993: 313-332.

Gasch, R. "Dynamic Behaviour of a Simple Rotor with a Cross Sectional Crack." *International Conference on Vibrations in Rotating Machinery.* IMechE, 1976. 123-128.

Gladwell, G. M. L. *Inverse Problems in Vibrations.* Amsterdam: Martinus Nijhoff, 1986.

Glover, K., and J. C. Doyle. "A state space approach to Hinfinity optimal control." *Three Decades of Mathematical Systems Theory: A Collection of Surveys at the Occasion of the 50th Birthday of Jan C. Willems*, 1989.

Glover, K., and J. Doyle. "State-space formulae for all stabilizing controllers that satisfy an Hinfinity norm bound and relations to risk sensitivity." *Systems and Control Letters*, 1988: 167-172.

Hsu, K., T. Vincent, and K. Poolla. "Robust Structured Nonlinear System Identification." *Proceedings of the 45th IEEE Conference on Decision and Control.* San Diego: IEEE, 2006. 2518-2522.

Jaishi, B., and W.-X. Ren. "Damage Detection by Finite Element Model Updating Using Model Flexibility Residual." *Journal of Sound and Vibration*, 2006: 369-387.

Jun, O. S., H. J. Eun, Y. Y. Earmme, and C.-W. Lee. "Modelling and Vibration Analysis of a Simple Rotor with a Breathing Crack." *Journal of Sound and Vibration*, 1992: 273-290.

Maslen, E. H., and J. T. Sawicki. "Mu-synthesis for Magnetic Bearings: Why Use Such a Complicated Tool?" *Proceeding of IMECE 2007.* Seattle: ASME, 2007. 1-10.

Maslen, E. H., J. A. Vazquez, and C. K. Sortore. "Reconciliation of Rotordynamic Models with Experimental Data." *Journal of Engineering for Gas Turbines and Power*, 2002: 351-356.

Mayes, I. W., and W. G. R. Davies. "A Method of Calculating the Vibrational Behaviour of Coupled Rotating Shafts Containing a Transverse Crack." *International Conference on Vibrations in Rotating Machinery.* IMechE, 1980. 17-27.

Mayes, I. W., and W. G. R. Davies. "Analysis of the Response of a Multi-Rotor-Bearing System Containing a Transverse Crack in a Rotor." *Jounal of Vibration, Acoustics, Stress and Reliability in Design*, 1984: 139-145.

Minas, C., and D. J. Inman. "Matching Finite Element Models to Modal Data." *Transactions of the American Society of Mechanical Engineers*, 1990: 84-92.

Mottershead, J. E., and M. I. Friswell. "Model Updating in Structural Dynamics: A Survey." *Journal of Sound and Vibration*, 1993: 347-375.

Nagpal, K. M., and P. P. Khargonekar. "Filtering and smoothing in an Hinfinity setting." *IEEE Trans. Automat. Contr.*, 1991: 152-166.

Packard, A. "Gain scheduling via linear fractional transformations." *Systems and Control Letters*, 1994: 79-92.

Packard, A., and J. C. Doyle. "The comple structured singular value." *Automatica*, 1993: 71-109.

Parks, T. R. *Manual For Model 210/210a Rectilinear Control System.* Bell Canyon: Educational Control Products, 1999.

Penny, J. E. T., and M. I. Friswell. "Simplified Modelling of Rotor Cracks." *ISMA 27.* Leuven, 2002. 607-615.

Penny, J. E. T., M. I. Friswell, and C. Zhou. "Condition Monitoring of Rotating Machinery using Active Magnetic Bearings." *Proceedings of ISMA2006.* Leuven, 2006. 3497-3506.

Pesch, A. H. "Damage Detection of Rotors using Magnetic Force Actuator: Analysis and Experimental Verification." *Cleveland State University Thesis.* 2008.

Ran, A. C. M., and R. Vreugdenhil. "Existence and comparison theorems for algebraic Riccati equations for continuous- and discrete-time systems." *Linear Algebra and its Applications*, 1988: 63-83.

Ruotolo, R., and C. Surace. "On-Line Health Monitoring of Aeronautical Structures Using a Vibration Based Method." *21st ICAS Congress.* Melbourne: ICAS and AIAA, 1998.

Sampei, M., T. Mita, and M. Nakamichi. "An algebraic approach to Hinfinity output feedback control problems." *Systems and Control Letters*, 1990: 13-24.

Sawicki, J. T., and M. I. Friswell. "Detecting Cracked Rotors Using Auxiliary Harmonic Excitation." *Journal of Sound and Vibration*, 2010.

Sinha, J. K., and M. I. Friswell. "The Use of Model Updating for Reliable Finite Element Modelling and Fault Diagnosis of Structural Components used in Nuclear Plants." *Nuclear Engineeering and Design*, 2003: 11-23.

Smith, R. S., and A. Packard. "Robust Control Theory and Applications Course Notes." *Roy S. Smith, University of California, Santa Barbara.* May 20, 1997. http://www.ece.ucsb.edu/~roy/cgi-bin/makepage.pl?nav=course_232 (accessed December 1, 2010).

Srinathkumar, S. "Eigenvalue/Eigenvector Assignment Using Output Feedback." *IEEE Transactions on Automatic Control*, 1978: 79-81.

Storozhev, D. L. "Smart Rotating Machines for Structural Health Monitoring." *Cleveland State University Thesis.* 2009.

Vazquez, J. A., E. H. Maslen, H.-J. Ahn, and D.-C. Han. "Reconciliation of Rotordynamic Models with Experimental Data." *Journal of Engineering for Gas Turbines and Power*, 2003: 351-356.

Wang, Q. "Model-based Identification and Quality Estimation in Rotordynamics." *Dissertation, University of Virginia* (Dissertation), 2008.

Wang, Q., and E. H. Maslen. "Identification of Frequency-Dependent Parameters in a Flexible Rotor System." *Transactions of the ASME*, 2006: 670-676.

Wang, Q., B. Pettinato, and E. Maslen. "Identification in Rotordynamics: Model-Based vs. Direct Measurements." *Proceedings of ASME Turbo Expo 2009: Power for Land, Sea and Air*. Orlando: ASME, 2009a. 1-7.

Wang, Q., B. Pettinato, and E. Maslen. "Identification in Rotordynamics: Uncertainty Analysis and Quality Estimation." *Proceedings of ASME Turbo Expo 2009: Power for Land, Sea and Air*. Orlando: ASME, 2009b. 1-8.

Wroblewski, A. C. "Health Monitoring of Cracked Rotor Systems using External Excitation Techniques." *Cleveland State University Thesis*. 2008.

Xiong, Y., W. Chen, and K. L. Tsui. "A Better Understanding of Model Updating Strategies in Validating Engineering Models." *AIAA/ASME/ASCE/AHS/ASC Structures,*

Structural Dynamics, and Materials Conference. Schaumburg: American Institure of Aeronautics and Astronautics, 2008.

Zames, G. "Feedback and optimal sensitivity: model reference transformations, multiplicative seminorms, and approximate inverses." *IEEE Tran. Automat. Contr.*, 1981: 301-320.

Zhou, K., and J. C. Doyle. *Essentials of Robust Control.* Upper Saddle River: Prentice-Hall, 1998.

Zhou, K., J. C. Doyle, and K. Glover. *Robust and Optimal Control.* Englewood Cliffs: Prentice-Hall, 1995.

APPENDIX

FINTE ELEMENT INPUT FILES

Parameters from left to right: added weight, length, outer diameter, inner diameter, added polar moment of inertia, added transverse moment of inertia, Young's modulus, mass density, and four binary values to indicate if the location is a translational input, translational output, rotational input, and rotational output.

Healthy Model

0	0.875	0.63	0	0	0	30	0.3	0	0	0	0
0	0.5	0.63	0	0	0	30	0.3	1	1	0	0
0	0.53	1.59	0	0	0	30	0.3	0	0	0	0
0	0.47	1.78	0	0	0	30	0.3	0	0	0	0
0	0.875	1.88	0	0	0	30	0.3	0	0	0	0
0	0.5	1.88	0	0	0	30	0.3	0	1	0	0
0	1.313	0.63	0	0	0	30	0.3	0	0	0	0
0	1.313	0.63	0	0	0	30	0.3	0	0	0	0
0	1.313	0.63	0	0	0	30	0.3	0	0	0	0
0	1.313	0.63	0	0	0	30	0.3	0	0	0	0
0	0.875	1.88	0	0	0	30	0.3	0	0	0	0
0	0.05	1.88	0	0	0	30	0.3	1	0	0	0
0	1.45	1.88	0	0	0	30	0.3	0	1	0	0
0	1	0.63	0	0	0	30	0.3	0	0	0	0
0	1	0.63	0	0	0	30	0.3	1	1	0	0
4.81	1.2	0.63	0	21.1	12.01	30	0.3	0	0	0	0
0	0.45	0.63	0	0	0	30	0.3	0	0	0	0
0	0.8	0.63	0	0	0	30	0.3	0	0	0	0
0	0.206	0.63	0	0	0	30	0.3	0	0	0	0
0	1.656	0.63	0	0	0	30	0.3	0	0	0	0
0	1.656	0.63	0	0	0	30	0.3	0	0	0	0
0	1.656	0.63	0	0	0	30	0.3	0	0	0	0
0	0.5	1.88	0	0	0	30	0.3	0	0	0	0
0	0.875	1.88	0	0	0	30	0.3	0	1	0	0
0	0.47	1.78	0	0	0	30	0.3	0	0	0	0
0	0.53	1.59	0	0	0	30	0.3	0	0	0	0
0	0.5	0.63	0	0	0	30	0.3	0	0	0	0
0	1.75	0.63	0	0	0	30	0.3	0	0	0	0
0	0.375	0.38	0	0	0	30	0.3	1	1	0	0
0	0	0.38	0	0	0	30	0.3	0	0	0	0

Cracked Model

0	0.875	0.63	0	0	0	30	0.29	0	0	0	0
0	0.5	0.63	0	0	0	30	0.29	1	1	0	0
0	0.53	1.59	0	0	0	30	0.29	0	0	0	0
0	0.47	1.78	0	0	0	30	0.29	0	0	0	0
0	0.875	1.88	0	0	0	30	0.29	0	0	0	0
0	0.5	1.88	0	0	0	30	0.29	0	1	0	0
0	1.313	0.63	0	0	0	30	0.29	0	0	0	0
0	1.313	0.63	0	0	0	30	0.29	0	0	0	0
0	1.313	0.63	0	0	0	30	0.29	0	0	0	0
0	1.313	0.63	0	0	0	30	0.29	0	0	0	0
0	0.875	1.88	0	0	0	30	0.29	0	0	0	0
0	0.05	1.88	0	0	0	30	0.29	1	0	0	0
0	1.45	1.88	0	0	0	30	0.29	0	1	0	0
0	1	0.63	0	0	0	30	0.29	0	0	0	0
0	1	0.63	0	0	0	30	0.29	1	1	0	0
5.21	1.2	0.63	0	21.1	12.01	30	0.29	0	0	0	0
0	0.45	0.63	0	0	0	30	0.29	0	0	0	0
0	0.8	0.63	0	0	0	30	0.29	0	0	0	0
0	0.206	0.63	0	0	0	30	0.29	0	0	0	0
0	1.656	0.63	0	0	0	30	0.29	0	0	0	0
0	1.656	0.63	0	0	0	30	0.29	0	0	0	0
0	1.656	0.63	0	0	0	30	0.29	0	0	0	0
0	0.5	1.88	0	0	0	30	0.29	0	0	0	0
0	0.875	1.88	0	0	0	30	0.29	0	1	0	0
0	0.47	1.78	0	0	0	30	0.29	0	0	0	0
0	0.53	1.59	0	0	0	30	0.29	0	0	0	0
0	0.5	0.63	0	0	0	30	0.29	0	0	0	0
0	1.75	0.63	0	0	0	30	0.29	0	0	0	0
0	0.375	0.38	0	0	0	30	0.29	1	1	0	0
0	0	0.38	0	0	0	30	0.29	0	0	0	0