IDENTIFICATION OF PARAMETRIC UNCERTAINTY FOR THE CONTROL OF FLEXIBLE STRUCTURES

by

Joel S. Douglas

B.S., Carnegie Mellon University (1989)
S.M., Massachusetts Institute of Technology (1991)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1995

Author .....
Department of Electrical Engineering and Computer Science
May 26, 1995

Certified by .....
Michael Athans
Professor of Electrical Engineering
Thesis Supervisor

Accepted by .....
Frederic R. Morgenthaler
Chairman, Departmental Committee on Graduate Students

IDENTIFICATION OF PARAMETRIC UNCERTAINTY FOR THE CONTROL OF FLEXIBLE STRUCTURES

by

Joel S. Douglas

Submitted to the Department of Electrical Engineering and Computer Science
on May 26, 1995, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Modern robust control techniques require a description of the uncertainty in the plant to be controlled. For lightly damped structures, the most appropriate description of the uncertainty is in terms of interval ranges for parameters such as natural frequencies and damping ratios. What is desired is an algorithm which can determine such interval ranges from noisy transfer function measurements using set membership identification techniques. We begin with a parameterization of the structural model which is numerically stable. However, because the parameterization is nonlinear, this will result in a set of nonlinear optimization problems. Our approach is to embed these problems into a set of convex optimization problems. The added conservatism of the embedding can be made arbitrarily small for a one mode system by partitioning the parameter space into a finite number of regions. For a multiple mode system, an overbound on the level of conservatism can be easily measured.

We then investigate the situation when the compensator designed for our uncertain system does not achieve the desired robust performance goal. The philosophy pursued is to determine a new input to apply to the open loop system in order to reduce the uncertainty. A new approach based upon sensitivity analysis is presented. Using the standard upper bound to the structured singular value as our measure of performance, we calculate the sensitivity of the performance to the size of the parametric uncertainty, and estimate the effect of the inputs on this uncertainty. This information is combined to determine the input with the largest expected improvement in the performance. Several examples demonstrate the ability of this procedure to achieve the desired performance using only a small number of data points.

Thesis Supervisor: Michael Athans
Title: Professor of Electrical Engineering

Acknowledgments

I would like to thank Professor Michael Athans, for his help and guidance over my years at MIT. He has given me great freedom in finding a thesis topic, and also has shown great patience in letting me explore my ideas. I have learned a lot about control from him, especially in using physical insight to help understand the problem. I also appreciate his efforts to make sure I always had funding.

I would also like to thank the members of my thesis committee. I wish to thank Professor Munther Dahleh, who, through the course of my graduate study, has taught me a great deal about control theory and identification. I wish to thank Professor Gunter Stein, who met with me many times in the course of this research, and his insights are greatly appreciated. I wish to thank Dr. David Miller, who has taught me a great deal about structures throughout my association with SERC.

I have benefited greatly through my interaction with many friends and colleagues at MIT. I would like to thank Peter Young for patiently answering my questions, even when I asked them more than once. I would like to thank Wesley McDermott for spending much time discussing ideas on the board. I would like to thank Steven Patek for many useful discussions on the theoretical aspects of my work. I would like to thank John Wissinger, for his enthusiasm and encouragement.

Many others have provided assistance, to which I am grateful. I would like to thank Frank Aguirre, Alan Chao, Marcos Escobar, Simon Grocott, Pat Kreidl, Mitch Livstone, and Leonard Lublin. I would especially like to thank Amy McCain for her invaluable friendship, and for cheering me up whenever I was discouraged. I would also like to thank Ed Moriarty for making MIT a more friendly place.

I would like to thank my wife, and best friend, Margaret. She is the most patient and understanding person I have ever known. Her unconditional love and support have been a tremendous assistance in all aspects of my life.

This research was carried out at the MIT Space Engineering Research Center with support from NASA Grant NAGW-1335 from NASA, Washington, D.C., with Mr. Gordon Johnston serving as technical monitor.

Contents

Chapter 1

Introduction

1.1 Overview

In this thesis, we examine the robust control problem from an identification point of view. We have an unknown flexible structure, and we wish to apply robust control techniques to guarantee a certain level of performance. The problem is to appropriately model the structure so that our control methodology can achieve the desired performance.

The problem examined in this work is aimed specifically at flexible structures. Structures are assumed to be open loop stable. By flexible, we mean that all of its poles are lightly damped, i.e. near the $j\omega$ axis in the complex plane. The design of a compensator which produces a desired level of performance for such a structure is very difficult when we do not have an accurate model of the system. A high performance control design can easily drive the system unstable when the exact pole or zero locations are not known.

Given a set of a priori information, we must generate an appropriate model from input–output data. Since there is noise in the system, any model we calculate will have some inherent model uncertainty. This uncertainty needs to be quantified in order to determine if the control design will meet the stability and performance goals.

There are two types of uncertainty for such a system, unstructured uncertainty and parametric uncertainty. Unstructured uncertainty is always present, due to un-

modelled dynamics. It could be argued that all the uncertainty in the system could be modelled this way. However, this would lead to unnecessary conservatism in the amount of uncertainty. It is more appropriate to model the uncertainty as having a contribution due to the unmodelled dynamics, and another part which is due to uncertainty in the parameters of the system. In this thesis, we will concentrate on parameter uncertainty.

Since we are concerned with parameter uncertainty, the parameterization of the model is very important. It immediately determines which parameters we would need to identify. We would like a parameterization which is appropriate for control, yet gives us physical insight. We would also like to work with a model which is numerically stable. With this model and a set of input–output data, we need to identify both the nominal parameters and the corresponding uncertainty to design robust controllers.

The discussion so far has assumed that we have a set of input–output data on which we will model the system. The next question to examine is how we should generate this data. We would like to be able to determine the "optimal" inputs to the system to identify our model parameters. We define optimal as the inputs we should apply to the system so that after the identification and control design are complete, we have the best possible robust performance.

In order to make this a tractable problem, we need to consider the input design in an iterative framework. We begin with a set of input–output data, identify an appropriate model of the system, and design a compensator. We then wish to determine which input to apply to best improve the current system. We then collect additional input–output data, and remodel the system. The process can then be repeated. The question of interest is how to choose these inputs so that the robust performance of the system is improving as much as possible at each step.

Both the question of identifying the parametric uncertainty in our system, and how to choose inputs to improve the performance are examined in this work. Throughout, we will stress the ability to apply these techniques to complicated systems. Suboptimal techniques are developed to ensure reasonable computational requirements.

1.2 Previous Work

A significant amount of work has been done in the field of identification. A good example of the work done in "classical" identification is Ljung [36]. This book describes in some detail the importance of having a model which is *identifiable*, i.e. a model for which two different sets of parameters yield two different systems. It also covers least squares identification of autoregressive type models. However, it does not cover identification for control, nor the appropriate models for control.

There have been several identification methods specifically designed for structures. One of the most popular is the Eigensystem Realization Algorithm (ERA) [29],[30]. The basic idea is to create the Hankel Matrix based upon impulse response data. The impulse response data is typically determined from measured frequency response data. From the Hankel matrix, a realization can be found. Methods based upon the Hankel matrix can be traced back to Ho and Kalman [27], and to Silverman [51]. A more general theory was presented by Akaike in [1]. In these methods, the Hankel matrix is factored to produce a minimal state space description.

Similar methods to ERA are the Q-Markov technique [35], and the Observability Range Subspace Extraction [34]. In Q-Markov, the first Q Markov parameters of a system driven by white noise are matched exactly. From these Markov parameters, a stochastic version of the Hankel matrix can be determined. ORSE is a generalization of both ERA and Q-Markov. ORSE can use arbitrary inputs, as long as the structure is sufficiently excited.

One of the inherent problems with these approaches is that they typically require us to compute the pseudoinverse of a very large matrix. For lightly damped systems, this matrix can become ill-conditioned. One method to avoid these problems is given in [31]. Here, instead of identifying the actual Markov parameters of the system, we identify the Markov parameters of an observer of the system, with poles at any location we choose. We then determine what the actual Markov parameters are based upon the observer Markov parameters, as well as determining the observer gain used. A state space realization is then obtained using ERA.

A recent method which has produced very good results is the log–least squares algorithm [28], [50]. In this algorithm, the cost function is defined as the two–norm of the difference between the logarithms of the data and the model. This cost function is appropriate for systems with a large dynamic range, like a lightly damped system, because it penalizes errors at low magnitudes (near the zeros) and errors at high magnitudes (near the poles) similarly. The drawback to this method is that it requires a nonlinear optimization.

So far, the methods described do not attempt to measure the uncertainty in the system. One method which explicitly finds a bound on the error is the so-called control-oriented system identification [25]. The goal is to determine an algorithm such that the worst case identification error converges to zero as the number of frequency domain data points increases and the noise decreases. In addition, a bound on the worst case error is sought. Several papers have examined this issue in depth, among them [2], [9], [10], [24], and [26]. Unfortunately, accuracy with this method is achieved through increasingly higher order models. Also, the only description of uncertainty is through the $\mathcal{H}_\infty$ bound of the error.

An approach which bounds the uncertainty in the parameters of the system is called set membership identification. One of the earlier papers on this subject was [18]. Here, the system is modelled as linear in the unknown parameters, with noise which is unknown but bounded in amplitude. The goal is to find the set of all parameters which is consistent with the bound on the noise. Since finding an exact solution quickly becomes intractable, an algorithm is proposed which finds an ellipsoid that bounds this "consistent" parameter region as tightly as possible. Several other papers have been written on the subject, and other similar algorithms have been proposed, e.g. [5]. See also surveys [12], [42], and [54]. The major drawback to this approach is that it is limited to systems with parameters entering linearly into the system (although there are some extensions; see for example [45]).

The model used for control design is very important. Most modern control techniques require a state space model of the system. Many state space descriptions for a given input-output model exist. In [55], the sensitivity of the poles of a system due to

changes in parameters is studied. It is shown that although the controllable canonical and observable canonical forms have low complexity (in terms of the number of multiplications and divisions needed for calculations with these representations), they are very sensitive to changes in parameters. They are therefore described as "unsuitable" models for controller design. One model which is more suitable for controller design is a block diagonal structure. In [21], an algorithm is given to convert a model described in terms of poles and residues into a block diagonal model. This algorithm works for multi-input multi-output systems, and produces models which are numerically stable.

The final topic to be considered is choosing optimal inputs for identification. A survey of the early work in this subject is found in [39]. The objectives are either accurate determination of parameter values, or prediction. The criteria are usually related to the Fisher information matrix. These approaches are therefore not necessarily applicable to improving the robust performance of a system. Similar work using a closed loop system is given in [37]. Here, the sensitivity of the outputs to the parameters is minimized, which turns out to be related to the Fisher information matrix. An interesting result shown in [39] is that an input consisting of a finite number of frequencies can be found which has the same information as any other stationary input with equal power.

Some of the more modern approaches have considered the joint identification–control problem. In [48], it is argued that for good performance, identification and control must be considered as a joint problem, and that an iterative scheme is needed. This type of approach is followed in [4], where the objective is to design a controller with the best robust performance over all plants consistent with the data. The uncertainty in this case is modelled as a nonparametric bound on the additive uncertainty. In [33], an estimator is considered for adaptive control systems. The estimator provides a nominal model and a measure of uncertainty. This uncertainty is in the form of a magnitude bound on the modeling errors in the frequency domain.

To summarize, there has been much research into the field of identification. In terms of identification algorithms for the purpose of designing robust controllers for structural systems, there have been three main areas of research. There are algo-

rithms for the identification of structures, which do not provide a measure of the uncertainty. There is a class of algorithms which provides uncertainty in terms of an $\mathcal{H}_\infty$ error instead of the uncertainty in parameters. Finally, there are algorithms which determine the uncertainty in parameters, but do not provide a parameterization which is appropriate for controlling structural systems. Clearly there is a need for a method which combines aspects of all three of these types of approaches, and this thesis was motivated by these considerations.

1.3 Contributions and Outline of Thesis

There are two main goals of this thesis. First, given a set of frequency domain data from a lightly damped structure, we would like to determine a model and a description of the uncertainty appropriate for robust control design. Furthermore, the model should be numerically stable. It is argued that the most appropriate description of the uncertainty is in terms of uncertainty in the parameters. Previous work has either used models which are not appropriate for lightly damped structures, or has not been able to determine the parametric uncertainty in the model.

The model we will use for the identification will be based upon a modal decomposition of the flexible structure. This is an appropriate model for this type of system, as shown in chapter 2. However, because it is nonlinear in the parameters, identification of the parametric uncertainty becomes very difficult. An iterative algorithm is introduced which can solve this problem. The algorithm uses a set of frequency domain input–output data. This data is corrupted both by noise bounded in magnitude, as well as by additional uncertainty from unmodelled dynamics. The algorithm will determine bounds on the parameters of the system by examining one mode of the structure at a time, and iteratively reducing the uncertainty.

The second goal of this thesis is to determine a method to choose new inputs to improve the robust performance of our closed loop system. Previous work has examined choosing inputs to reduce parametric uncertainty, but not in a manner which is the most appropriate for the performance of the system. Here, a new method

for input design is introduced. The design is based upon a closed loop sensitivity analysis. We assume that we have a specified control design methodology. We wish to choose an input to apply to the system from a finite set of inputs. To do so, we estimate the sensitivity of the closed loop performance measure to each of the inputs. We then apply the input with the highest sensitivity, for this is the one which we expect will improve the performance the most. These sensitivities must be estimated both due to the noise, and also due to the nonlinear relationship between the inputs and the performance. The analysis is done strictly using open loop data, as the philosophy is to guarantee robust stability and performance before closing the loop.

A computationally efficient method to determine these sensitivities will be determined. It is based upon using the chain rule to write the sensitivity in terms of the sensitivity of the performance metric to the size of the uncertainties, and the sensitivity of the uncertainties to the inputs. Several new techniques are introduced for this analysis, including the μ–sensitivities, which determine the sensitivity of the structured singular value μ to any parameter in the system.

An iterative scheme is then proposed for identification, control design, and input design. It is seen through several examples that this algorithm can quickly improve the robust performance of our system. It does this by choosing inputs to reduce the parametric uncertainty in the system in a manner which is most appropriate for the control design. Some convergence issues are examined, as are the limitations of this approach.

It is important to note that the inputs resulting from this algorithm are not necessarily "optimal" solutions. The nonlinear relationship between the performance, the control design, and the identification make truly optimal solutions impossible. It is necessary to decompose the problem into a number of separate steps in order to avoid the complicated interrelationships present. Approximations are then required to create a scheme with a reasonable amount of computation. The iterative methodology presented is a heuristic approach, and few convergence results can be guaranteed.

This thesis is organized as follows. In chapter 2, several important issues are examined. We describe the parameterization of flexible structures used in this work.

We also define the robust performance metric μ, and show how it is computed. We then motivate the need for an iterative algorithm for input design, and provide some notation which will be useful in the derivations.

In chapter 3, we derive the identification algorithm. The algorithm proceeds by reducing our system to a set of one mode problems. To determine the parametric uncertainty, a set of convex optimization problems are solved. The conservatism of the approach is analyzed, and a method to reduce the conservatism is outlined.

In chapter 4, we introduce the methodology to determine the next input to apply to our system through sensitivity analysis. In this chapter, the computational aspects of the method are stressed. The convergence issues of this approach in an iterative scheme are examined in chapter 5.

Several examples of the algorithms introduced are shown in chapter 6. The first examples show the ability of the identification algorithm to determine accurate parameter intervals from a very small amount of data. We then examine in detail the ability of the input design algorithm to intelligently choose inputs which improve the guaranteed performance of the system.

Finally in chapter 7, we will summarize the results presented, and discuss some avenues for future work.

Chapter 2

Preliminaries

In this chapter, several key elements of the problems to be addressed are introduced, and some notation is defined. We begin by looking at system parameterization, and how it relates to flexible structures. A numerically stable representation is described. It is this model which will be used in chapter 3 for the identification algorithm. We restrict the discussion to single–input single–output systems. Some issues with multiple–input multiple–output systems are addressed in appendix A.

In section 2.2, we will define the measure of robust performance to be used in this work, and show how it can be calculated. We then motivate and discuss iterative algorithms in the framework introduced. Some notation is introduced which will ease the presentation in chapters 4 and 5.

2.1 Parameterization of Flexible Structures

A lightly damped system is most easily described in terms of its modes. Each mode is a pair of complex conjugate, lightly damped poles, and can be represented in the form

$$G_i(s) = \frac{r_i(s)}{s^2 + 2\zeta_i\omega_i s + \omega_i^2} \tag{2.1}$$

where $r_i(s)$ is the residue of the i^{th} mode, ω_i is its natural frequency, and ζ_i is its

damping ratio. We can write $r_i(s)$ as a first order term:

$$r_i(s) = b_{1i}s + b_{0i} \tag{2.2}$$

Since the system can be written as a sum of these terms, we can write the system $G(s)$ as

$$G(s) = \sum_{i=1}^{n} G_i(s) \tag{2.3}$$

In [21], a method for creating a state space description from this particular parameterization is given. The resulting system is a block diagonal system, given by

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.4}$$

$$y(t) = Cx(t) \tag{2.5}$$

$$A = blockdiag(A_1, \ldots, A_n); \qquad B = \begin{bmatrix} B_1 \\ \vdots \\ B_n \end{bmatrix}; \qquad C = \begin{bmatrix} C_1 & \ldots & C_n \end{bmatrix} \tag{2.6}$$

Here, B_i and C_i are determined by the algorithm in [21], and A_i is given by

$$A_i = \begin{bmatrix} 0 & 1 \\ -\omega_i^2 & -2\zeta_i\omega_i \end{bmatrix} \tag{2.7}$$

For single-input single-output systems, without loss in generality, we can choose

$$B_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \qquad C_i = \begin{bmatrix} b_{0i} & b_{1i} \end{bmatrix} \tag{2.8}$$

This state space description is easy to compute once we have a frequency description of the model. Furthermore, it allows us to use the physical parameters ω_i and ζ_i directly in the model. The model is therefore parameterized in terms of these physical parameters, and also the residues of the system. The block diagonal structure makes this description numerically stable, i.e. not sensitive to changes in the parameters.

For single-input single-output systems, this description immediately lets us incorporate uncertainty in the parameters directly into the system. Assuming we have identified uncertainty in all the terms in $G_i(s)$, we can write

$$A_{i,true} = A_i + \begin{bmatrix} 0 & 0 \\ \delta_{1i} & \delta_{2i} \end{bmatrix} \tag{2.9}$$

$$C_{i,true} = C_i + \begin{bmatrix} \delta_{3i} & \delta_{4i} \end{bmatrix} \tag{2.10}$$

Here, δ_{1i} is the uncertainty in ω_i^2, δ_{2i} is the uncertainty in $2\zeta_i\omega_i$, δ_{3i} is the uncertainty in b_{0i}, and δ_{4i} is the uncertainty in b_{1i}. Notice that we are not directly determining uncertainty in the structural parameters such as stiffness and damping parameters, but rather in the coefficients of characteristic polynomials for each term. As it turns out, this is a convenient description which we use at the expense of some physical insight.

2.2 The Mixed μ Problem

The underlying goal of this work is to be able to design a compensator for a flexible structure. The compensator needs to guarantee the stability of the structure, and also that we have met certain performance goals. Furthermore, we need to make these guarantees even in the face of uncertainty. In this section, we will describe an appropriate measure of robust performance. We will use the standard upper bound to mixed μ, defined in [57].

We consider the system shown in figure 2-1. In this system M represents the stable closed loop nominal system including the nominal plant and the compensator, and Δ represents the uncertainty in the system. This uncertainty includes parametric uncertainty, unstructured uncertainty (unmodelled dynamics), and fictitious uncertainty used to transform the robust performance problem into a robust stability problem (see, for example, [16]). The system is normalized so that $\|\Delta\|_\infty \leq 1$.

We will limit our discussion here to where the uncertainty consists of nonrepeated

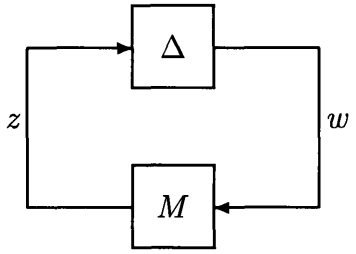

Figure 2-1: Standard M-Δ system

parametric uncertainty, and scalar unstructured uncertainty. This is done for notational simplicity, and is sufficient for the problem considered in this research. As seen in section 2.1, the parametric uncertainty enters in a nonrepeated fashion. Also, since we restrict ourselves to single–input single–output systems, the unstructured uncertainty is scalar. For a more general description, see [57] and the references therein.

The Δ block therefore consists of these scalar uncertainties. We will order our uncertainties so that all of the parametric (real) uncertainties are first. The uncertainty block will therefore have the structure of the following set.

$$\boldsymbol{\Delta} = \{\Delta = diag(\delta_1, ..., \delta_p, \delta_{p+1}, ...\delta_q) : \delta_i \in \mathcal{R}, \quad i = 1, ..., p; \quad \delta_i \in \mathcal{C} \quad i = p+1, ..., q\} \tag{2.11}$$

Any robust performance question can be recast into the above "M-Δ" form. The structured singular value now indicates whether or not we have met robust stability and performance. It is defined at a particular frequency as follows.

Definition 2.1

$$\mu(M(\omega)) = \left(\inf_{\Delta \in \boldsymbol{\Delta}} \{\overline{\sigma}(\Delta) : \det(I - M(\omega)\Delta) = 0\}\right)^{-1} \tag{2.12}$$

with $\mu(M(\omega)) = 0$ *if there is no* $\Delta \in \boldsymbol{\Delta}$ *such that* $\det(I - M(\omega)\Delta) = 0$.

μ is calculated in this manner for each frequency point. It is a necessary and sufficient condition for the robust stability and performance of our loop, in the sense that $\mu \leq 1$ for all frequencies if and only if we have met the robust stability and

performance specifications. However, it is very difficult to compute μ exactly. Instead, we compute an upper bound $\overline{\mu}$. This upper bound provides a sufficient condition for robust stability and performance in the sense that $\overline{\mu} \leq 1$ for all frequencies guarantees robust stability and performance.

We can find an upper bound to μ following [58]. To find an upper bound, we need to define two different types of scaling matrices, the "D-scales", and the "G-scales". For our problem, they are defined by

$$\mathcal{D} = \{diag(d_1, ..., d_q) : 0 < d_i \in \mathcal{R}\} \tag{2.13}$$

$$\mathcal{G} = \{diag(g_1, ..., g_p, 0, ..., 0) : g_i \in \mathcal{R}\} \tag{2.14}$$

Now, an upper bound $\overline{\mu}$ over all frequencies is given by the following definition.

Definition 2.2

$$\overline{\mu}(M) = \sup_{\omega} \inf_{D\in\mathcal{D},G\in\mathcal{G}} \left[\min_{\beta\geq 0} \left\{\beta : \overline{\lambda}(M_D^*(\omega)M_D(\omega) + j(GM_D(\omega) - M_D^*(\omega)G)) \leq \beta^2\right\}\right] \tag{2.15}$$

where $M_D(\omega) \triangleq DM(\omega)D^{-1}$, *and* $\overline{\lambda}(X)$ *denotes the maximum eigenvalue of* X.

This upper bound will be our measure of robust performance. The goal is therefore to determine a model of our system, and design a compensator so that $\overline{\mu}$ is less than 1. $\overline{\mu}$ can be computed as discussed in [58], and is available as a Matlab toolbox [3].

An alternate definition to $\overline{\mu}$ can be found in [58], and is presented in the following lemma. The proof can be found in [58].

Lemma 2.1 *An alternate definition to* $\overline{\mu}$ *is the following.*

$$\overline{\mu}(M) = \sup_{w} \inf_{D\in\mathcal{D},G\in\mathcal{G}} \left[\min_{\beta\geq 0} \left\{\beta : M^*(w)DM(w) + j(GM(w) - M^*(w)G) \leq \beta^2 D\right\}\right] \tag{2.16}$$

Before leaving the discussion on $\overline{\mu}$, we will present a result which will be useful in this work. This result shows us that if the size of one of our uncertainties decreases

while everything else remains fixed (i.e. the nominal model and the compensator), then $\overline{\mu}$ will be less than or equal to the value of $\overline{\mu}$ before the uncertainty decreased. We can represent the system with the uncertainty in the j^{th} parameter reduced by a factor of $(1-\epsilon)^2$ as in figure 2-2.

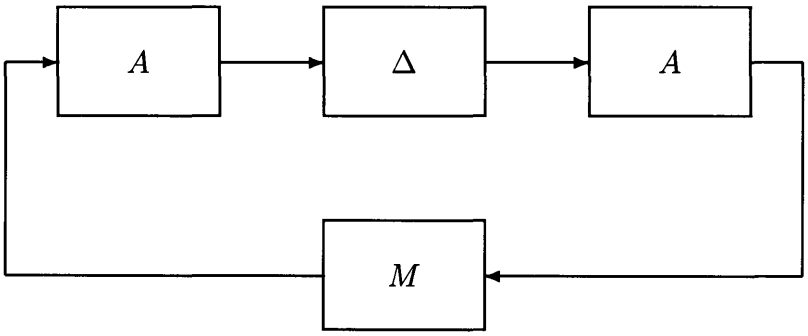

Figure 2-2: M-Δ system with a decrease in the uncertainty represented by A.

In the figure, $A = diag(1, 1, ..., 1, 1-\epsilon, 1, ..., 1)$; that is a matrix with ones on the diagonal except for a $1-\epsilon$ in the (j, j) position. We will assume $\epsilon < 1$. This represents reducing the uncertainty in the j^{th} parameter by a factor $(1-\epsilon)^2$. That is, $|\delta_j| \leq (1-\epsilon)^2$, which indicates the uncertainty in this parameter has decreased from the normalized value $|\delta_j| \leq 1$. In order to recalculate $\overline{\mu}$, we need to renormalize the uncertainty. To do so, we could include the perturbation as part of M, and compute $\overline{\mu}(AMA)$. It should be noted that we consider the decrease in a balanced framework (i.e. the matrix A both premultiplies and postmultiplies M) because it tends to be more numerically stable.

With this notation, we now have the following result.

Lemma 2.2 *Pointwise in frequency,* $\overline{\mu}(AMA) \leq \overline{\mu}(M)$.

Proof: We begin by using the alternate expression for $\overline{\mu}$ in lemma 2.1. If we let M represent the closed loop system at a particular frequency, then $\overline{\mu}(M)$ at this frequency can be calculated as follows.

$$\overline{\mu}(M) = \inf_{D\in\mathcal{D},G\in\mathcal{G}} \left[\min_{\beta\geq 0} \left\{\beta : M^*DM + j(GM - M^*G) \leq \beta^2 D\right\}\right] \tag{2.17}$$

Assume therefore that there is a $D \in \mathcal{D}$ and $G \in \mathcal{G}$ such that

$$M^*DM + j(GM - M^*G) \leq \beta^2 D \tag{2.18}$$

Let us define the following scaled matrices.

$$\tilde{D} \triangleq A^{-1}DA^{-1} \qquad \tilde{G} = AGA^{-1} \tag{2.19}$$

From the definition of A, it is clear that $\tilde{D} \in \mathcal{D}$ and $\tilde{G} \in \mathcal{G}$. Furthermore, by the structure of G and A, $AGA^{-1} = A^{-1}GA$. Substituting these quantities, we have

$$M^*A\tilde{D}AM + j(A^{-1}\tilde{G}AM - M^*A\tilde{G}A^{-1}) \leq \beta^2 ADA \tag{2.20}$$

We will not change the sign definiteness of the expression by premultiply and post-multiply the expression by A, since A is diagonal and positive definite. We therefore get the following result.

$$\begin{aligned} AM^*A\tilde{D}AMA + j(\tilde{G}AMA - AM^*A\tilde{G}) &\leq \beta^2 A^2 DA^2 && (2.21)\\ &\leq \beta^2 D && (2.22)\\ & && (2.23) \end{aligned}$$

Thus $\overline{\mu}(M) \leq \beta \Rightarrow \overline{\mu}(AMA) \leq \beta$. ■

2.3 Philosophy Behind the Iterative Scheme

It is clear that for our measure of robust performance to be meaningful we need both a model of the system as well as a description of the uncertainty. A method to determine parametric uncertainty for flexible structures will be derived in chapter 3. Based upon this model, we can design a compensator, and then determine if we have met our robust performance goals using $\overline{\mu}$. The question we would like to consider is: what should we do if we have not met our goals?

There are several steps in the design process where we could try to improve on our current robustness guarantees. For instance, we could try to redesign the compensator to improve our performance, by either changing weighting functions, or using a new design methodology. In this work, we will consider this redesign to be part of the design procedure. The control design may therefore be iterative in nature, and may involve some tuning. The resulting compensator is considered to be the best design using the control methodology under consideration.

We must therefore improve the model in order to improve our robustness guarantees. In this work, the approach will be to choose a new input to apply to the system in order to collect more data. With this new data, we use the identification algorithm to determine a better model of the system, and redesign the compensator. We would like to choose an input which will help improve our performance as much as possible.

In this type of framework, it is necessary to use an iterative scheme. The identified model of the system is clearly a function of the inputs we have already applied to the system. However, we wish to choose the inputs based upon the performance measure. Since we can not determine the performance without a model of the system and a compensator, the inputs are necessarily a function of our model. We are therefore unable to choose the optimal inputs a priori. A good input for one model is not necessarily a good input for another model. The input we choose must be a function of our current model.

In general we must design the inputs based upon the closed loop system, but our understanding of the closed loop system is based upon the inputs we have chosen. We therefore proceed by using an iterative scheme. Details on the steps of such a scheme will be introduced in the next section, when we define some notation which will be useful to us later.

2.4 Notation

Let us now define some general notation, and outline the steps in the iterative scheme. We will assume that the system has p unknown parameters, which we will

write as a vector $\theta \in \mathcal{R}^p$. In the identification scheme, we will identify upper and lower bounds for the parameters. These will be denoted as $\overline{\theta}$, and $\underline{\theta}$ respectively. Thus, we will have the following relationship.

$$\underline{\theta} \leq \theta \leq \overline{\theta} \tag{2.24}$$

The midpoint of the uncertainty intervals will be denoted $\hat{\theta}$, which is determined in a straightforward manner as follows.

$$\hat{\theta} = \frac{1}{2}(\overline{\theta} + \underline{\theta}) \tag{2.25}$$

At times it will be useful to refer to all of the bounds in one vector. We will define the vector $\phi \in \mathcal{R}^{2p}$ by writing

$$\phi = \begin{bmatrix} \underline{\theta} \\ \overline{\theta} \end{bmatrix} \tag{2.26}$$

The parametric uncertainty in the system will be denoted by δ. It is calculated as follows.

$$\delta = \overline{\theta} - \underline{\theta} \tag{2.27}$$

When referring to a particular element of a vector, we will use a subscript. For instance, the j^{th} element of δ will be written as δ_j. Also, since we are using an iterative algorithm, we will need an iteration index. This will be done using a superscript. For example, at the k^{th} iteration, the current bounds are written as ϕ^k.

We will assume that our model structure is fixed, and that we have a set bound on the effects of the unmodelled dynamics in the system. Modulo this, we are left with the bounds ϕ as completely describing the current model. We will refer to ϕ as being the current model of the system, including the nominal model and parametric uncertainty.

Since the model structure is assumed to be fixed, we can consider the goal of the identification algorithm to be determining upper and lower bounds on the parameters θ. We will assume that for the k^{th} iteration we have collected N_k data points from the

open loop system. Since we are using frequency domain data, this data is complex. The identification algorithm is viewed as a map $h : \mathcal{R}^{2p} \times \mathcal{C}^{N_k} \rightarrow \mathcal{R}^{2p}$

$$\phi^{k+1} = h(\phi^k, y^k) \tag{2.28}$$

where y^k is the data at the k^{th} iteration.

After determining the model of the system, we will design a compensator. We will represent the control design as a map from our parameter space to the compensator.

$$C^k = f(\phi^k) \tag{2.29}$$

In general, there may be some weighting functions that could be used in the control design to tune the compensator to the particular model under consideration. In this work, we will consider the weights or any other tuning of the compensator as part of the map f.

Our performance will be based upon our model of the closed loop system. The nominal system is chosen to be the closed loop system with the parameters $\hat{\theta}$, with the uncertainty δ. We will represent this model in one of two ways. We will typically write it as a function of the bounds ϕ and the compensator C. Thus, at the k^{th} iteration, we will represent the closed loop system as $M(\phi^k, C^k)$. Here, the fact that the nominal system is at the midpoint of the bounds is implicit. However, sometimes we will want to emphasize the difference between the nominal model and the uncertainty. In this case, we will write the closed loop system as $M(\hat{\theta}^k, \delta^k, C^k)$.

Finally, we note that we will drop all arguments when it will cause no confusion.

The steps in the iterative algorithm are therefore the following. We begin with a set of transfer function data from our flexible structure. Using the algorithm to be introduced in chapter 3, we determine a nominal model, and the corresponding parametric uncertainty.

A compensator is then designed based upon this model. In this research, the control design methodology is not specified. Any methodology could be used, and it is expected that this may include tuning of frequency weights or using engineering

judgment to determine an appropriate compensator.

Finally, we can evaluate $\overline{\mu}$, and see if we have met our robust performance goals. If not, we choose an input to the open loop system to help improve our robustness guarantees as much as possible. Note that this is not necessarily the input which reduces the uncertainty as much as possible (measured by some norm). In this respect, the philosophy of this approach is different from previous work.

Finally, with the new data collected, we can update our model and redesign the compensator.

2.5 Summary

To summarize, we will outline the steps needed for designing a compensator with the philosophy presented here. This outline will serve as a roadmap for the algorithms presented.

1. Determine the inputs for the initial experiment, and measure the data y^0. Let $k = 1$, and let ϕ^0 denote our a priori knowledge of the parameters.

2. Determine the model $\phi^k = h(\phi^{k-1}, y^{k-1})$. Essentially, we need to determine the upper and lower bounds to the parameters of a flexible structure. This is done in chapter 3.

3. Design the compensator $C^k = f(\phi^k)$. We do not specify the control methodology, as the choice is made based upon performance goals, computational limitations, and other engineering considerations.

4. If $\overline{\mu}(M(\phi^k, C^k)) < 1$, then stop.

5. Determine the next input to apply to the open loop system, using a sensitivity analysis. This is done in chapters 4 and 5.

6. Set $k = k + 1$, and go to step 2.

Chapter 3

Set Membership Identification of Lightly Damped Systems

In this chapter, we examine the problem of determining uncertainty intervals for the parameters in the model described in section 2.1. Set membership identification techniques are adapted to the case where the parameters are not linear in the data. An iterative algorithm is derived which guarantees the set of all plants consistent with the data and our a priori knowledge is contained in our uncertainty description. The conservatism of the algorithm is examined, and ways to reduce the conservatism are discussed.

3.1 Set Membership Identification

We begin with an introduction to set membership identification. Various methods to solve the standard set membership identification problem are discussed. These methods will be extended in later sections for our problem.

3.1.1 The Standard Problem

The basic assumptions in set membership identification are as follows. We have a single-input single-output system described by

$$y_k = \theta^T X_k + w_k; \qquad k = 1, 2, \ldots \tag{3.1}$$

where $\theta^T = [\ \theta_1,\ \theta_2,\ \ldots,\ \theta_p\ ]$ is the vector of unknown parameters. Typically, X_k consists of past inputs and outputs, i.e. $X_k^T = [\ y_{k-1},\ \ldots\ y_{k-m},\ u_k,\ \ldots,\ u_{k-l}\ ]$ with $p = m + l - 1$. However, any X_k and y_k which satisfies equation (3.1) is valid. $\{w_k\}$ is a noise sequence which is bounded by

$$|w_k| \leq r_k; \qquad r_k > 0; \qquad k = 1, 2, \ldots \tag{3.2}$$

Notice that with our assumption on the noise, equation (3.1) is equivalent to

$$|y_k - \theta^T X_k| \leq r_k \qquad k = 1, 2, \ldots \tag{3.3}$$

Following the discussion in [18], the goal is to find the set of parameters Θ^* which is consistent with these equations. When we have k data points, this set Θ_k^* is the one given by

$$\Theta_k^* = \bigcap_{i=1}^{k} \left\{\theta : (y_i - \theta^T X_i)^2 \leq r_i^2; \theta \in \mathcal{R}^p\right\} \tag{3.4}$$

Finding this set Θ_k^* is very difficult due to the large amount of information at each step, and the complexity of the resulting set. Thus, we would like to find an overbounding set Θ_k such that

$$\Theta_k^* \subset \Theta_k \tag{3.5}$$

We would like this set Θ_k to be simple computationally, while overbounding as "tightly" as possible.

To visualize the problem, note that equation (3.3) summarizes all of the information available from a given data point. In parameter space, this is the area between two hyperplanes. Thus, the set of parameters consistent with all of the data points

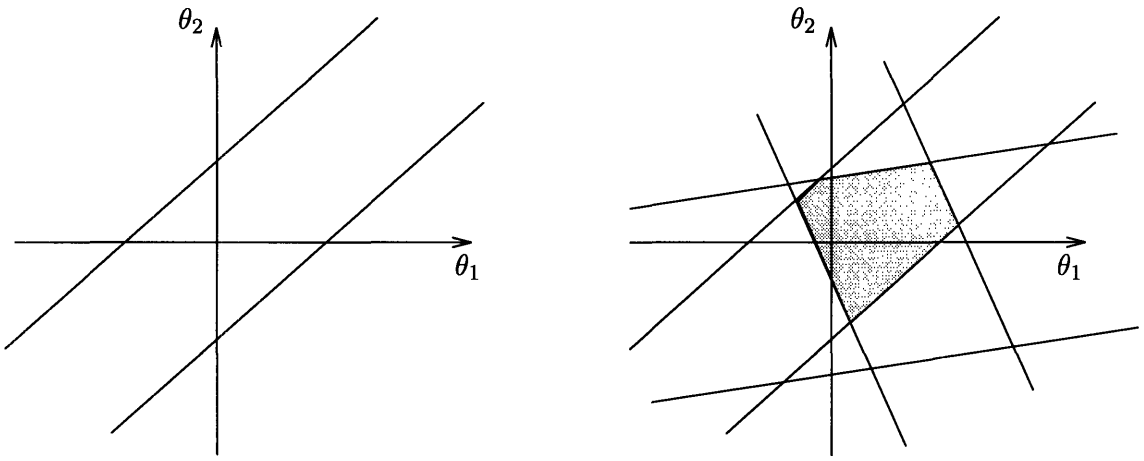

Figure 3-1: Information from a single data point, and the exact set of consistent parameters from three data points (shaded area).

is the set of parameters in the intersection of all of these hyperplanes. See figure 3-1.

There are many ways to attack this problem. Over the years, researchers have developed algorithms to outerbound the set of feasible parameters. One of the first such methods, described in [18], finds an ellipsoid which contains the set of feasible parameters. As we obtain more data, the ellipsoid is recursively updated. The advantage of this type of approach is that as we get more data, it is computationally very simple to update our set. The disadvantage is that it can be conservative, because not only can the ellipsoid bound be poor, but also because we only use one data point for each update. See [5], [11], [41], [44], [46], and [54] for modifications to the basic algorithm of [18] which attempt to reduce this conservatism.

Another common method is to find a polytope which bounds the feasible set as tightly as possible. This is the philosophy used in this thesis and is outlined in the next section. The advantage of this approach is that the resulting set is typically much less conservative. The disadvantage is that it is nonrecursive in nature, and thus it is not as easy to update the feasible set as we collect new data.

3.1.2 Solution Through Linear Programming

A common approach to solving the set membership problem (3.3) is to bound the feasible set Θ_k^* with a polytope. A case which will be of interest to us is when we

wish to determine independent interval ranges for each of the uncertain parameters. Thus the question asked is what are the maximum and minimum values for each of the uncertain parameters in our set Θ_k^* of feasible parameters. This is equivalent to finding the smallest box aligned with the parameter axes which contains the set Θ_k^*.

This problem can be cast into a linear programming problem as introduced in [40]. We can solve the following set of problems.

Problem 3.1

1. *Determine upper bounds:*

$$\max \theta_i$$

$$s.t. \quad |y_k - \theta^T X_k| \leq r_k \quad k = 1, 2, 3, ...$$

2. *Determine lower bounds:*

$$\min \theta_i$$

$$s.t. \quad |y_k - \theta^T X_k| \leq r_k \quad k = 1, 2, 3, ...$$

This is clearly a set of linear programs which can be solved using standard software for the case of finite data.

3.2 Application to Lightly Damped Structures

We now introduce the set membership problem for lightly damped structures. We will use the parameterization of section 2.1, and assume we have measured frequency domain data. The noise free system is given by

$$g_0(s) = \sum_{i=1}^{n} \frac{b_{1i}s + b_{0i}}{s^2 + a_{1i}s + a_{0i}} \tag{3.6}$$

We will assume that we have measured data at N frequency points ω_i, and that this data is corrupted by noise bounded in magnitude.

$$g(j\omega) = g_0(j\omega) + n(j\omega) \qquad |n(j\omega)| \leq R(j\omega) \tag{3.7}$$

We will assume that we know the bound $R(j\omega)$. The set membership problem in which we are interested is to find all parameters consistent with the set of equations

$$\left| g(j\omega_i) - \sum_{k=1}^{n} \frac{b_{1k} j\omega_i + b_{0k}}{-\omega_i^2 + a_{1k} j\omega_i + a_{0k}} \right| \leq R(j\omega_i) \quad i = 1, ..., N \tag{3.8}$$

This is clearly a very difficult problem. The set of parameters consistent with this equation is, in general, nonconvex. To solve this problem, we will first reduce the problem into a set of single mode problems, and then examine how to solve each of the one mode problems.

3.2.1 Reduction to a One Mode Problem

Here we describe how to reduce our problem to a set of problems with one mode. To simplify notation, let us define

$$g_i(s) = \frac{b_{1i} s + b_{0i}}{s^2 + a_{1i} s + a_{0i}} \tag{3.9}$$

So we can rewrite equation (3.8) for $\omega = \omega_i$ as

$$\left| g(j\omega) - \sum_{i=1}^{n} g_i(j\omega) \right| \leq R(j\omega) \tag{3.10}$$

Let $\hat{g}_i$ be an estimate of the i^{th} mode, determined either by curve fitting or choosing the midpoints from a priori intervals for each parameter. We will estimate the parameters of the i^{th} mode by subtracting out our estimates of the other modes. The philosophy is to remove all the dynamics from the system other than those of the i^{th} mode. Of course, we need to take into account any possible error due to model mismatch. Using the triangle inequality, we see

$$\left| (g(j\omega) - \sum_{k \neq i} \hat{g}_k(j\omega)) - g_i(j\omega) \right| \leq R(j\omega) + \sum_{k \neq i} |\hat{g}_k(j\omega) - g_k(j\omega)|_{max} \tag{3.11}$$

where $|\hat{g}_k(j\omega) - g_k(j\omega)|_{max}$ is a bound on the maximum this term can achieve due to the parametric uncertainty (to be described shortly). Equation (3.11) is now in

the form of a problem with one uncertain mode, with noise bound given by the right hand side of the inequality. We have therefore reduced the multiple mode case to a set of single mode problems. We can solve these problems using the methods to be outlined in the following sections. Once we have done this however, we may then be able to reduce the bound on $|\hat{g}_k(j\omega) - g_k(j\omega)|_{max}$. So we reidentify the bounds of the parameters in g_i, and continue to iterate in this fashion until we no longer can reduce the bounds. Thus, we get the following iterative algorithm.

Algorithm 3.1 (Multiple Mode System)

1. *Determine an upper bound to $|\hat{g}_k(j\omega) - g_k(j\omega)|$ near frequencies of all modes.*
2. *Solve each of the one mode set membership problems described by equation (3.11) to get bounds on the parameters of g_i for each mode i.*
3. *Recalculate the values of $|\hat{g}_k(j\omega) - g_k(j\omega)|_{max}$.*
4. *Have the bounds on the parameters improved (more than some tolerance)? If yes, go to step 2. If no, stop.*

The result is an iterative algorithm to estimate the parameters. Several comments need to be made. First, it should be noted that we are estimating the parameters of each mode separately. While we do this, we are treating the errors in the other modes as noise. To do this, we really need a new piece of a priori information: the maximum error contribution of each mode to the overall frequency response, which we clearly do not have a priori. Our solution to this problem is to use an upper bound to this contribution. The algorithm is therefore most effective when this contribution can be made small.

Let us describe how we can get such bounds. Since $\hat{g}_i(j\omega)$ is a known complex number, and $g_i(j\omega)$ contains uncertain parameters, we can use the following lemma to derive a bound which is less conservative than the triangle inequality.

Lemma 3.1 *Assume* $a \in \mathcal{R}$, $a > 0$, $b = Be^{j\psi}$, $0 < \underline{B} \leq B \leq \overline{B}$, $\underline{\psi} \leq \psi \leq \overline{\psi}$, $-2\pi \leq \underline{\psi}, \overline{\psi} \leq 2\pi$. *If* $-\pi, \pi \notin \left[\underline{\psi}, \overline{\psi}\right]$, *then,*

$$|a-b| \leq \max\left\{|a-\underline{B}e^{j\underline{\psi}}|, |a-\underline{B}e^{j\overline{\psi}}|, |a-\overline{B}e^{j\underline{\psi}}|, |a-\overline{B}e^{j\overline{\psi}}|\right\} \tag{3.12}$$

Proof: Let us define

$$f(B,\psi) = |a-Be^{j\psi}| = \left(a^2+B^2-2aB\cos\psi\right)^{\frac{1}{2}} \tag{3.13}$$

Since the set of all possible values of B and ψ is compact, we have for some values B^* and ψ^*

$$f(B^*,\psi^*) = \max_{B\in\left[\underline{B},\overline{B}\right],\psi\in\left[\underline{\psi},\overline{\psi}\right]} f(B,\psi) \tag{3.14}$$

First, if $\pm\pi \in \left[\underline{\psi}, \overline{\psi}\right]$, then we have for all $\psi \in \left[\underline{\psi}, \overline{\psi}\right]$,

$$\cos(\pm\pi) = -1 \leq cos(\psi) \tag{3.15}$$

$$-2aB^*\cos(\pm\pi) \geq -2aB^*\cos(\psi) \tag{3.16}$$

Thus we see that

$$f(B^*,\pm\pi) \geq f(B^*,\psi) \tag{3.17}$$

Clearly, if $\pm\pi$ is in our allowable range, it is optimal (this is the triangle inequality).

Now, assume that $-\pi < \underline{\psi} \leq \psi \leq \overline{\psi} < \pi$. If $\psi^* \neq \underline{\psi}$ and $\psi^* \neq \overline{\psi}$, then there is some $\epsilon > 0$ such that

$$\psi^* \pm \epsilon \in \left[\underline{\psi}, \overline{\psi}\right] \tag{3.18}$$

This implies that

$$\cos(\psi^*+\epsilon) < \cos(\psi^*) \quad \text{or} \quad \cos(\psi^*-\epsilon) < \cos(\psi^*) \tag{3.19}$$

Thus we have

$$f(B^*, \psi^* + \epsilon) > f(B^*, \psi^*) \text{ or } f(B^*, \psi^* - \epsilon) > f(B^*, \psi^*) \tag{3.20}$$

In either case, it shows that ψ^* is not optimal. Since this is a contradiction, it must be that $\psi^* = \underline{\psi}$ or $\psi^* = \overline{\psi}$.

Now assume that $B^* \neq \underline{B}$ and $B^* \neq \overline{B}$. Thus we have that for some $\epsilon > 0$,

$$\underline{B} < B^* \pm \epsilon < \overline{B} \tag{3.21}$$

We have

$$\left(a^2 + (B^* + \epsilon)^2 - 2a\,(B^* + \epsilon)\cos(\psi^*)\right) - \left(a^2 + (B^*)^2 - 2aB^*\cos(\psi^*)\right)$$

$$= \epsilon^2 + (2B^*\epsilon - 2a\epsilon cos\psi^*) \tag{3.22}$$

$$\left(a^2 + (B^* - \epsilon)^2 - 2a\,(B^* - \epsilon)\cos(\psi^*)\right) - \left(a^2 + (B^*)^2 - 2aB^*\cos(\psi^*)\right)$$

$$= \epsilon^2 - (2B^*\epsilon - 2a\epsilon cos\psi^*) \tag{3.23}$$

Thus we have

$$f^2(B^* + \epsilon, \psi^*) - f^2(B^*, \psi^*) > 0 \text{ or } f^2(B^* - \epsilon, \psi^*) - f^2(B^*, \psi^*) > 0 \tag{3.24}$$

Since $f(B, \psi) \geq 0$, we have found B^* is not optimal. Thus it must be true that $B^* = \underline{B}$ or $B^* = \overline{B}$. ■

In this lemma, we have assumed $a \in \mathcal{R}$ without loss in generality, since in general we can rotate both a and b by the phase of a. Although there is still conservatism in this bound since we considered the magnitude and phase to be independent, this overbound has generated accurate parameter estimates.

Using this lemma, we can determine a bound on $|\hat{g}_k(j\omega) - g_k(j\omega)|_{max}$. Using a priori bounds on the parameters a_{1k}, a_{0k}, b_{1k}, and b_{0k}, it is straightforward to bound

the magnitude and phase of $\hat{g}_k(j\omega)$. If $\pm\pi \in [\underline{\psi}, \overline{\psi}]$, then we use the triangle inequality. Otherwise, we apply lemma 3.1. Initially the a priori information may be very poor, so instead we can use a suitable constant determined through engineering judgment.

Since the system is lightly damped, this value is typically small near the natural frequencies of other modes. We will consider the amount our upper bound adds to the noise level an indication of the conservatism added due to mismatch in other modes. Although it can only be reduced through iteration, it is easy to measure the conservatism introduced in this process.

3.2.2 Embedding a One Mode Problem into a Convex Space

Let us now consider the set membership problem when we have a one mode system. We have now reduced the problem to one of the form

$$\left| g(j\omega_i) - \frac{b_1 j\omega_i + b_0}{\omega_i^2 + a_1 j\omega_i + a_0} \right| \leq E(j\omega_i) \quad i = 1, ..., N \tag{3.25}$$

Here, $E(j\omega_i)$ is the bound on the noise plus the bound on the errors in other modes. This is still a difficult problem because the set of feasible parameters is not, in general, a convex set. We will proceed by embedding the feasible set into a convex set. To do this, we will assume that we have a priori upper and lower bounds on the natural frequency ω_n, and an upper bound on the damping ratio ζ. Since $a_0 = \omega_n^2$ and $a_1 = 2\zeta\omega_n$, these bounds give us upper and lower bounds on a_0 and an upper bound on a_1.

We can rewrite our set-membership equations (3.25) by multiplying through by the denominator. We then overbound the right hand side of the equation as follows.

$$\begin{aligned} \left| g(j\omega_i)(-\omega_i^2 + a_1 j\omega_i + a_0) - (b_1 j\omega_i + b_0) \right| &\leq |-\omega_i^2 + a_1 j\omega_i + a_0| E(\omega_i) \\ &\leq \max_{a_0, a_1} |-\omega_i^2 + a_1 j\omega_i + a_0| E(\omega_i) \\ &= \max_{a_0 \in \{\underline{a}_0, \overline{a}_0\}} |-\omega_i^2 + \overline{a}_1 j\omega_i + a_0| E(\omega_i) \\ &\triangleq C(\omega_i) \end{aligned} \tag{3.26}$$

This equation can be rewritten in the form $x^T Q x \leq C^2$, where

$$x = \begin{bmatrix} a_1 \\ a_0 - \omega_i^2 \\ b_1 \\ b_0 \end{bmatrix} \qquad Q = \begin{bmatrix} \frac{g_R^2+g_I^2}{\omega^2} & 0 & -\frac{g_R}{\omega^2} & \frac{g_i}{\omega^3} \\ 0 & \frac{g_R^2+g_I^2}{\omega^4} & -\frac{g_i}{\omega^3} & -\frac{g_R}{\omega^4} \\ -\frac{g_R}{\omega^2} & -\frac{g_i}{\omega^3} & \frac{1}{\omega^2} & 0 \\ \frac{g_i}{\omega^3} & -\frac{g_R}{\omega^4} & 0 & \frac{1}{\omega^4} \end{bmatrix} \tag{3.27}$$

where $g_R = \text{Real}(g(j\omega_i))$, $g_I = \text{Imag}(g(j\omega_i))$, $\omega = \omega_i$, and $C = C(\omega_i)$.

Since each equation describes an ellipsoid, the set of parameters consistent with equation (3.26) is the intersection of N ellipsoids. Since the intersection of a finite number of convex sets is itself convex, we have found a convex set which contains the feasible set.

It is important to understand the significance of overbounding the set of feasible parameters with a set which is larger. We desire an overbounding set because we wish to ensure that the true feasible set is contained in our parameter intervals. We desire a convex set because, as will be shown, it is straightforward to determine interval ranges for a convex set.

With these convex feasible sets, it is fairly easy to determine corresponding parameter ranges. We need to solve the following set of problems.

Problem 3.2

1. *Determine upper bounds:*

$$\max \theta_i$$

$$\textit{s.t. } \theta \textit{ satisfies equation (3.26)} \quad i = 1, ..., N$$

$$\theta = \begin{bmatrix} a_1 & a_0 & b_1 & b_0 \end{bmatrix}^T$$

2. *Determine lower bounds:*

$$\min \theta_i$$

$$\textit{s.t. } \theta \textit{ satisfies equation (3.26)} \quad i = 1, ..., N$$

$$\theta = \begin{bmatrix} a_1 & a_0 & b_1 & b_0 \end{bmatrix}^T$$

Since this is a convex programming problem, it is considered a solved problem. For instance, it is possible to recast this problem using linear matrix inequalities [7]. It can then be solved using the software described in [20], [43].

Notice that once we have solved this problem, we may have tighter bounds on a_0 and a_1. We could then determine a tighter bound $C(\omega_i)$ in equation (3.26). With a tighter bound, we could then resolve the convex programming problems. We thus have the following algorithm.

Algorithm 3.2 (Set Membership ID for a Single Mode System)

1. *Estimate* $\overline{\omega}_n$, $\underline{\omega}_n$, $\overline{\zeta}$
2. *Calculate the bound* $C(\omega)$.
3. *Use convex programming to find the smallest axis-aligned box containing the set of parameters consistent with equation (3.26), for* $i = 1, ..., N$.
4. *Have the bounds on* ω_n *and* ζ *improved (more than some tolerance)? If yes, go to step 2. If no, stop.*

3.2.3 Reducing the Conservatism

We now analyze the conservatism of the one mode problem. If we divide equation (3.26) by $|-\omega_i^2 + a_1 j\omega_i + a_0|$, we see that we have solved for the set of parameters consistent with the following equation

$$\left| g(j\omega_i) - \frac{b_1 j\omega_i + b_0}{-\omega_i^2 + a_1 j\omega_i + a_0} \right| \leq \frac{\max_{a_0 \in \{\overline{a}_0, \underline{a}_0\}} |-\omega_i^2 + \overline{a}_1 j\omega_i + a_0|}{|-\omega_i^2 + a_1 j\omega_i + a_0|} E(j\omega_i) \qquad (3.28)$$

Let us define the "ratio of conservatism" due to the embedding as

$$\gamma \triangleq \max_{i=1,...,N} \frac{\max_{a_0 \in \{\overline{a}_0, \underline{a}_0\}} |-\omega_i^2 + \overline{a}_1 j\omega_i + a_0|}{|-\omega_i^2 + a_1 j\omega_i + a_0|} > 1 \qquad (3.29)$$

Let us discuss exactly what this ratio means. Let us assume we have a candidate value of the parameter vector $[a_1, a_0, b_1, b_0]^T$. Let us assume that it is not a feasible

value, but it is included in our convex set. Thus at some ω_i,

$$E(j\omega_i) < \left| g(j\omega_i) - \frac{b_1 j\omega_i + b_0}{-\omega_i^2 + a_1 j\omega_i + a_0} \right| \leq \frac{\max_{a_0 \in \{\overline{a}_0, \underline{a}_0\}} |-\omega_i^2 + \overline{a}_1 j\omega_i + a_0|}{|-\omega_i^2 + a_1 j\omega_i + a_0|} E(j\omega_i) \tag{3.30}$$

In the denominator on the right hand side, the values of a_1 and a_0 used are our candidate values, while the numerator contains the value which maximizes the expression. The conservatism ratio γ is an indication of how conservative our bound is. γ is always greater than or equal to one, and as γ gets closer to one, the box which results from our optimization gets closer to the optimal box around the nonconvex region. Clearly we would like to make γ small.

What we will now do is show how to guarantee γ is as small as we desire. To do so, we need to make the following assumption.

Assumption 3.1 *The lower bound on a_1, $\underline{a}_1$ is greater than 0.*

With our assumptions on the plant (i.e. strictly stable), this assumption is not an unreasonable one. Physically, this amounts to saying each mode has some damping. In fact, even if we initially assume that the lower bound is 0, the lower bound will in general become nonzero after a few iterations of algorithm 3.2.

To reduce the ratio of conservatism, we first need the following lemma.

Lemma 3.2 *Given $\delta \geq \frac{\overline{a}_0}{\underline{a}_0}$, $\delta > 1$. Then*

$$\frac{\max_{a_0 \in [\underline{a}_0, \overline{a}_0], a_1 \in [\underline{a}_1, \overline{a}_1]} | - w^2 + a_1 j\omega + a_0|}{\min_{a_0 \in [\underline{a}_0, \overline{a}_0], a_1 \in [\underline{a}_1, \overline{a}_1]} | - w^2 + a_1 j\omega + a_0|} \leq \delta \tag{3.31}$$

for all $\omega > 0$ if the following two inequalities hold

$$\overline{a}_1^2 \left(1 - \left(\frac{\underline{a}_1}{\overline{a}_1}\right)^2 \delta^2\right) \leq 2\underline{a}_0 \left(\sqrt{\delta^2 - 1}\sqrt{\delta^2 - \left(\frac{\overline{a}_0}{\underline{a}_0}\right)^2} - \left(\delta^2 - \frac{\overline{a}_0}{\underline{a}_0}\right)\right) \tag{3.32}$$

$$\overline{a}_1^2 \left(1 - \left(\frac{\underline{a}_1}{\overline{a}_1}\right)^2 \delta^2\right) \leq 2\underline{a}_0 \left(\sqrt{\delta^2 - 1}\sqrt{\left(\frac{\overline{a}_0}{\underline{a}_0}\right)^2 \delta^2 - 1} - \left(\frac{\overline{a}_0}{\underline{a}_0}\delta^2 - 1\right)\right) \tag{3.33}$$

Proof: Let us first determine the maximum and minimum values of the term $|-\omega^2 + a_1 j\omega + a_0|$ as a function of frequency. It is straightforward to see

$$\max_{a_1\in[\underline{a}_1,\overline{a}_1],a_0\in[\underline{a}_0,\overline{a}_0]} |-\omega^2 + a_1 j\omega + a_0| = \begin{cases} [(\overline{a}_0 - \omega^2)^2 + (\overline{a}_1\omega)^2]^{\frac{1}{2}} & \omega^2 \leq \frac{\underline{a}_0+\overline{a}_0}{2} \\ [(\underline{a}_0 - \omega^2)^2 + (\overline{a}_1\omega)^2]^{\frac{1}{2}} & \omega^2 \geq \frac{\underline{a}_0+\overline{a}_0}{2} \end{cases} \tag{3.34}$$

$$\min_{a_1\in[\underline{a}_1,\overline{a}_1],a_0\in[\underline{a}_0,\overline{a}_0]} |-\omega^2+a_1 j\omega+a_0| = \begin{cases} [(\underline{a}_0 - \omega^2)^2 + (\underline{a}_1\omega)^2]^{\frac{1}{2}} & \omega^2 \leq \underline{a}_0 \\ \underline{a}_1\omega & \underline{a}_0 \leq \omega^2 \leq \overline{a}_0 \\ [(\overline{a}_0 - \omega^2)^2 + (\underline{a}_1\omega)^2]^{\frac{1}{2}} & \omega^2 \geq \overline{a}_0 \end{cases} \tag{3.35}$$

To satisfy equation (3.31), we thus need to satisfy the following equations.

$$\begin{cases} (\overline{a}_0 - \omega^2)^2 + (\overline{a}_1\omega)^2 \leq \delta^2\left((\underline{a}_0 - \omega^2)^2 + (\underline{a}_1\omega)^2\right) & \omega^2 \leq \underline{a}_0 \\ (\overline{a}_0 - \omega^2)^2 + (\overline{a}_1\omega)^2 \leq \delta^2(\underline{a}_1\omega)^2 & \underline{a}_0 \leq \omega^2 \leq \frac{\underline{a}_0+\overline{a}_0}{2} \\ (\underline{a}_0 - \omega^2)^2 + (\overline{a}_1\omega)^2 \leq \delta^2(\underline{a}_1\omega)^2 & \frac{\underline{a}_0+\overline{a}_0}{2} \leq \omega^2 \leq \overline{a}_0 \\ (\underline{a}_0 - \omega^2)^2 + (\overline{a}_1\omega)^2 \leq \delta^2\left((\overline{a}_0 - \omega^2)^2 + (\underline{a}_1\omega)^2\right) & \omega^2 \geq \overline{a}_0 \end{cases} \tag{3.36}$$

Rearranging, we see that we need to satisfy the following set of inequalities.

$$\overline{a}_1^2 - \delta^2\underline{a}_1^2 \leq \begin{cases} \delta^2\left(\frac{\underline{a}_0}{\omega} - \omega\right)^2 - \left(\frac{\overline{a}_0}{\omega} - \omega\right)^2 & \omega^2 \leq \underline{a}_0 \\ -\left(\frac{\overline{a}_0}{\omega} - \omega\right)^2 & \underline{a}_0 \leq \omega^2 \leq \frac{\underline{a}_0+\overline{a}_0}{2} \\ -\left(\frac{\underline{a}_0}{\omega} - \omega\right)^2 & \frac{\underline{a}_0+\overline{a}_0}{2} \leq \omega^2 \leq \overline{a}_0 \\ \delta^2\left(\frac{\overline{a}_0}{\omega} - \omega\right)^2 - \left(\frac{\underline{a}_0}{\omega} - \omega\right)^2 & \omega^2 \geq \overline{a}_0 \end{cases} \tag{3.37}$$

Notice that in this expression we are looking over all positive frequencies. Our goal is therefore to find the frequency at which the right hand side of this expression is minimized, and check that the inequality (3.31) holds at this frequency.

Let us define the following functions for ease of notation.

$$f_1(\omega) = \delta^2\left(\frac{\underline{a}_0}{\omega} - \omega\right)^2 - \left(\frac{\overline{a}_0}{\omega} - \omega\right)^2 \tag{3.38}$$

$$f_2(\omega) = -\left(\frac{\overline{a}_0}{\omega} - \omega\right)^2 \tag{3.39}$$

$$f_3(\omega) = -\left(\frac{\underline{a}_0}{\omega} - \omega\right)^2 \tag{3.40}$$

$$f_4(\omega) = \delta^2\left(\frac{\overline{a}_0}{\omega} - \omega\right)^2 - \left(\frac{\underline{a}_0}{\omega} - \omega\right)^2 \tag{3.41}$$

Let us first consider the middle two equations. They are of the form

$$f_{23}(\omega) = -\left(\frac{x}{\omega} - \omega\right)^2 \tag{3.42}$$

Notice that this is a concave function. Thus, it is minimized at one of the endpoints. For each inequality, we need to check both endpoints to find the minimizing value. We see that

$$f_2\left(\sqrt{\underline{a}_0}\right) = -\frac{1}{\underline{a}_0}(\overline{a}_0 - \underline{a}_0)^2 \leq -\frac{1}{2(\overline{a}_0 + \underline{a}_0)}(\overline{a}_0 - \underline{a}_0)^2 = f_2\left(\sqrt{\frac{\overline{a}_0 + \underline{a}_0}{2}}\right) \tag{3.43}$$

$$f_3\left(\sqrt{\overline{a}_0}\right) = -\frac{1}{\overline{a}_0}(\overline{a}_0 - \underline{a}_0)^2 \leq -\frac{1}{2(\overline{a}_0 + \underline{a}_0)}(\overline{a}_0 - \underline{a}_0)^2 = f_3\left(\sqrt{\frac{\overline{a}_0 + \underline{a}_0}{2}}\right) \tag{3.44}$$

Thus we see that f_2 is minimized at $\omega = \sqrt{\underline{a}_0}$ and f_3 is minimized at $\omega = \sqrt{\overline{a}_0}$. Since $f_2(\sqrt{\underline{a}_0}) = f_1(\sqrt{\underline{a}_0})$, and $f_3(\sqrt{\overline{a}_0}) = f_4(\sqrt{\overline{a}_0})$, we see that the appropriate inequalities for f_2 and f_3 will be met if the inequalities for f_1 and f_4 are met. Thus, we will focus on these two inequalities.

Let us first consider f_1. Differentiating with respect to ω and setting equal to zero, we find that there is a unique stationary point over all $\omega > 0$ at

$$\omega_1^2 = \underline{a}_0 \frac{\sqrt{\delta^2 - \left(\frac{\overline{a}_0}{\underline{a}_0}\right)^2}}{\sqrt{\delta^2 - 1}} \tag{3.45}$$

Notice that this is well defined by our assumptions on δ. Also notice that this stationary point ω_1 satisfies $\omega_1^2 \leq \underline{a}_0$, so that this frequency point is in the acceptable

range for f_1. To see that this stationary point is actually the minimizing point for f_1, we use that fact that ω_1 is the unique stationary point and

$$\lim_{\omega \to 0} f_1(\omega) = \infty \tag{3.46}$$

$$\lim_{\omega \to \infty} f_1(\omega) = \infty \tag{3.47}$$

Evaluating $f_1(\omega_1)$, we get the right hand side of equation (3.32). Thus (3.32) guarantees that our inequalities are satisfied for f_1 and f_2.

Following the same steps for f_4, we find that the minimum value of f_4 is the right hand side of equation (3.33). Thus (3.33) guarantees that our inequalities are satisfied for f_3 and f_4.

In general we must check both the inequalities (3.32) and (3.33) since either one could be the binding constraint, depending upon the problem. ■

What this lemma indicates is that we can find a bound δ on the ratio of conservatism if we are given upper and lower bounds on a_0 and a_1. It is interesting to note that in the lemma, it is the ratios $\frac{\bar{a}_1}{\underline{a}_1}$ and $\frac{\bar{a}_0}{\underline{a}_0}$ which are important. In order to make the ratio of conservatism small, what we will do is to split up the parameter space into different regions where these ratios are small. To see how this is done, see figure 3-2.

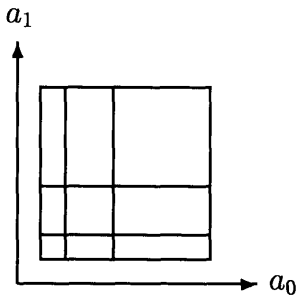

Figure 3-2: Splitting the parameter space into a finite number of regions.

In general, we will split the a priori bounds for a_0 into n_0 regions and the a priori bounds for a_1 into n_1 regions. In each of the regions we will keep the ratio of the

upper bound to lower bound for each parameter constant. We can then solve the set membership problem where we restrict a_0 and a_1 to each region. In each region, we determine the bound $C(j\omega_i)$ in equation (3.26). We also restrict our search over parameters to the region where a_1 and a_0 are in the partition. To combine the results, we simply take the maximum and minimum values found for each parameter over all partitions.

We are thus solving $n_0 n_1$ problems. First, we must show that for any $\delta > 1$, there are a finite number of partitions needed to guarantee that the ratio of conservatism is less that δ. We do this with the following lemmas.

Lemma 3.3 *If we partition a_0 into n_0 regions and a_1 into n_1 regions where in each partition the ratio of the upper bound to lower bound is constant, then in each region*

$$\frac{\max_{a_0 \in [\underline{a}_0, \overline{a}_0], a_1 \in [\underline{a}_1, \overline{a}_1]} \left| -w^2 + a_1 j\omega + a_0 \right|}{\min_{a_0 \in [\underline{a}_0, \overline{a}_0], a_1 \in [\underline{a}_1, \overline{a}_1]} \left| -w^2 + a_1 j\omega + a_0 \right|} \leq \delta$$

if the following inequalities hold

$$\overline{a}_1^2 \left(1 - \left(\frac{\underline{a}_1}{\overline{a}_1} \right)^{2/n_1} \delta^2 \right) \leq 2\underline{a}_0 \left(\sqrt{\delta^2 - 1} \sqrt{\delta^2 - \left(\frac{\overline{a}_0}{\underline{a}_0} \right)^{2/n_0}} - \left(\delta^2 - \left(\frac{\overline{a}_0}{\underline{a}_0} \right)^{1/n_0} \right) \right) \tag{3.48}$$

$$\overline{a}_1^2 \left(1 - \left(\frac{\underline{a}_1}{\overline{a}_1} \right)^{2/n_1} \delta^2 \right) \leq 2\underline{a}_0 \left(\sqrt{\delta^2 - 1} \sqrt{\left(\frac{\overline{a}_0}{\underline{a}_0} \right)^{2/n_0} \delta^2 - 1} - \left(\left(\frac{\overline{a}_0}{\underline{a}_0} \right)^{1/n_0} \delta^2 - 1 \right) \right) \tag{3.49}$$

$$\delta \geq \left(\frac{\overline{a}_0}{\underline{a}_0} \right)^{1/n_0} \tag{3.50}$$

Proof: Let us define ϵ_0 and ϵ_1 as

$$\epsilon_0 = \left(\frac{\overline{a}_0}{\underline{a}_0} \right)^{1/n_0} \qquad \epsilon_1 = \left(\frac{\overline{a}_1}{\underline{a}_1} \right)^{1/n_1} \tag{3.51}$$

Thus, in the k_0^{th} region for a_0 and the k_1^{th} region for a_1, the ranges for a_0 and a_1 are given by

$$a_0 \in \left[\underline{a}_0 \epsilon_0^{k_0 - 1}, \underline{a}_0 \epsilon_0^{k_0} \right] \tag{3.52}$$

$$a_1 \in \left[\underline{a}_1 \epsilon_1^{k_1-1}, \underline{a}_1 \epsilon_1^{k_1}\right] \tag{3.53}$$

Applying lemma 3.2 to this particular partition, we see that the equalities we wish to satisfy are given by

$$(\underline{a}_1 \epsilon_1^{k_1})^2 \left(1 - \left(\frac{1}{\epsilon_1}\right)^2 \delta^2\right) \leq 2\underline{a}_0 \epsilon_0^{k_0-1} \left(\sqrt{\delta^2-1}\sqrt{\delta^2-\epsilon_0^2} - \left(\delta^2 - \epsilon_0\right)\right) \tag{3.54}$$

$$(\underline{a}_1 \epsilon_1^{k_1})^2 \left(1 - \left(\frac{1}{\epsilon_1}\right)^2 \delta^2\right) \leq 2\underline{a}_0 \epsilon_0^{k_0-1} \left(\sqrt{\delta^2-1}\sqrt{\epsilon_0^2\delta^2-1} - \left(\epsilon_0\delta^2 - 1\right)\right) \tag{3.55}$$

$$\delta \geq \epsilon_0 \tag{3.56}$$

Since $\epsilon_0, \epsilon_1 \geq 1$, we have

$$\epsilon_0^{k_0} \geq \epsilon_0^{k_0-1} \tag{3.57}$$

$$\epsilon_1^{k_1} \geq \epsilon_1^{k_1-1} \tag{3.58}$$

Thus, the left hand side of the inequalities are maximized when $k_1 = n_1$. Similarly, the right hand side of the inequalities are minimized when $k_0 = 1$. Thus if the inequalities are satisfied for $k_1 = n_1$ and $k_0 = 1$, then they are satisfied for all partitions.

Substituting in for k_0, k_1, and using the definitions for ϵ_0 and ϵ_1, we recover the desired result. ■

We now show that given $\delta > 1$, there is always at least one feasible way to partition our parameter region with a finite number of partitions.

Lemma 3.4 *Given $\delta > 1$, $0 < \underline{a}_0 \leq \overline{a}_0$, $0 < \underline{a}_1 \leq \overline{a}_1$, then one feasible partitioning is given by*

$$n_1 > \frac{\log(\overline{a}_1) - \log(\underline{a}_1)}{\log(\delta)} \tag{3.59}$$

$$\epsilon = \min\left\{\frac{\overline{a}_1^2}{2\underline{a}_0}\left(\left(\frac{\underline{a}_1}{\overline{a}_1}\right)^{2/n_1}\delta^2 - 1\right), \frac{\delta^2-1}{2}\right\} \tag{3.60}$$

$$n_0 \geq \max\left\{\frac{2(\log(\overline{a}_0) - \log(\underline{a}_0))}{\log\left(\frac{\delta^2(\delta^2-1)-(\delta^2-1-\epsilon)^2}{\delta^2-1}\right)}, \frac{\log(\overline{a}_0) - \log(\underline{a}_0)}{\log\left(\frac{\epsilon+\delta^2}{\delta^2}\right)}\right\} \tag{3.61}$$

Proof: First, we will show that the right hand sides of equations (3.48) and

(3.49) are both increasing functions of n_0. Let us define the function

$$f(r) = \left(\delta^2 - 1\right)^{\frac{1}{2}} \left(\delta^2 - r^2\right)^{\frac{1}{2}} - (\delta^2 - r) \tag{3.62}$$

The first derivative of this function is given by

$$\frac{\partial f(r)}{\partial r} = -\left(\delta^2 - 1\right)^{\frac{1}{2}} \left(\delta^2 - r^2\right)^{-\frac{1}{2}} r + 1 \tag{3.63}$$

For $r \geq 1$,

$$\left(\delta^2 - 1\right)^{\frac{1}{2}} \left(\delta^2 - r^2\right)^{-\frac{1}{2}} \geq 1 \tag{3.64}$$

Thus the first derivative of $f(r)$ is negative for all $r \geq 1$. We will use $f(r)$ with r defined by the following.

$$r = \left(\frac{\overline{a}_0}{\underline{a}_0}\right)^{1/n_0} \geq 1 \tag{3.65}$$

As n_0 increases, r decreases. Since the derivative of $f(r)$ with respect to r is negative, as r decreases, $f(r)$ increases. This shows that the right hand side of equation (3.48) is an increasing function of n_0. Furthermore, we have

$$\lim_{n_0 \to \infty} \left(\delta^2 - 1\right)^{\frac{1}{2}} \left(\delta^2 - \left(\frac{\overline{a}_0}{\underline{a}_0}\right)^{2/n_0}\right)^{\frac{1}{2}} - \left(\delta^2 - \left(\frac{\overline{a}_0}{\underline{a}_0}\right)^{1/n_0}\right) = 0 \tag{3.66}$$

Thus we see the right hand side of equation (3.48) is always nonpositive.

The same properties hold for the right hand side of equation (3.49), and can be shown in exactly the same way. Thus, we have shown that the right hand side of our inequalities are negative. For the inequalities to hold, we need the left hand sides to be negative as well. This will be true if we choose

$$n_1 > \frac{\log(\overline{a}_1) - \log(\underline{a}_1)}{\log(\delta)} \tag{3.67}$$

Notice that n_1 is well defined since $\delta > 1$.

With the definition of ϵ, it is clear that we need a n_0 such that

$$-\epsilon \leq \left(\delta^2-1\right)^{\frac{1}{2}}\left(\delta^2-\left(\frac{\overline{a}_0}{\underline{a}_0}\right)^{2/n_0}\right)^{\frac{1}{2}}-\left(\delta^2-\left(\frac{\overline{a}_0}{\underline{a}_0}\right)^{2/n_0}\right) \tag{3.68}$$

$$-\epsilon \leq \left(\delta^2-1\right)^{\frac{1}{2}}\left(\left(\frac{\overline{a}_0}{\underline{a}_0}\right)^{2/n_0}\delta^2-1\right)^{\frac{1}{2}}-\left(\left(\frac{\overline{a}_0}{\underline{a}_0}\right)^{2/n_0}\delta^2-1\right) \tag{3.69}$$

Such a n_0 exists since the right hand sides are increasing functions of n_0, and become 0 as $n_0 \to \infty$. Straightforward manipulation confirms that choosing n_0 as in the lemma satisfies these bounds. It should be noted that the value of n_0 is well defined by allowing ϵ to be no larger than $\frac{1}{2}(\delta^2-1)$. ■

Given δ, let us call a partition (n_0, n_1) feasible if equations (3.48)--(3.50) hold. In order to have the least number of partitions, we would like to minimize the product $n_0 n_1$ over all feasible partitions. Given that there is a feasible solution, this can be done with a brute force search method. This search is made easier using the following facts, which are a direct result of the previous lemmas.

Corollary 3.1

1. *If* (n_0, n_1) *are feasible for* $\delta_1 > 1$, *then they are also feasible for any* δ_2 *where* $\delta_2 \geq \delta_1$.

2. *If* (n_0, n_1) *are infeasible for* $\delta_1 > 1$, *then they are also infeasible for any* δ_2 *where* $\delta_2 \leq \delta_1$.

3. *If* (n_0, n_1) *are feasible for* $\delta > 1$, *then so are* (k_0, k_1) *where* $k_0 \geq n_0$ *and* $k_1 \geq n_1$.

4. *If* (n_0, n_1) *are infeasible for* $\delta > 1$, *then so are* (k_0, k_1) *where* $k_0 \leq n_0$ *and* $k_1 \leq n_1$.

In summary, we have shown that we can embed our problem into a problem which is convex. The added conservatism due to this embedding can be made smaller than any specified value by partitioning the a priori bounds into a finite number of partitions.

3.3 Summary of the Parameter Identification Algorithm

Based upon the results presented in this chapter, we can summarize the steps in the parametric identification as follows. We use algorithm 3.1 to generate parameter intervals. In step 2, we can specify a desired level of conservatism due to embedding the one mode problem into a convex space. We determine the appropriate partitioning of the parameter space, and apply algorithm 3.2 to each partition. In order to restrict the optimization problems to a particular partition, we include the following convex constraints in the optimization.

$$\underline{a}_0(k_0) \leq a_0 \leq \overline{a}_0(k_0) \qquad \underline{a}_1(k_1) \leq a_1 \leq \overline{a}_1(k_1) \tag{3.70}$$

where $\underline{a}_0(k_0)$ and $\overline{a}_0(k_0)$ are the lower and upper bounds to the k_0^{th} partition for a_0, with similar notation for a_1. To combine the results from these optimization problems, we find the upper and lower bounds to each parameter over all partitions.

A block diagram of this algorithm is shown in figure 3-3.

3.4 Summary

This chapter has solved the main identification problem addressed in this thesis. We are given a set of frequency domain data for a lightly damped structure. We have parameterized the system in a fashion which is numerically stable. We would like to determine interval ranges for each of the parameters such that the true system is guaranteed to lie within the interval ranges given our a priori assumptions on the noise.

This problem can be contrasted with previous work in that we are assuming a large amount of a priori information. We know the order of the system, and a priori bounds on the natural frequency and damping ratio of each mode. Unfortunately, this also leads to a problem which is nonconvex.

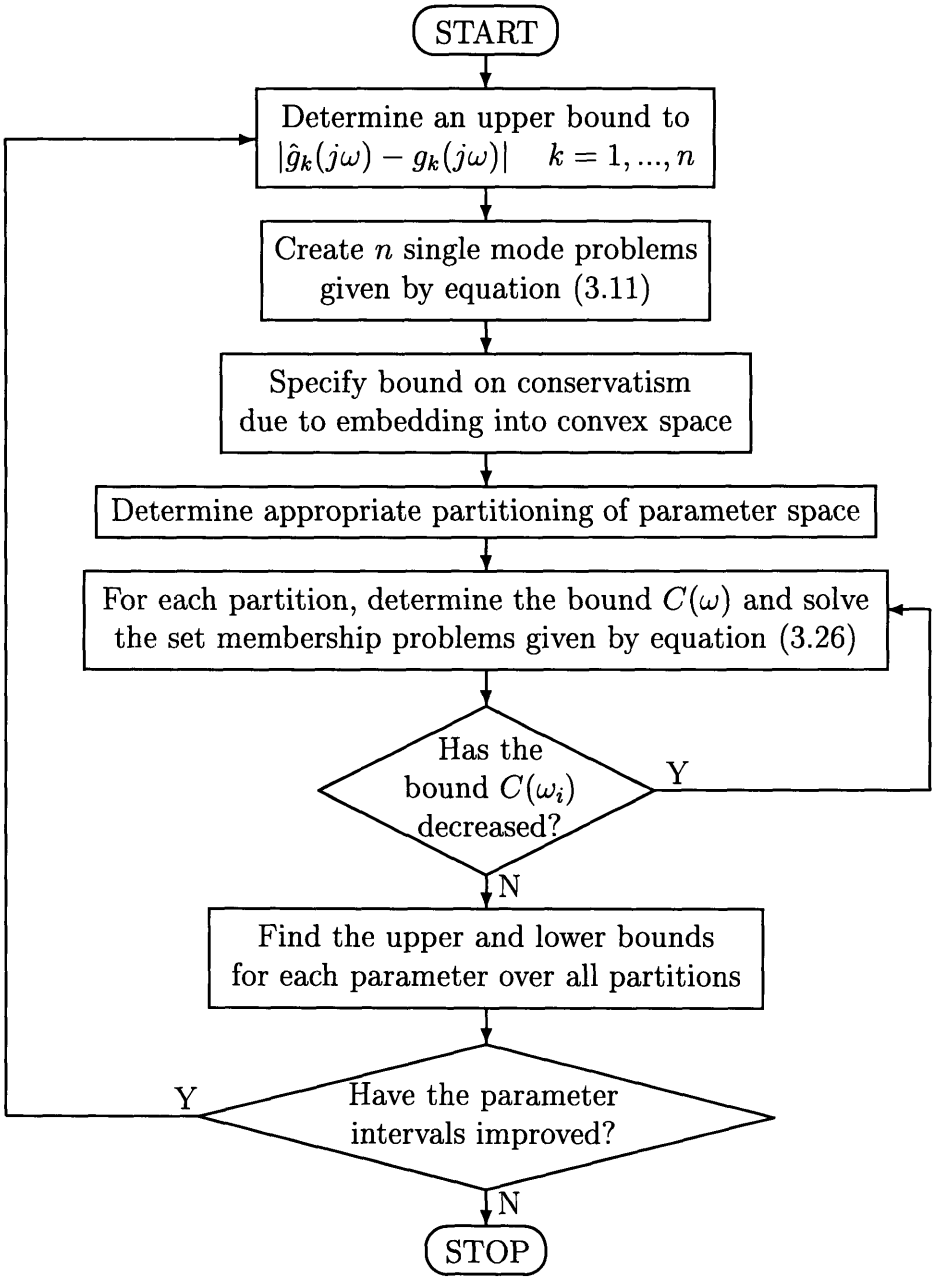

Figure 3-3: Block diagram of the identification algorithm.

In order to guarantee that our intervals contain the true system, we have applied the following procedure. First, using our knowledge that the modes are separated in frequency, we have reduced the problem to a set of one mode problems. While looking at each mode, we subtract the estimates of the other modes, and take into account the possible errors as part of our noise bound. The added conservatism can be easily measured. It is important to note that this conservatism can not necessarily be removed through iteration. A counterexample to this type of convergence is presented in section 6.2.1. In the example, a significant model mismatch in the first mode of the system prevents accurate identification of the parameters of the other modes.

For each of the one mode problems, we have embedded the set of feasible parameters into a larger convex set. We determine the interval ranges using convex programming. The added conservatism due to the embedding can be made arbitrarily small by partitioning the parameter space into a finite number of regions, and solving the convex programming problems in each region.

In summary, we have outlined a procedure to solve the problem of identifying parametric uncertainty. The novelty of the approach was our ability to use a parameterization which is nonlinear in the data.

Chapter 4

Computation of Inputs

4.1 Overview

We now begin to examine the problem of determining which input to apply to our system in order to improve our closed loop robust performance measure as much as possible. This is done through a sensitivity analysis. We apply the chain rule to create two quantities we must calculate: the μ–sensitivities and the expected effect of the inputs on the uncertainties. In this chapter, these quantities will be defined, and their computation is examined.

We consider the case when we have a finite number of choices for the next input. For each of these choices, we would like to determine the expected improvement in the performance measure. We can not determine the actual improvement a priori due to the noise in the system. Instead, we would like to calculate the expected change in performance given a specific input. This is represented as

$$\mathcal{E}_n \{\Delta\overline{\mu} \mid u\} \tag{4.1}$$

Here, $\mathcal{E}_n$ represents an expectation over the measurement noise and the uncertainty. For this to have meaning, we must now make stronger assumptions on the noise.

In determining the effects of the inputs, we assume that our current nominal model is the true system, and the errors in our model (including unmodelled dynamics) are

considered part of the noise. Thus, one portion of the noise is a fixed but unknown deterministic error due to this model mismatch. We will also assume that there is a zero mean stochastic component to the noise. We will write

$$n(j\omega) = n_d(j\omega) + n_s(j\omega) \tag{4.2}$$

The stochastic portion, $n_s(j\omega)$, will have a probability density function given by $p_{n_s}(n)$. Thus, the expectation in equation (4.1) is with respect to this probability density function, and n_d. Note that in this setting, n_d can be considered the unknown mean of the noise.

Our theory allows for several different parametric models for performance. We assume that the system performance is defined by p variables α_i. These variables could be the size of the uncertainties of each parameter, or they could be the upper and lower bounds to each parameter. Both of these possibilities will be examined in more detail later.

We will apply the chain rule to equation (4.1), and use a first order approximation as follows.

$$\mathcal{E}_n\left\{\Delta\overline{\mu} \mid u\right\} \approx \sum_{i=1}^{p} \frac{\partial\overline{\mu}}{\partial\alpha_i}\mathcal{E}_n\left\{\Delta\alpha_i \mid u\right\} \tag{4.3}$$

Notice that there are two terms we need to calculate. The first is the sensitivity of $\overline{\mu}$ with respect to the parameters α_i. Essentially this tells us how $\overline{\mu}$ will change as the parameters change. This will be explored in the next section. The other term is the expected effect the new inputs will have on each of the parameters. For instance, if the parameters α_i represented the size of the uncertainty in each of the parameters, this term would be the expected decrease in the uncertainty with each new input.

In the following sections, we will examine these quantities in more detail. We will stress the computation of the appropriate quantities, and defer discussion of convergence issues in the iterative framework to chapter 5.

4.2 μ–Sensitivities

The sensitivity of μ to a given parameter is a problem which has received considerable attention. This type of analysis was first introduced in [19]. Here, the robustness was measured using the singular values of the closed loop system, and the sensitivities of the singular values to a perturbation in the system were determined. Unfortunately, the maximum singular value is too conservative a measure of robustness when there are real uncertain parameters. The approach was extended in [8], where the μ–sensitivities were introduced. This is essentially the sensitivity of μ to the size of the uncertainty in the parameters. The approach taken was to approximate the sensitivity using finite differences.

Here, a new approach to solving the μ–sensitivity problem is presented. Once we have determined $\overline{\mu}$, the sensitivity calculation involves only an eigenvalue decomposition. The calculation can thus be done more efficiently than by finite difference approximations.

4.2.1 Definitions

We consider the closed loop system shown in figure 2-1. We determine the measure of robust performance of this closed loop system using the mixed μ problem of section 2.2. The robust performance metric $\overline{\mu}$ is given in definition 2.2, and is repeated here for convenience.

$$\mathcal{D} = \{diag(d_1, ..., d_q) : 0 < d_i \in \mathcal{R}\} \tag{4.4}$$

$$\mathcal{G} = \{diag(g_1, ..., g_p, 0, ..., 0) : g_i \in \mathcal{R}\} \tag{4.5}$$

$$\overline{\mu}(M) = \sup_{\omega} \inf_{D\in\mathcal{D},G\in\mathcal{G}} \left[\min_{\beta\geq 0} \left\{\beta : \overline{\lambda}(M_D^*(\omega)M_D(\omega) + j(GM_D(\omega) - M_D^*(\omega)G)) \leq \beta^2\right\}\right] \tag{4.6}$$

$$M_D(\omega) \triangleq DM(\omega)D^{-1} \tag{4.7}$$

where M is the nominal closed loop system, and $\overline{\lambda}(X)$ denotes the maximum eigenvalue of X.

As a practical matter, we only check the value of $\overline{\mu}$ over a finite set of frequencies.

We will thus assume that there is a *finite* set Ω over which we determine $\overline{\mu}$. Our robust performance measure $\overline{\mu}$ can therefore be redefined as follows.

Definition 4.1

$$\overline{\mu}(M) = \max_{\omega\in\Omega} \inf_{D\in\mathcal{D},G\in\mathcal{G}} \left[\min_{\beta\geq 0} \left\{\beta : \overline{\lambda}(M_D^*(\omega)M_D(\omega) + j(GM_D(\omega) - M_D^*(\omega)G)) \leq \beta^2\right\}\right] \tag{4.8}$$

where $M_D(\omega) \triangleq DM(\omega)D^{-1}$.

Let us now examine what is meant by the μ sensitivities. We want to understand how the value of $\overline{\mu}$ changes as a parameter in the closed loop system changes. This parameter could be the size of one of the uncertainties, or it could be any other parameter which changes in a smooth fashion. This will be made more clear in the sequel. For now, we will assume that there is a parameter ϵ, and we want to determine the sensitivity of $\overline{\mu}(M)$ around $\epsilon = 0$.

For notational convenience, we will write the closed loop system as M_ϵ, so that $M_0 = M$. The problem of finding the sensitivity of $\overline{\mu}$ is now clear. The sensitivity is defined by

$$\frac{\partial}{\partial\epsilon}\overline{\mu}(M_\epsilon) \tag{4.9}$$

We will examine how to calculate this quantity in the next section.

4.2.2 Calculation of the μ-Sensitivities

We have restricted our calculation of $\overline{\mu}$ to a finite number of frequency points. We will therefore have a finite set of points where the maximum of equation (4.8) is achieved. The performance of the system is thus the value of $\overline{\mu}$ at these frequencies, and we would need to calculate the μ-sensitivities only at these frequencies. The frequency with the smallest decrease (or largest increase) in $\overline{\mu}$ as we change ϵ will define the new value of the performance. Thus, we can calculate the μ-sensitivities at each of the frequencies where the maximum is achieved, and the μ-sensitivity is defined as the largest.

From this discussion, we see that we can restrict the problem to finding the μ-sensitivity at a particular frequency point. For notational convenience, we will assume that M_ϵ is a fixed matrix, and not a function of frequency. Thus, we are concerned with the following quantity.

$$\overline{\mu}(M_\epsilon) = \inf_{D\in\mathcal{D},G\in\mathcal{G}} \left[\min_{\beta\geq 0}\left\{\beta : \overline{\lambda}(M_{D\epsilon}^* M_{D\epsilon} + j(GM_{D\epsilon} - M_{D\epsilon}^* G)) \leq \beta^2\right\}\right] \tag{4.10}$$

where $M_{D\epsilon} \triangleq DM_\epsilon D^{-1}$.

To begin the calculation of the μ–sensitivities, we will first make an assumption which says that we can achieve the infimum of equation (4.10). Essentially, this implies that our problem is nondegenerate.

Assumption 4.1 *For each ϵ in a small neighborhood around zero, there exists $\gamma > 0$, and scalars d_i, g_j such that*

$$\gamma < d_i < \frac{1}{\gamma} \quad i = 1, ..., q \tag{4.11}$$

$$-\frac{1}{\gamma} < g_j < \frac{1}{\gamma} \quad j = 1, ..., p \tag{4.12}$$

$$\overline{D} = diag(d_1, ..., d_q) \tag{4.13}$$

$$\overline{G} = diag(g_1, ..., g_p, 0, ..., 0) \tag{4.14}$$

$$(\overline{\mu}(M_\epsilon))^2 = \overline{\lambda}(M_{\overline{D}\epsilon}^* M_{\overline{D}\epsilon} + j(\overline{G}M_{\overline{D}\epsilon} - M_{\overline{D}\epsilon}^* \overline{G})) \tag{4.15}$$

Thus, we have achieved the infimum at some optimal scalings $\overline{D}$ and $\overline{G}$ for each ϵ.

We will now outline the general methodology for calculating the μ–sensitivities. It should be noted that results similar to those to be presented here have appeared in [59]. However, the results in [59] were limited to the case where all the uncertainties are complex, and the maximum eigenvalue in (4.10) is unique. As shown in [56], this is only true when μ equals its upper bound, and thus is very restrictive.

To understand the philosophy behind the calculation of μ–sensitivities, consider

the problem

$$\min_x f(x, \epsilon) \tag{4.16}$$

where f is differentiable with respect to x and ϵ, and for each ϵ the minimum is achieved. Let $\overline{x}(\epsilon)$ be the value which achieves the global minimum, as well as a local minimum (so the gradient at $\overline{x}(\epsilon)$ is zero). Assuming that $\overline{x}(\epsilon)$ is differentiable as a function of ϵ, we would like to calculate

$$\frac{d}{d\epsilon} f(\overline{x}(\epsilon), \epsilon) \tag{4.17}$$

The following lemma tells us how to do this.

Lemma 4.1 *Assume*

$$\nabla_x f(x, \epsilon) \Big|_{x=\overline{x}(\epsilon)} = 0 \tag{4.18}$$

$$\nabla^2_{xx} f(x, \epsilon) \Big|_{x=\overline{x}(\epsilon)} > 0 \tag{4.19}$$

Then we have

$$\frac{d}{d\epsilon} f(\overline{x}(\epsilon), \epsilon) = \frac{\partial}{\partial \epsilon} f(x, \epsilon) \Big|_{x=\overline{x}(\epsilon)} \tag{4.20}$$

Proof: First, we note that equations (4.18) and (4.19) allow us to use the Implicit Function Theorem [47] to show that $\overline{x}(\epsilon)$ is a well defined and differentiable function. We can therefore use the multivariable chain rule as follows.

$$\frac{d}{d\epsilon} f(\overline{x}(\epsilon), \epsilon) = \frac{\partial}{\partial \epsilon} f(x, \epsilon) \Big|_{x=\overline{x}(\epsilon)} + (\nabla_x f(x, \epsilon))^T \Big|_{x=\overline{x}(\epsilon)} \frac{\partial}{\partial \epsilon} \overline{x}(\epsilon) \tag{4.21}$$

By assumption, the second term equals zero. ■

To use lemma 4.1, we let

$$x = \begin{bmatrix} d_1 & \dots & d_q & g_1 & \dots & g_p \end{bmatrix}^T \tag{4.22}$$

and $f(x, \epsilon) = \overline{\mu}(M_\epsilon)$. By assumption 4.1, the gradient with respect to the scalings of $\overline{\mu}(M_\epsilon)$ at the optimizing point is zero.

Remark 4.1 *In Lemma 4.1, we assumed that the hessian of f is positive definite. For $\overline{\mu}$, this can be made true in general by setting one of the scaling elements d_i to be equal to 1. We are thus excluding degenerate problems, such as when M is diagonal and any set of scalings will achieve the minimum.*

All that remains is to determine how to actually calculate the derivatives in question. Note that we are taking the derivative of the eigenvalues of a symmetric linear operator. We need to ensure that the derivatives of the eigenvalues and eigenvectors exist. This is done in the following theorem, which is taken from Chapter 2 of [32].

Theorem 4.1 *Let $X(\alpha) \in \mathcal{C}^{n\times n}$ be a well defined and holomorphic function of $\alpha \in \mathcal{R}$ in some open neighborhood $B(\overline{\alpha})$ of $\overline{\alpha}$. Let us also assume that $X(\alpha)$ is symmetric, i.e. $X^*(\alpha) = X(\alpha)$.*

1. *For each $\alpha \in B(\overline{\alpha})$, there are functions $\lambda_i(\alpha) \in \mathcal{C}$ and $u_i(\alpha) \in \mathcal{C}^n$, $i = 1, ..., n$, such that $\lambda_i(\alpha)$ is an eigenvalue of $X(\alpha)$ with eigenvector $u_i(\alpha)$.*

2. *The matrix $U(\alpha)$, which has i^{th} column $u_i(\alpha)$, is orthonormal for each value of $\alpha \in B(\overline{\alpha})$.*

3. *$\lambda_i(\alpha)$ and $u_i(\alpha)$, $i = 1, ..., n$ are differentiable functions of α in $B(\overline{\alpha})$.*

Much work exists to determine the derivatives of the eigenvalues of a matrix. See for example [53]. Since we know that the derivatives exist, it is actually much simpler to derive equations for the derivatives. This is given in the next theorem, whose proof is simple linear algebra manipulations and follows the proof for singular values given in [19].

Theorem 4.2 *Let $X(\alpha) \in \mathcal{C}^{n\times n}$ be symmetric, and let λ_1 be an eigenvalue of $X(\overline{\alpha})$ repeated m times. For α in a small open neighborhood $B(\overline{\alpha})$ of $\overline{\alpha}$, let $X(\alpha)$ have the eigenvalue decomposition $X(\alpha) = U(\alpha)\Lambda(\alpha)U^*(\alpha)$ where we have*

$$U(\alpha) = \begin{bmatrix} U_1(\alpha) & U_2(\alpha) \end{bmatrix} \tag{4.23}$$

$$U_1(\alpha) \in \mathcal{C}^{n\times m} \quad U_2(\alpha) \in \mathcal{C}^{n\times(n-m)} \tag{4.24}$$

$$\Lambda(\alpha) = \begin{bmatrix} \Lambda_1(\alpha) & 0 \\ 0 & \Lambda_2(\alpha) \end{bmatrix} \tag{4.25}$$

$$\Lambda_1(\alpha) \in \mathcal{C}^{m\times m} \quad \Lambda_2(\alpha) \in \mathcal{C}^{(n-m)\times(n-m)} \tag{4.26}$$

Here, $\Lambda_1(\overline{\alpha}) = \lambda_1 I$. *Then the derivatives of the eigenvalues* $\Lambda_1(\alpha)$ *evaluated at* $\alpha = \overline{\alpha}$ *are given by*

$$\frac{\partial}{\partial\alpha}\Lambda_1(\alpha)\Big|_{\alpha=\overline{\alpha}} = U_1^*(\overline{\alpha})\frac{\partial}{\partial\alpha}X(\alpha)\Big|_{\alpha=\overline{\alpha}} U_1(\overline{\alpha}) \tag{4.27}$$

Proof: First, we note that such a eigenvalue decomposition exists because $X(\alpha)$ is symmetric, and the previous theorem guarantees the existence of differentiable eigenvalues and eigenvectors.

By the eigendecomposition, we know that

$$U_1^*(\alpha)X(\alpha)U_1(\alpha) = \Lambda_1(\alpha) \tag{4.28}$$

Taking derivatives of this equation, and evaluating at $\alpha = \overline{\alpha}$ gives

$$\frac{\partial}{\partial\epsilon}U_1^*(\alpha)\Big|_{\alpha=\overline{\alpha}} X(\overline{\alpha})U_1(\overline{\alpha}) + U_1^*(\overline{\alpha})X(\overline{\alpha})\frac{\partial}{\partial\epsilon}U_1(\alpha)\Big|_{\alpha=\overline{\alpha}} + U_1^*(\overline{\alpha})\frac{\partial}{\partial\epsilon}X(\alpha)\Big|_{\alpha=\overline{\alpha}} U_1(\overline{\alpha})$$

$$= \frac{\partial}{\partial\epsilon}\Lambda_1(\alpha)\Big|_{\alpha=\overline{\alpha}} \tag{4.29}$$

We use the properties that $X(\overline{\alpha})U_1(\overline{\alpha}) = \lambda_1 I U_1(\overline{\alpha})$ and $U_1^*(\overline{\alpha})X(\overline{\alpha}) = U_1^*(\overline{\alpha})\lambda_1 I$ to get

$$\lambda_1\frac{\partial}{\partial\epsilon}U_1^*(\alpha)\Big|_{\alpha=\overline{\alpha}} U_1(\overline{\alpha}) + \lambda_1 U_1^*(\overline{\alpha})\frac{\partial}{\partial\epsilon}U_1(\alpha)\Big|_{\alpha=\overline{\alpha}} + U_1^*(\overline{\alpha})\frac{\partial}{\partial\epsilon}X(\alpha)\Big|_{\alpha=\overline{\alpha}} U_1(\overline{\alpha})$$

$$= \frac{\partial}{\partial\epsilon}\Lambda_1(\alpha)\Big|_{\alpha=\overline{\alpha}} \tag{4.30}$$

We note that the first two terms sum to zero since

$$0 = \frac{\partial}{\partial\epsilon}I \;=\; \frac{\partial}{\partial\epsilon}(U_1^*(\alpha)U_1(\alpha))\Big|_{\alpha=\overline{\alpha}} \tag{4.31}$$

$$= \frac{\partial}{\partial \epsilon} U_1^*(\alpha)\Big|_{\alpha=\overline{\alpha}} U_1(\overline{\alpha}) + U_1^*(\overline{\alpha}) \frac{\partial}{\partial \epsilon} U_1(\alpha)\Big|_{\alpha=\overline{\alpha}} \tag{4.32}$$

Thus we have

$$U_1^*(\overline{\alpha}) \frac{\partial}{\partial \epsilon} X(\alpha)\Big|_{\alpha=\overline{\alpha}} U_1(\overline{\alpha}) = \frac{\partial}{\partial \epsilon} \Lambda_1(\alpha)\Big|_{\alpha=\overline{\alpha}} \tag{4.33}$$

■

Theorem 4.3 *Under the assumptions of Theorem 4.2, let V_1 be any orthonormal set of vectors which span the right eigenspace of $X(\overline{\alpha})$ corresponding to the maximum eigenvalue. Then the derivatives of the eigenvalues of $\Lambda_1(\alpha)$ evaluated at $\alpha = \overline{\alpha}$ are given by the eigenvalues of the matrix*

$$V_1^* \frac{\partial}{\partial \alpha} X(\alpha)\Big|_{\alpha=\overline{\alpha}} V_1 \tag{4.34}$$

Proof: Let $U_1 \triangleq U_1(\overline{\alpha}) \in \mathcal{C}^{n\times m}$. U_1 is an orthonormal set of eigenvectors. Let $V_1 \in \mathcal{C}^{n\times m}$ be an orthonormal set of vectors which span the same space as U_1. We will write U_1 and V_1 in terms of their columns as follows.

$$U_1 = [u_{11}, ..., u_{1m}] \qquad V_1 = [v_{11}, ..., v_{1m}] \tag{4.35}$$

Since U_1 and V_1 span the same space, there must be constants α_{ji} such that

$$v_{1i} = \sum_{j=1}^{m} \alpha_{ji} u_{1j} \tag{4.36}$$

Define the matrix T by

$$T = \begin{bmatrix} \alpha_{11} & \dots & \alpha_{1m} \\ \vdots & & \vdots \\ \alpha_{m1} & \dots & \alpha_{mm} \end{bmatrix} \tag{4.37}$$

We now have

$$V_1 = U_1 T \tag{4.38}$$

For V_1 to be orthonormal, T must be an invertible matrix. Thus T is a similarity transformation, and we have

$$\begin{aligned}\lambda_i(V_1^* \frac{\partial}{\partial \alpha} X(\alpha)\Big|_{\alpha=\overline{\alpha}} V_1) &= \lambda_i(T^* U_1^* \frac{\partial}{\partial \alpha} X(\alpha)\Big|_{\alpha=\overline{\alpha}} U_1 T) \\ &= \lambda_i(U_1^* \frac{\partial}{\partial \alpha} X(\alpha)\Big|_{\alpha=\overline{\alpha}} U_1) \end{aligned} \tag{4.39}$$

■

Thus we can perform the following steps in calculating the μ-sensitivities.

1. Let the optimal scaling matrices in (4.10) be given by $\overline{D}$ and $\overline{G}$, and assume that the value of $\overline{\mu}$ equals $\overline{\beta}$.

2. Calculate the eigenstructure of the matrix in (4.10) such that

$$U \Lambda U^* = M_{\overline{D}}^* M_{\overline{D}} + j(\overline{G} M_{\overline{D}} - M_{\overline{D}}^* \overline{G}) \tag{4.40}$$

   Assume that the maximum eigenvalue is repeated m times, and let U_1 be the first m columns of U.

3. Define the matrix M_ϵ as above to represent the perturbation under consideration.

4. Let X be the matrix calculated by

$$X = \frac{\partial}{\partial \epsilon} M_{\overline{D}\epsilon}^*\Big|_{\epsilon=0} M_{\overline{D}} + M_{\overline{D}}^* \frac{\partial}{\partial \epsilon} M_{\overline{D}\epsilon}\Big|_{\epsilon=0} + j\left(\overline{G} \frac{\partial}{\partial \epsilon} M_{\overline{D}\epsilon}\Big|_{\epsilon=0} - \frac{\partial}{\partial \epsilon} M_{\overline{D}\epsilon}^*\Big|_{\epsilon=0} \overline{G}\right) \tag{4.41}$$

5. The μ-sensitivity is calculated by finding the largest eigenvalue of $U_1^* X U_1$. We find the largest eigenvalue since not all of the m eigenvalues increase at the same rate, and we are interested in the one that increases the fastest. Since we are interested in the sensitivity of $\overline{\mu}$ and not $\overline{\mu}^2$, we must also divide be $2\overline{\beta}$.

4.2.3 Application of the μ–Sensitivities

We will now discuss how to use the μ-sensitivities for our particular application. In the preceding discussion, we assumed that we had access to a well defined function M_ϵ describing the closed loop effect of perturbations ϵ. For us, determining the sensitivity of $\overline{\mu}$ requires the sensitivity of the control design methodology (2.29) to perturbations in the uncertainty. Thus, as will be discussed below, we will not be able to determine the sensitivities exactly. Instead, we will use an approximation, and discuss a property of the control methodology which makes this approximation reasonable.

What we would like to determine is the sensitivity of $\overline{\mu}$ as the parameter bounds change. In general, the upper bound and the lower bound will not change by the same amount for a new data point. Thus, we would expect the midpoint of the parameter intervals to move, and we must account for this while determining the sensitivity. We thus are interested in the how $\overline{\mu}$ changes as the bound $\overline{\theta}_j$ changes, and how it changes as the bound $\underline{\theta}_j$ changes. We can represent these perturbations as $\overline{\theta}_j(1-\epsilon)$ and $\underline{\theta}_j(1+\epsilon)$ respectively. More generally, we will represent this change as $\phi(\epsilon)$.

Using this framework, the μ–sensitivity can be written as

$$\frac{\partial}{\partial \epsilon}\overline{\mu}(M(\phi(\epsilon), f(\phi(\epsilon))))\Big|_{\epsilon=0} \tag{4.42}$$

Notice that we must take into account the changes to the nominal closed loop system M from both changes in the nominal plant and from changes in the compensator. Although the changes to the nominal model are complicated, they do not pose any conceptual difficulty, and it is straightforward to calculate these changes for a state space description. However, we typically are unable to get an analytic expression for the changes in the compensator due to changes in ϕ. When the control design includes an iterative method to solve an optimization problem, or when the compensator includes some sort of tuning, we are unable to determine this sensitivity.

One possible approximation is to assume that the compensator remains fixed.

Thus, we are determining

$$\frac{\partial}{\partial \epsilon}\overline{\mu}(M(\phi(\epsilon), C))\Big|_{\epsilon=0} \tag{4.43}$$

This approximation in general will be very pessimistic. We are letting the nominal model of the system change, but not retuning the compensator to the new model. The result is that the sensitivity analysis could easily conclude that improving the parameter bounds would make $\overline{\mu}$ worse. In order to avoid this result, we will use a different approximation.

We will assume that as the midpoint of the parameter intervals shift, the compensator is retuned so that $\overline{\mu}$ is approximately the same. In other words, the performance is not changed by much as the nominal model shifts because the compensator is redesigned for the new model. Instead, the changes in $\overline{\mu}$ are attributed to the changes in the size of the uncertainty intervals.

This argument allows us to calculate the sensitivities by assuming that the size of the uncertainty interval decreases, but the midpoint remains fixed. For instance, consider the case when we determine the sensitivity of $\overline{\mu}$ to the j^{th} uncertainty interval. We want to determine the change in $\overline{\mu}$ as δ_j reduces by ϵ. We will represent this by $\delta(\epsilon)$. We can then write the μ–sensitivities as

$$\frac{\partial}{\partial \epsilon}\overline{\mu}(M(\hat{\theta}, \delta(\epsilon), f(\phi(\epsilon))))\Big|_{\epsilon=0} \tag{4.44}$$

We again make the assumption that the compensator remains fixed as the parameters change. This is written as

$$\frac{\partial}{\partial \epsilon}\overline{\mu}(M(\hat{\theta}, \delta(\epsilon), f(\phi(\epsilon))))\Big|_{\epsilon=0} \approx \frac{\partial}{\partial \epsilon}\overline{\mu}(M(\hat{\theta}, \delta(\epsilon), C))\Big|_{\epsilon=0} \tag{4.45}$$

The benefit of this approximation is that because the nominal model remains fixed, lemma 2.2 guarantees that $\overline{\mu}$ will decrease as the uncertainty decreases.

In summary, since in general we are unable to analytically determine the sensitivities of the control design methodology, we need to make an approximation of the μ–sensitivities in which we are interested. By making an assumption on the control

design methodology, we can solve a μ–sensitivity problem where the nominal model does not change. We then approximate this by assuming the compensator does not change as well. The resulting μ–sensitivities will always indicate that $\overline{\mu}$ decreases, which is the desired property. It is interesting to note that this sensitivity tells us which of the parameters is most adversely affecting the current closed loop system. Once the uncertainty does decrease, we can always redesign the compensator to improve $\overline{\mu}$ even more than that indicated by the $\overline{\mu}$ sensitivities.

4.3 Effect of Inputs on Uncertainties

4.3.1 Philosophy of Approach

Based on the previous discussion, we have chosen the parameters α_i in equation (4.3) to be the size of the uncertainty intervals δ_i. In this section, we will outline a computationally efficient method to calculate the value of

$$\mathcal{E}_n \{\Delta\delta_i \mid u\} \tag{4.46}$$

Equation (4.46) is referred to as the effect of input u on δ_i. For the moment, let us assume that we know the true plant and the actual value of the noise which will occur when we apply the next input. In this case, we know the output of the system y_j when we apply the input u_j. Given the pair (u_j, y_j), we could add this information to the data we already have, and recalculate the value of the parameter δ_i. We would need to do this for every input in our set U^k. This could be computationally intensive due to the number of optimization problems that would need to be solved. Instead, we will describe an approximation which we can use to rank the inputs when we know the output. This will be discussed in section 4.3.2.

To determine the expected change in δ_j when we don't know the actual value of the noise which will occur, we could determine the change from every possible value of the noise, and take the average. This is also computationally intensive, and an approximation needs to be made. This is the subject of section 4.3.3.

4.3.2 Calculation with Noise Known

We will first examine the case of how to determine the change in the value of δ_i when we are given a particular (u_j, y_j) pair. If the identification used a recursive framework, it would be easy to find the effects of the additional data point; we would simply use the recursive equations to update the parameter bounds. However, the set membership algorithm of chapter 3 is not a recursive procedure. Instead, we would need to redo the entire optimization procedure for each choice of u_j. As an alternative, we will use the following approximation.

To determine the amount the bounds on an arbitrary parameter θ_i change, we recall that the bound is the result of a convex optimization. Let us assume that we are concerned with the upper bound to θ_i. In determining the upper bound (before adding a new input), we solved the optimization problem for a parameter vector θ^* for which

$$\overline{\theta}_i = \theta_i^* \tag{4.47}$$

To estimate the improvement in $\overline{\theta}_i$, we examine how the optimization problem changes with the constraint added by the new data point. With this new constraint, the optimization problem will yield a new parameter vector $\tilde{\theta}$, where the new upper bound $\tilde{\theta}_i$ is achieved. We will assume that except for the i^{th} component, this new optimum point is the same as θ^*. That is, we assume

$$\tilde{\theta}_k = \theta_k^* \qquad k \neq i \tag{4.48}$$

This is pictured in figure 4-1 where we estimate the improvement in both the upper and lower bounds from a new data point. In the figure, we represent the ellipsoidal constraint (3.26) for each data point as two parallel hyperplanes (degenerate ellipsoids). The solid lines represent the previous constraints, giving rise to the parameter intervals depicted by the rectangle. The constraints due to the new input–output pair are represented by the dotted lines. Our approximation says that the new optimum has the same value in the vertical direction (for this example) as the old optimum.

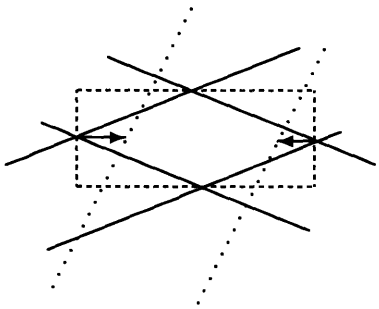

Figure 4-1: Estimate of improvement in parameter bounds due to a new data point (dotted lines). The arrows indicate the estimated improvement.

It is clear from the figure that our estimate of the improvement in the parameter bounds can be optimistic. However, it does give a method to rank each of our choices for the input in a computationally efficient manner. The computation requires solving a quadratic equation, and involves finding the points where a p dimensional ellipsoid intersect a $p-1$ dimensional hyperplane. To see this, we write the new constraint in an ellipsoidal form as in equation (3.27).

$$x^T Q x \leq C^2 \tag{4.49}$$

Let us assume, without loss in generality, that we are interested in the bounds on the first component of the vector x. We partition the matrices as follows.

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix} \tag{4.50}$$

Here, x_1 and Q_{11} are real scalars, and $x_2 \in \mathcal{R}^{p-1}$, $Q_{12} \in \mathcal{R}^{1\times(p-1)}$, and $Q_{22} \in \mathcal{R}^{(p-1)\times(p-1)}$. The vector x_2 is fixed at the value from the previous optimum. We then solve for x_1 such that

$$\begin{bmatrix} x_1 & x_2^T \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = C^2 \tag{4.51}$$

This occurs where

$$x_1^2 Q_{11} + 2x_1 Q_{12} x_2 + x_2^T Q_{22} x_2 = C^2 \tag{4.52}$$

If (4.52) has two solutions, say x_{11} and x_{12} with $x_{11} \leq x_{12}$, the estimate of the new upper bound is the minimum of x_{12} and the previous upper bound. Similarly, the estimate of the new lower bound is the maximum of x_{11} and the previous lower bound.

Once we have done this approximation, we can solve the convex optimization problem for only those inputs which have the best improvements. Our approximation is used as a coarse ranking for all of the possible inputs, and we only need to solve a small number of optimization problems.

4.3.3 Certainty Equivalence

We so far have studied how to compute improvements in parameter bounds given the exact output of the system for any given input. Due to the noise, however, we need to determine the expected value of the improvement. This requires a good estimate of the probability distribution of the noise. To determine such an estimate would be computationally intensive, and thus should be avoided.

As an alternative, we make the following simplifying assumptions. We will assume that the average improvement in the bounds is equal to the improvement when our nominal model is the true system, and the system is noise free. This will be referred to as certainty equivalence. To write this mathematically, let us denote the noise free output of our nominal system as $\hat{y}$, and the change in the j^{th} uncertainty interval if the output of the system was $\hat{y}$ as $\Delta\delta_j(\hat{y})$. The assumption is then written as

$$\mathcal{E}_n \{\Delta\delta_j \mid u\} = \Delta\delta_j(\hat{y}) \tag{4.53}$$

The advantage of such an approach is that it is computationally feasible. We are not required to do any Monte Carlo averaging, which reduces the number of optimization problems we must solve. It also requires little knowledge about the statistics of the noise.

For this to be a reasonable assumption, however, the noise must have certain properties. For instance, the deterministic error n_d must be small, so that the output $\hat{y}$ is close to the actual output of the system. Also, for $\Delta\delta_j(\hat{y})$ to estimate the average change of $\Delta\delta_j$, the probability density function of n_s must be small away from $n_s = 0$.

Unfortunately, it is not possible to verify these properties without estimating $p_{n_s}(n)$ and n_d. The approach here, however, is to choose an input without using this information. In the next chapter, when we explore some of the convergence issues, we will describe in more detail conditions that ensure our approximation appropriately ranks inputs, even when (4.53) is not valid. We will also discuss the limitations of such an approach.

4.4 Summary

In this chapter, we have introduced new tools for experiment design. The philosophy is to choose the input, from a finite set, which we would expect on average to best improve our measure of robust performance. To do so, we have performed a sensitivity analysis on the closed loop system to see the effects of additional inputs.

There were two steps in this analysis. First, we introduced the μ–sensitivites. This problem has been examined in the literature in the past. Here, we solved the problem when our performance is the standard upper bound to the structured singular value. After computing this upper bound, the only additional computation to calculate the μ–sensitivities is an eigenvalue decomposition.

We then examined how to determine the effects of the inputs on the parameter ranges. The philosophy here was more brute force in that we simulate the system with each input, and compared the change in parameter intervals. Using an approximation to the identification algorithm and a certainty equivalence assumption, a computationally efficient method was determined.

Here, we have stressed computational issues. In the next chapter, we will examine the convergence properties of the iterative identification, control design, and experiment design algorithm.

Chapter 5

Iterative Method

5.1 Overview

In this chapter, we will combine the results of the previous two chapters to create the iterative algorithm. The basic steps in this algorithm are identification of the model and uncertainty, robust control design, and choosing a new input to apply to the system. So far, little has been said of the control design. The philosophy in this work is to not advocate any particular methodology; instead, we provide the flexibility to choose any methodology which may be appropriate for a specific system. For instance, we often may want to sacrifice some performance in order to reduce the computation in either the control design or implementation.

In this chapter, we will examine the convergence properties of our algorithm. In the process, we will describe some necessary properties of the control design. We will also describe in more detail assumptions on the noise, and their implications for the input design.

5.2 Outline of the Iterative Scheme

In this section, we will summarize the overall procedure for identification, control design, and experiment design. The notation used here will be that introduced in chapter 2.

We begin by assuming we have a fixed model structure. The first step is thus the identification of the uncertain parameters in the system. An identification algorithm was developed in chapter 3. It should be noted that no attempt is made to estimate the unmodelled dynamics; it is assumed that this is specified a priori, and is fixed.

A compensator is then designed based upon the current model. We have not limited ourselves to a particular methodology, so that any desired methodology can be used. When we examine the convergence properties, however, we will describe a set of properties on the control design which are sufficient to guarantee convergence.

If we have not yet achieved the desired performance, we then perform a new experiment. We choose a finite set of possible inputs. Each input is a sinusoid at a fixed frequency. Based upon a sensitivity analysis, we choose the input which we expect will improve our resulting performance as much as possible. We can then repeat these steps beginning with a revised identification.

We have described the following algorithm. A block diagram of the algorithm is shown in figure 5-1.

Algorithm 5.1 (Iterative ID, Control, and Experiment Design)

1. *Determine the inputs for the initial experiment, and measure the data* y^0. *Let* $k = 1$, *and let* ϕ^0 *denote our a priori knowledge of the parameters.*

2. *Determine the model parameters* $\phi^k = h(\phi^{k-1}, y^{k-1})$.

3. *Design the compensator* $C^k = f(\phi^k)$.

4. *If* $\overline{\mu}(M(\phi^k, C^k)) < 1$, *then stop.*

5. *Choose a finite set* U^k *of possible inputs for the next experiment.*

6. *Determine the* μ*–sensitivities*

$$\frac{\partial}{\partial \delta_i} \overline{\mu}(M(\hat{\theta}^k, \delta^k, C^k)) \tag{5.1}$$

and the effects of the inputs

$$\mathcal{E}_n \{\Delta \delta_i \mid u\} \tag{5.2}$$

for each $u \in U^k$.

7. *Apply the input* u *which maximizes*

$$\sum_{i=1}^{N} \frac{\partial}{\partial \delta_i} \overline{\mu}(M(\hat{\theta}^k, \delta^k, C^k)) \mathcal{E}_n \{\Delta \delta_i \mid u\} \tag{5.3}$$

8. *Set* $k = k + 1$, *and go to step 2.*

5.3 Convergence of $\overline{\mu}$

The first thing we would like to guarantee about this algorithm is that the performance measure $\overline{\mu}$ is monotonically nonincreasing. There are two steps to this procedure. First, we must guarantee that we are not losing information in the identification step. Specifically, we must guarantee that the upper and lower bounds are not getting worse, i.e.

$$\underline{\theta}^k \leq \underline{\theta}^{k+1} \tag{5.4}$$

$$\overline{\theta}^{k+1} \leq \overline{\theta}^k \tag{5.5}$$

Note that (5.4) and (5.5) are convex constraints. Therefore it is straightforward to include these constraints in the identification optimization problem. We will assume that this has been done, so that (5.4) and (5.5) are automatically satisfied. In fact, the use of this a priori knowledge motivated the representation of the identification scheme as a function of the previous bound.

Secondly, we need to guarantee that as the uncertainty decreases, the robust performance measure will be nonincreasing (where we assume that a lower value of the performance measure indicates better robust performance). We will state this as an assumption on the control methodology.

Assumption 5.1 *Given (5.4) and (5.5), then*

$$\overline{\mu}(M(\phi^{k+1}, f(\phi^{k+1}))) \leq \overline{\mu}(M(\phi^k, f(\phi^k))) \tag{5.6}$$

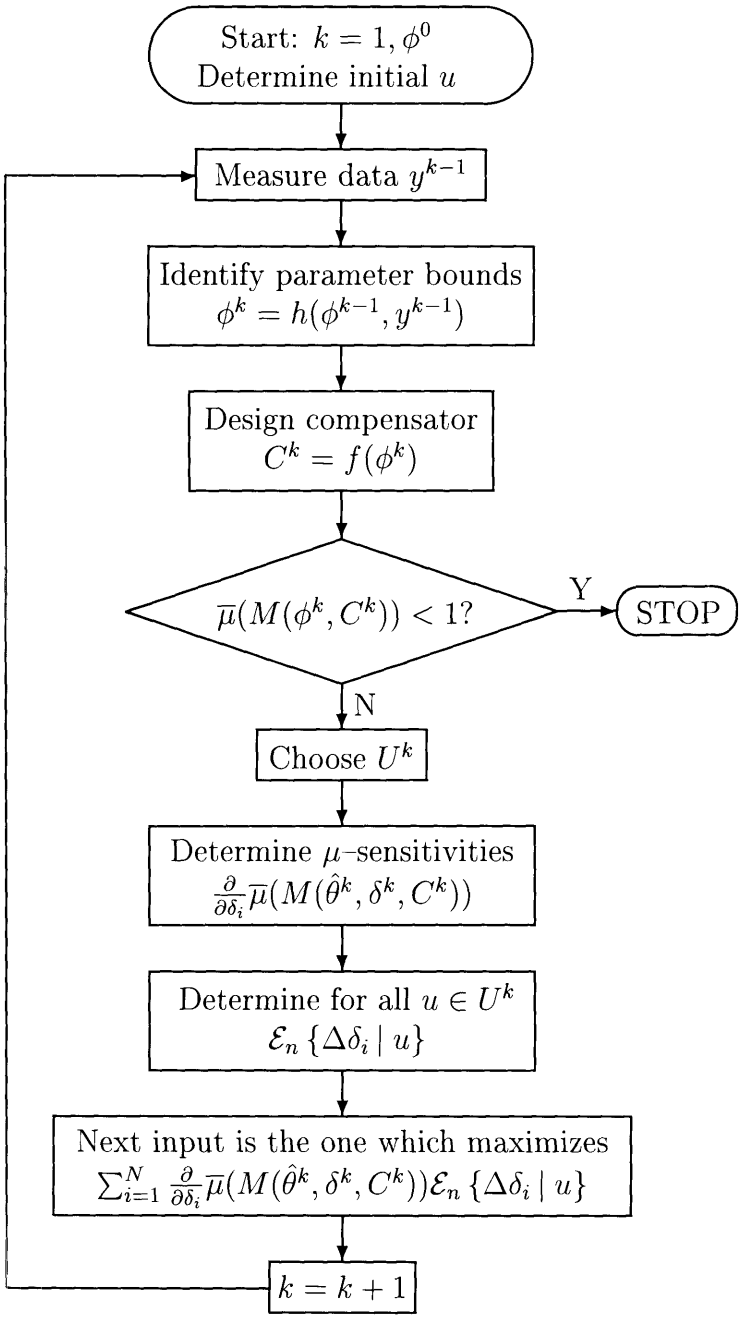

Figure 5-1: Block diagram of the iterative algorithm.

This assumption need not hold for an arbitrary control methodology. However, from the discussion in section 2.2 we know there are special cases when it will hold. Specifically, if the uncertainty decreases in such a way that the nominal model remains fixed, and if the control methodology produces the same compensator for a fixed nominal model, then $\overline{\mu}$ will not increase. That is,

$$\overline{\mu}(M(\hat{\theta}^k, \delta^{k+1}, f(\phi^k))) \leq \overline{\mu}(M(\hat{\theta}^k, \delta^k, f(\phi^k))) \text{ for } \delta^{k+1} \leq \delta^k \tag{5.7}$$

The difficulty arises when the midpoint of the parameter intervals shift. In that case there are many control methodologies for which assumption 5.1 does not hold.

Since the philosophy in this work is to allow as many control methodologies as possible, the following procedure can be used. After determining the new bounds on the uncertainty intervals, we will design our compensator. If $\overline{\mu}$ increases, we will increase the uncertainty intervals in such a way that the midpoint of each interval is the same as on the previous iteration. Although we may increase the uncertainty in the system from the control design perspective, we are still guaranteed that (5.4) and (5.5) hold. The increase is accomplished as follows:

$$\hat{\theta} = \hat{\theta}^k \tag{5.8}$$

$$\delta = 2\max\left\{ \overline{\theta}^{k+1} - \hat{\theta},\ \ \hat{\theta} - \underline{\theta}^{k+1} \right\} \tag{5.9}$$

$$\overline{\theta} = \hat{\theta} + \frac{\delta}{2} \qquad \underline{\theta} = \hat{\theta} - \frac{\delta}{2} \tag{5.10}$$

$$\tilde{\phi} = \begin{bmatrix} \underline{\theta} \\ \overline{\theta} \end{bmatrix} \tag{5.11}$$

This is visualized in figure 5-2. The bounds decreased from iteration $k-1$ to iteration k. However, the midpoint moved as well. To recover the previous midpoint, the uncertainty is increased. After redesigning the compensator, if $\overline{\mu}$ still has increased, we will use the compensator from the previous iteration with this increased uncertainty.

The control design procedure is thus modified as follows. Let $F(\phi)$ be the desired

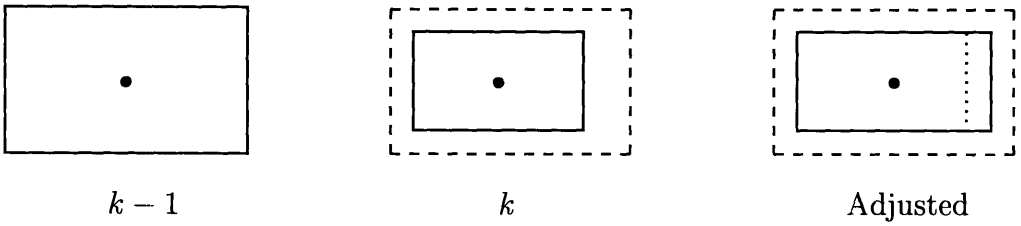

Figure 5-2: Expanding the uncertainty region to guarantee the control design will result in a nonincreasing value of $\overline{\mu}$.

control methodology. Then $f(\phi^{k+1})$ is defined by

$$f(\phi^{k+1}) = \begin{cases} F(\phi^{k+1}) & \text{if } \overline{\mu}(M(\phi^{k+1}, F(\phi^{k+1}))) \leq \overline{\mu}(M(\phi^k, C^k)) \\ F(\tilde{\phi}) & \text{if } \overline{\mu}(M(\phi^{k+1}, F(\phi^{k+1}))) > \overline{\mu}(M(\phi^k, C^k)) \\ & \text{and } \overline{\mu}(M(\tilde{\phi}, F(\tilde{\phi}))) \leq \overline{\mu}(M(\phi^k, C^k)) \\ C^k & \text{otherwise} \end{cases} \tag{5.12}$$

With this procedure, assumption 5.1 will be satisfied, regardless of what our choice for the control methodology F may be.

Thus, $\overline{\mu}$ is nonincreasing. Since it is bounded from below by 0, it must converge as the algorithm proceeds. This result does not, however, indicate whether or not we will converge to where $\overline{\mu} < 1$. This will be examined in the next section.

5.4 Asymptotic Performance Bound

Our closed loop guaranteed performance is necessarily a function of the control methodology. One question we would like to ask is whether the iterative methodology will achieve the best possible guaranteed performance that our particular control methodology is capable of achieving. In other words, we would like to understand the conditions under which our analysis correctly indicates which inputs we should apply to improve our performance.

Our specific goal is the following. Assume that there exist a sequence of identification experiments such that the uncertainty intervals would asymptotically become

small enough so that $\overline{\mu}$ becomes less than 1. We would like to guarantee that, through our sensitivity analysis, we will choose inputs so that we eventually achieve $\overline{\mu} < 1$.

Achieving this goal depends on the interaction of the sensitivity analysis, the experimentation process, and the characteristics of the noise. We will now examine these relationships in more detail.

There are two issues we need to address. First, we need to ensure that the sensitivity analysis will not indicate a high sensitivity for an input which has little chance of improving $\overline{\mu}$. In other words, we need to guarantee that we are not choosing inputs that will not improve the appropriate parameter bounds when one exists which will improve them. By appropriate bounds, we mean parameter intervals which are limiting our performance.

The other issue is whether the algorithm can terminate prematurely. Termination occurs when all the sensitivities are zero. We need to guarantee that when we terminate without achieving our goal, then no choice of inputs could improve our guaranteed performance.

5.4.1 Guaranteed Improvement

We will first determine conditions that guarantee we will not choose incorrect inputs. From section 5.3, we know that $\overline{\mu}$ will never increase. Let us assume that for the i^{th} parameter, the μ–sensitivity equals ϵ_1 for some $\epsilon_1 > 0$. This implies that as δ_i decreases, $\overline{\mu}$ will decrease at a rate of ϵ_1 (to first order) if we use the same compensator. When we update the compensator, we will be doing at least as well. So clearly a positive μ–sensitivity is an indication that decreasing the uncertainty interval will decrease $\overline{\mu}$.

Therefore, we can choose the incorrect input only when we determine that an input will decrease our uncertainty bounds, when this is in fact not possible. This is clearly a function of how we analyze the effects of the inputs on the uncertainty. To proceed further, we need to impose more structure on the noise in the system.

Recall that from chapter 4 that the noise is written as

$$n(j\omega) = n_d(j\omega) + n_s(j\omega) \tag{5.13}$$

where n_d is a fixed but unknown deterministic error which includes the errors due to model mismatch (since we assume our nominal model $\hat{\theta}$ is the true system), and n_s is zero mean stochastic noise with probability density function $p_{n_s}(n)$. We will determine characteristics of the noise which guarantee that the method of approximating $\mathcal{E}_n \{\Delta\delta_i \mid u\}$ is nonzero only if applying u will cause δ_i to decrease with nonzero probability.

There are two parts to this argument. First, we must show that if the output of the experiment is close to what we expect, then the decrease in δ_i is also close to what we expect. This is a continuity argument. We will then determine conditions on the noise which guarantee we are close to the expected output with nonzero probability.

Let us first address the continuity argument. Let $\hat{y}$ be our expected output of the system when we assume our nominal model is the true system and the system is noise free. Let $\Delta\delta_j(y)$ be the change in δ_j if the actual output was given by y. Then we will make the following assumption.

Assumption 5.2 *Given $\hat{y}$, for all $\epsilon > 0$ there exists an open neighborhood Y around $\hat{y}$ such that*

$$|\Delta\delta_i(y) - \Delta\delta_i(\hat{y})| \leq \epsilon \quad \forall y \in Y \tag{5.14}$$

Although this is stated as an assumption, it is clear that it holds true for the identification algorithm in chapter 3. The convex constraint added by the additional data point is continuous in the value of the output, and therefore so is the result of the convex optimization.

We will now determine conditions to guarantee the output is in the open set Y with nonzero probability. Essentially we need to guarantee that the deterministic error is not too large relative to the stochastic noise. However, we would like to do so without making strong assumptions on $p_{n_s}(n)$. Instead, we will define the function

$H(\epsilon)$ such that

$$p_{n_s}(n) > \epsilon \quad \forall n \quad \text{such that} \quad |n| < H(\epsilon) \tag{5.15}$$

This implies that there is a region given by $H(\epsilon)$ such that the probability density function is greater than ϵ in this region. With this notation, the necessary assumption is the following.

Assumption 5.3 *$\exists \epsilon > 0$ such that $|n_d| < H(\epsilon)$*

This assumption describes the allowable size of the deterministic error. It must be small enough such that there is a nonzero probability that the noisy output is inside a small open set around $\hat{y}$, the noise free output.

It is important to understand the implication of this. Under the assumptions given, an input we choose will cause the uncertainty intervals to decrease (in such a way so that $\overline{\mu}$ will decrease) with nonzero probability. If our compensator and performance do not change, we can try the same input again. With probability one, we will eventually get the uncertainty bounds to decrease.

5.4.2 Preventing Premature Termination

There are two issues that need to be addressed in examining whether the algorithm can terminate prematurely. First, we are limiting our choice of inputs to a finite set, when there may be an infinite number of inputs we would like to choose from. For example, we may be able to apply any unit amplitude sinusoid between f_1 and f_2 hertz. We need to choose the sets in such a way that every one of these sinusoids can be chosen. We also need to show that the sensitivity analysis is not ignoring useful inputs by thinking that they will not help decrease $\overline{\mu}$ when in fact they can. These issues are addressed in this section.

We begin with a possible uncountable set of inputs from which we would like to choose the next input. Let us call this set U^*. Instead of choosing from U^*, we create a finite set, U^k, from which we choose an input at time k. We will require each element of U^* to appear in $\{U^1, U^2, ...\}$ infinitely often. This allows us to guarantee

that given an arbitrarily large iteration index k, and any useful input $u \in U^*$, then $u \in U^{k_1}$ for some $k_1 > k$. Since the input–output map of the plant is continuous (i.e. the transfer function is a continuous function of frequency), this can be weakened to the following. We will choose the sets U^k such that

$$\lim_{k \to \infty} U^k \text{ is a dense subset of } U^*$$

Instead of choosing from U^* at time k, we choose from the finite set U^k.

We now examine what happens if for all inputs u we determine that

$$\mathcal{E}_n \{\Delta\overline{\mu} \mid u\} = 0 \tag{5.16}$$

This will happen when for all j

$$\frac{\partial \overline{\mu}}{\partial \delta_j} = 0 \quad \text{or} \quad \mathcal{E}_n \{\Delta\delta_j \mid u\} = 0 \tag{5.17}$$

The first condition has implications for the control design, while the second has implications on our analysis of the noise.

To prevent the μ–sensitivities from becoming zero we will assume that there is a value $\epsilon > 0$ such that at least one of the μ– sensitivities has a value greater than ϵ. Let us examine what this assumption indicates about the control methodology. If all the μ–sensitivities are zero, then decreasing any of the uncertainties by a small amount will (to first order) have no effect on the resulting robust performance. This implies that the robust performance is not being limited by any of the uncertain parameters. Instead, it is being limited by unstructured uncertainty, or the robust performance specification. We wish to avoid this situation.

The control design must therefore make use of the uncertainty intervals to the extent that they are limiting performance. This means the compensator must be designed in some intelligent fashion. We are not considering in this work designs which try to get performance by only putting energy where there is no parametric uncertainty.

Let us now examine what happens if our approximation says that $\mathcal{E}_n \{\Delta\delta_j \mid u\} = 0$ for all inputs u. If we were, in fact, taking the expected value over the noise, then this could only happen when the probability that δ_j will decrease with the next input is 0. With the approximation presented in section 4.3.3 however, it is not clear that this will always hold.

It is possible that although the noise free system will not produce an output which will improve the bounds, there is a high probability that the noise will be such that an improvement will occur. Such a situation is shown in figure 5-3. In this figure, it does not appear as though the new data point will improve the parameter bounds on the noise free system. However, if the data was shifted by a small amount (due to noise), then a decrease would, in fact, occur.

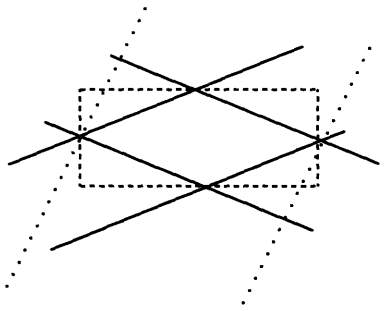

Figure 5-3: A new data point which will not improve the parameter bounds, while small deviations due to noise could cause improvement. The new data point is represented by the dotted lines. Compare to figure 4-1.

In the case when our approximation says that none of the bounds would improve for any input, we could then look at which inputs have a greater probability of improving the bounds. Instead of ranking the inputs by approximated improvements in parameter bounds, we are now ranking based on an estimate of how close the nominal ellipsoid for each input is to causing a parameter bound improvement.

The possibility of this occurrence is a drawback to the approximation presented. At this point, it may become necessary to estimate the statistics of the noise from the measured data. In essence, we would be seeking to estimate the probability density function of $n_s(j\omega)$. In this way, we could improve our estimation of the expected

improvement. The drawback is the additional computation required to determine the expected decrease in the uncertainty intervals.

5.5 Summary

In this chapter, we have combined the results of the previous chapters to create the iterative algorithm for identification, control design, and experiment design. The emphasis has been on the convergence properties of the algorithm. The first property was to show that $\overline{\mu}$ would be nonincreasing. This required an assumption on the control methodology. We were able to show how to adapt any control methodology to satisfy this assumption.

We then discussed further properties which would be needed to guarantee we asymptotically achieve the desired robust performance from our control methodology. These properties are not easily verifiable a priori. We need the deterministic error to be small relative to the variation in the stochastic noise. Also, the control design must make use of the parametric uncertainty to the extent that it is limiting performance. Since these are properties which make engineering sense, we could describe the input design process as being an intelligent heuristic. However, we can not a priori guarantee convergence to the best possible performance of the specified control methodology. We can proceed with the algorithm until the improvement in the parameter bounds becomes small. If we wish to continue choosing inputs, we may need to use the data to estimate the statistics of the noise.

Chapter 6

Examples

In this chapter, we will provide several examples of the algorithms presented in this work. The first examples demonstrate the ability of the identification algorithm to determine accurate parameter estimates from only a few data points. We also demonstrate the need to use physical insight in determining the correct parameterization of the structure.

Several examples of the input design algorithm are then presented. It is shown how the algorithm makes full use of the available information to determine the most appropriate input to apply. Several different bounds on the noise are used, and insights into the choice of inputs are made.

6.1 Bernoulli-Euler Beam

The examples in this chapter are based upon a Bernoulli-Euler model of a cantilevered beam [6], shown in figure 6-1. A force is applied at the tip of the beam, and we measure displacement at the tip.

We have truncated the infinite order dynamics to the first 16 states. Thus, the "true" plant will consist of the first eight modes. The length of the beam has been normalized so that the first mode is at 1 rad/sec. We will identify the parameters of the first three modes, and consider the other modes as unmodelled dynamics. The noise free dynamics are shown in figure 6-2.

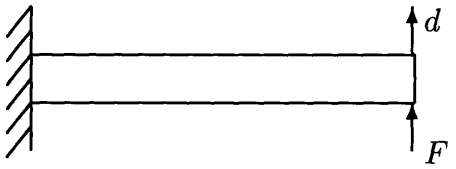

Figure 6-1: Bernoulli-Euler Beam.

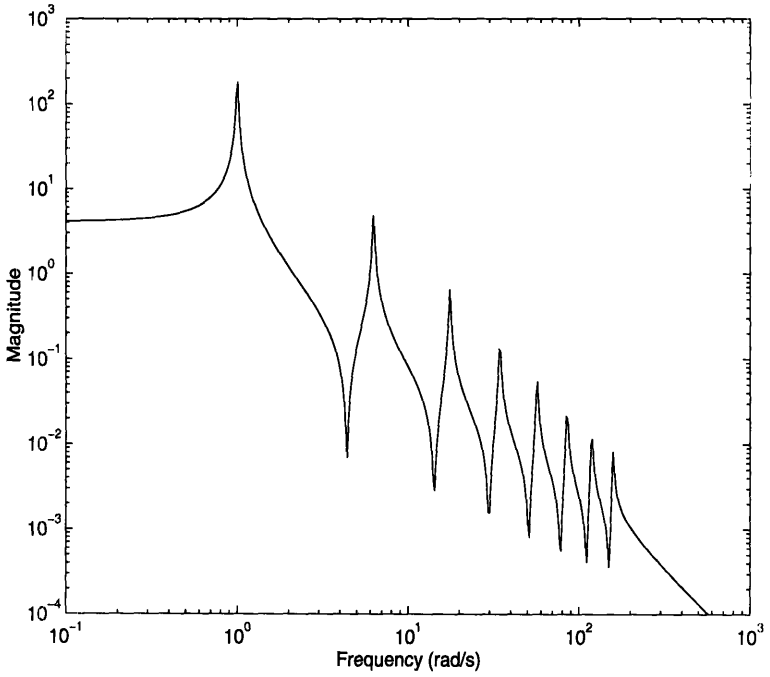

Figure 6-2: Noise free dynamics of the Bernoulli-Euler Beam.

The a priori information needed for the identification algorithm includes a bound on the noise, an upper bound on the damping ratio for each mode, and upper and lower bounds on the natural frequencies of each mode. The a priori bounds on the natural frequencies are shown in figure 6-3. Also shown in the figure is an additive bound on the unmodelled dynamics. The noise bound used must account for both the additive noise in the system, and the unmodelled dynamics. The noise bound for the identification algorithm will therefore be the sum of this bound on unmodelled dynamics plus the bound on the additive noise. The bound on the unmodelled dynamics can be represented as

$$W_1(s) = \frac{.182(s^2 + 15.2s + 361)^2}{(s^2 + 26.4s + 1089)^2} \tag{6.1}$$

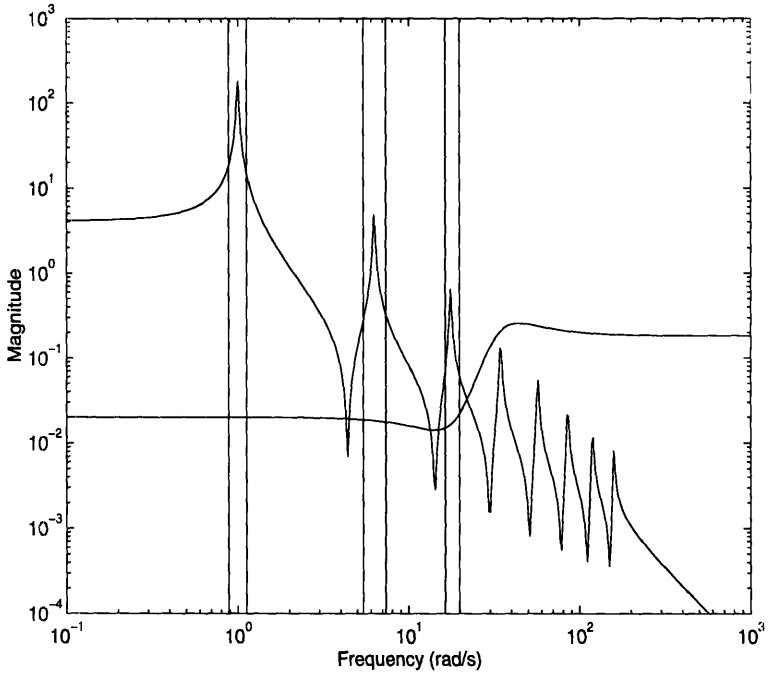

Figure 6-3: A priori knowledge for use with the identification algorithm. The vertical lines are the a priori bounds on the natural frequencies. Also shown is the bound on unmodelled dynamics $|W_1(j\omega)|$.

For an a priori bound on the damping ratio, we will use .5, i.e. $0 \leq \zeta \leq .5$. Since the true value of the damping ratio is .01 for each mode, we see that the initial guess is very conservative. The a priori bounds on the natural frequency of the first mode

are $.884 \leq \omega_n \leq 1.125$; the a priori bounds for the second mode are $5.41 \leq \omega_n \leq 7.34$; and the a priori bounds for the third mode are $16.35 \leq \omega_n \leq 19.81$.

In the following sections, a number of examples will be presented with this system. For each example, the bound on the additive noise will be specified, and will be used with the a priori information presented in this section.

6.2 Identification Algorithm

In this section, we will provide two examples of the identification algorithm. In the first example, we will use the parameterization discussed in section 2.1. For the second example, we will use physical insight to reduce the number of parameters in the model, and consequentially improve our parameter estimates.

For both of these examples, the noise will be bounded in magnitude by the bound shown in figure 6-4. The overall noise bound is therefore the sum of this bound, and the bound on the unmodelled dynamics $W_1(s)$. This noise bound was chosen so that the signal to noise ratio was on the same order for all of the modes. It was derived by taking the noise free model of the Bernoulli-Euler beam, and setting the damping on both the poles and zeros to 1. Of course, this knowledge is not used by the identification algorithm.

Since the noise sequence must be complex, it was generated uniformly in both magnitude and phase, with the bound on the magnitude shown in figure 6-4. An example of the noisy data is shown in figure 6-5. It should be noted that very similar results were obtained when the identification algorithm was given the same bound on the noise, but the data was actually noise free. Thus the bound used is very significant, and we should strive to use as tight a bound as possible.

During the identification algorithm, the estimate for each mode was chosen as the system with the parameter values at the midpoints of the interval ranges. Initially, the estimates where chosen to be 0. This happens to be a particularly bad estimate, but the error due to this estimate was quickly reduced.

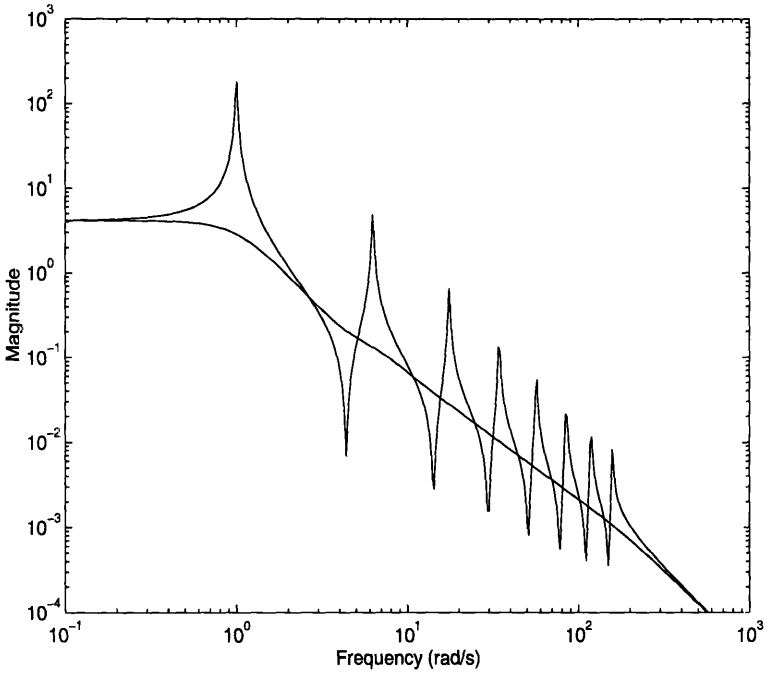

Figure 6-4: Noise free system, and bound on the additive noise as a function of frequency.

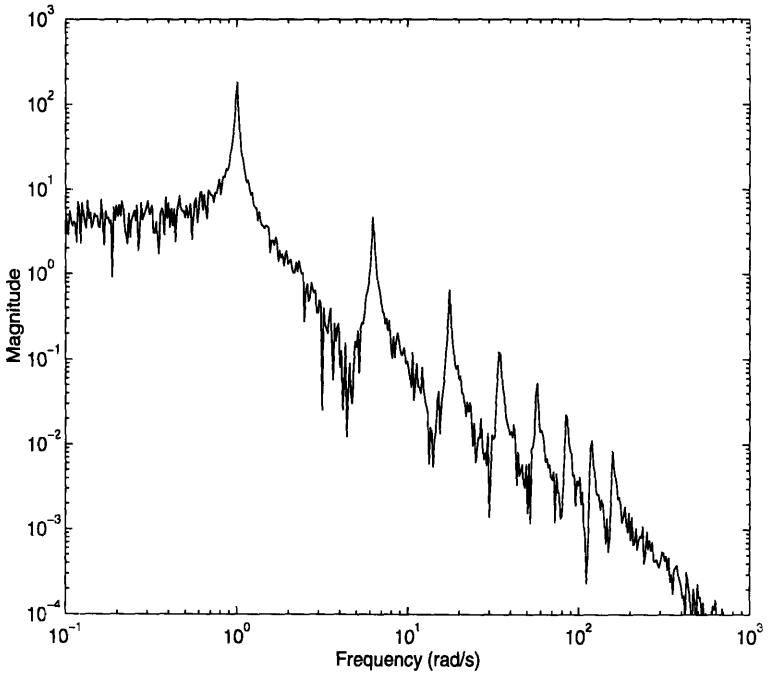

Figure 6-5: Noisy data for the first example.

6.2.1 Identification of the System

To identify this system, we use a three mode parameterization as in section 2.1. Thus, there are 12 uncertain parameters. We run algorithm 3.1, and on step 2 of this algorithm we apply the steps of algorithm 3.2 only once (i.e. no iteration). The final parameter variations occurred after 8 iterations of algorithm 3.1.

There is clearly a tradeoff between the number of data points used in the identification algorithm, and the accuracy of the parameter intervals. The more data points we use, the more information we have about each of the parameters. However, if we increase the number of data points, we are also increasing the number of constraints used in the convex programming problems. For the examples presented here, we will use only 5 data points for each mode. These data points are chosen as the 5 data points closest to the peak in the frequency response for the mode, where the signal to noise ratio is highest. Thus, we use only 15 data points to identify the system.

We would like to guarantee the ratio of conservatism γ, defined in section 3.2.3, is less than 1.1. Before running the algorithm, however, we are unable to partition the parameter space because the lower bound on the damping ratio is initially equal to 0. Instead, we run the algorithm without partitioning. After one iteration, we are then able to partition the parameter space. However, as shown in table 6-1, the number of partitions necessary to achieve $\gamma < 1.1$ is unreasonably large. Instead, we do not partition, and perform the steps in the iterative algorithm again. We continue in this fashion for 7 iterations. After the seventh iteration, the number of partitions necessary to achieve $\gamma < 1.1$ is reasonable. We can now partition the parameter space as appropriate for the last iteration of the algorithm.

The resulting parameter intervals are shown in table 6-2. Notice that even though we have only used 5 data points for each mode, the parameter intervals are small. It is also interesting to note that we did not fully use the physical information available to us. We are measuring displacement, and based upon our model structure, this implies that $b_{1i} = 0$ for each of the modes. We did not use this information, and for the first mode, the parameter interval is significant. The result is that there is a large

Iteration completed	Partitions we would need for Mode 1	Partitions we would need for Mode 2	Partitions we would need for Mode 3
1	3432	5744	4332
2	8	3864	5408
3	2	78	4044
4	2	18	296
5	2	10	102
6	2	8	75
7	2	8	65

Table 6-1: The number of partitions needed to achieve $\gamma < 1.1$.

Parameter	Lower bound	Upper bound	True Value
a_{11}	1.9222×10^{-2}	2.0931×10^{-2}	2.0000×10^{-2}
a_{01}	9.9911×10^{-1}	1.0008	1.0000
b_{11}	-1.3869×10^{-1}	1.2570×10^{-1}	0
b_{01}	3.8820	4.1493	4.0000
a_{12}	1.1278×10^{-1}	1.3975×10^{-1}	1.2534×10^{-1}
a_{02}	3.9200×10^{1}	3.9369×10^{1}	3.9274×10^{1}
b_{12}	-5.6459×10^{-2}	4.7304×10^{-2}	0
b_{02}	3.6984	4.3906	4.0000
a_{13}	2.6480×10^{-1}	4.7965×10^{-1}	3.5095×10^{-1}
a_{03}	3.0599×10^{2}	3.0970×10^{2}	3.0791×10^{2}
b_{13}	-4.4073×10^{-2}	5.8050×10^{-2}	0
b_{03}	3.2468	5.0450	4.0000

Table 6-2: Final parameter intervals.

model mismatch, which causes the other modes to not be identified as accurately.

To see exactly the effects of ignoring the physical knowledge available, let us examine the conservatism in our identification in more detail. Recall that there are two sources of conservatism. There is the conservatism due to the embedding, which we have already specified is no more than 10%. There is also conservatism due to model mismatch, when we reduce our problem to a one mode problem by subtracting estimates of the other modes. In table 6-3, this conservatism is listed for each mode. The total conservatism is therefore the product of the conservatism from model mismatch and the conservatism from the embedding. The total conservatism listed is therefore the effective increase of the noise bound over the actual noise bound. We see that without using the physical knowledge, the conservatism can be large. We also note that we do not necessarily reduce this conservatism as the algorithm proceeds.

Mode	Conservatism due to model mismatch	Total Conservatism
1	1.005	1.11
2	1.18	1.30
3	1.29	1.42

Table 6-3: The conservatism in each mode due to model mismatch, and the total conservatism.

6.2.2 Using Physical Knowledge

We will now use the fact that we are only measuring displacement, and not velocity. We will adapt the model of the system by setting $b_{1i} = 0$ for each mode.[1] The same procedure for determining the bounds as in the previous example is followed.

The final parameter intervals are shown in table 6-4. We see that by using the correct structure for the system, the uncertainty is decreased. We also see in table

[1]This was actually implemented by choosing a priori bounds for b_{1i} as $\overline{b}_{1i} = \sqrt{10^{-5}}$ and $\underline{b}_{1i} = -\sqrt{10^{-5}}$.

6-5 that the conservatism has dramatically decreased. We can thus conclude that using all the available knowledge in creating the parameterization of the system is important. This physical knowledge will be used for the remainder of the examples.

Parameter	Lower bound	Upper bound	True Value
a_{11}	1.9211×10^{-2}	2.0921×10^{-2}	2.0000×10^{-2}
a_{01}	9.9946×10^{-1}	1.0005	1.0000
b_{01}	3.8806	4.1480	4.0000
a_{12}	1.1428×10^{-1}	1.3777×10^{-1}	1.2534×10^{-1}
a_{02}	3.9241×10^{1}	3.9326×10^{1}	3.9274×10^{1}
b_{02}	3.7375	4.3394	4.0000
a_{13}	2.8057×10^{-1}	4.5110×10^{-1}	3.5095×10^{-1}
a_{03}	3.0731×10^{2}	3.0854×10^{2}	3.0791×10^{2}
b_{03}	3.4042	4.8278	4.0000

Table 6-4: Final parameter intervals, when using physical knowledge.

Mode	Conservatism due to model mismatch	Total Conservatism
1	1.0038	1.104
2	1.0418	1.146
3	1.0392	1.143

Table 6-5: The conservatism in each mode due to model mismatch, and the total conservatism for the case when we use physical knowledge.

6.3 Input Design

We now give several examples of the iterative methodology for input design. We begin by describing the control methodology. Examples are then presented which demonstrate the ability of the algorithm to reduce $\overline{\mu}$ very quickly. Although this input design algorithm is a heuristic, and few convergence properties are guaranteed, these examples demonstrate the potential of this approach to achieve our robust performance goals, while using only a small number of data points. Throughout, we

will use the model described in the previous section, which uses physical insight to reduce the parameter errors.

6.3.1 Control Design

In this section, we will describe the control design methodology. We first need to construct a state space representation of the system. We will then design a compensator based upon this model. The description here will be general in terms of the number of uncertain parameters.

We begin with a standard representation of the nominal model plus parametric uncertainty. We will assume that there are p uncertain parameters, of which the first r are in the A matrix, and the remaining $p-r$ are in the C matrix. These uncertain parameters enter in a linear fashion, so we can write the system as a nominal model plus the uncertainty as follows:

$$\dot{x}(t) = Ax(t) + \sum_{i=1}^{r} q_i l_i n_i^T x(t) + Bu(t) \tag{6.2}$$

$$y_p(t) = Cx(t) + \sum_{i=r+1}^{p} q_i l_i n_i^T x(t) \tag{6.3}$$

$$|q_i| \leq 1 \quad i = 1, ..., p \tag{6.4}$$

Here, q_i represents the uncertainty of the i^{th} parameter, with l_i and n_i representing the structure of how the uncertainty enters the system.

The midpoints of the uncertainty intervals are always chosen for the nominal model. It should be noted that the choices for l_i and n_i are not unique since they can be scaled arbitrarily. In the examples presented, each parameter enters in a rank 1 fashion, so that l_i and n_i are vectors. We will always choose the scaling such that $\|l_i\|_2 = \|n_i\|_2$.

As in [13], we can put all of these uncertainties into larger matrices as follows:

$$E_a = \begin{bmatrix} l_1 & \dots & l_r \end{bmatrix} \qquad E_c = \begin{bmatrix} l_{r+1} & \dots & l_p \end{bmatrix} \tag{6.5}$$

$$F_a = \begin{bmatrix} n_1^T \\ \vdots \\ n_r^T \end{bmatrix} \qquad F_c = \begin{bmatrix} n_{r+1}^T \\ \vdots \\ n_p^T \end{bmatrix} \tag{6.6}$$

We can now describe the system in the following state space form.

$$\dot{x}(t) = Ax(t) + Bu(t) + E_a w_a(t) \tag{6.7}$$

$$y_p(t) = Cx(t) + E_c w_c(t) \tag{6.8}$$

$$z_a(t) = F_a x(t) \tag{6.9}$$

$$z_c(t) = F_c x(t) \tag{6.10}$$

$$w_a(t) = Q_a z_a(t); \qquad Q_a = diag(q_1, ..., q_r) \tag{6.11}$$

$$w_c(t) = Q_c z_c(t); \qquad Q_c = diag(q_{r+1}, ..., q_p) \tag{6.12}$$

We will assume there are unmodelled dynamics in our system, which we will represent as an *additive* uncertainty. This uncertainty will be written as $\Delta_1 W_1(s)$, with W_1 defined in (6.1).

We will also consider our performance as a weighted sensitivity. The robust performance goal is to keep the quantity $|W_2(j\omega)S(j\omega)| \leq 1$, with the weighting function

$$W_2(s) = \frac{.33(s^2 + 4.9s + 12.25)^2}{(s^2 + 3.2s + 4)^2} \tag{6.13}$$

The desired bound on the sensitivity is shown in figure 6-6. The bound is defined as the inverse of the magnitude of W_2 (i.e. we wish to keep the sensitivity less than this bound). Note that we wish to make this guarantee even in the face of uncertainty. It is therefore referred to as a robust performance bound. It will be included in the uncertainty block as the unstructured uncertainty Δ_2 [16].

We will now include the weighting functions in our state space model. Let us

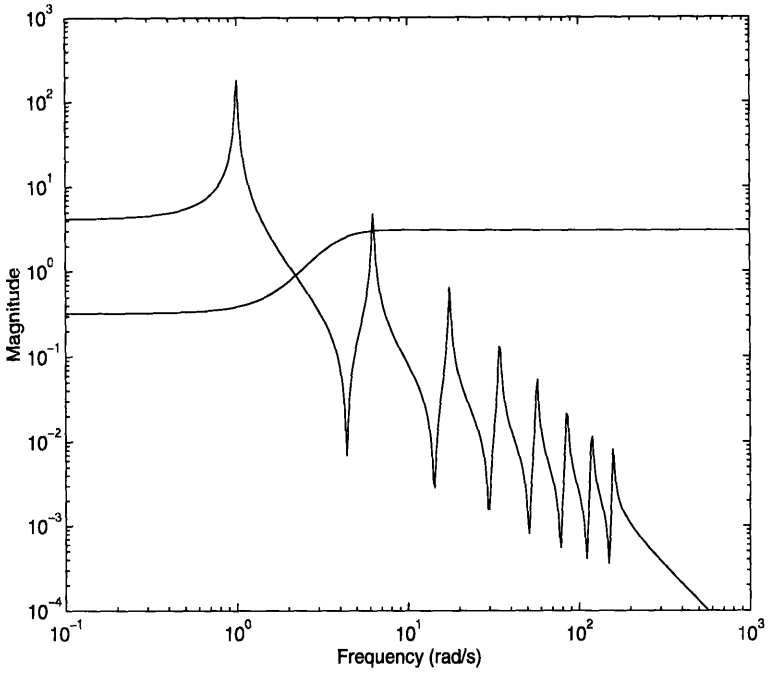

Figure 6-6: Open loop system, and desired bound on the sensitivity. The bound is given by $|W_2(j\omega)|^{-1}$.

assume that the weighting functions have the following state space description.

$$W_1(s) = D_1 + C_1(sI - A_1)^{-1}B_1 \tag{6.14}$$

$$W_2(s) = D_2 + C_2(sI - A_2)^{-1}B_2 \tag{6.15}$$

Augmenting the states of the weighting functions to those of the plant, we have the following state space description.

$$\dot{\tilde{x}}(t) = \tilde{A}\tilde{x}(t) + \tilde{L}w(t) + \tilde{B}u(t) \tag{6.16}$$

$$z(t) = \tilde{C}_1\tilde{x}(t) + \tilde{D}_{11}w(t) + \tilde{D}_{12}u(t) \tag{6.17}$$

$$y(t) = \tilde{C}_2\tilde{x}(t) + \tilde{D}_{21}w(t) \tag{6.18}$$

$$w(t) = \Delta z(t) \tag{6.19}$$

$$\tilde{A} = \begin{bmatrix} A & 0 & 0 \\ 0 & A_1 & 0 \\ B_2C & 0 & A_2 \end{bmatrix} \quad \tilde{L} = \begin{bmatrix} E_a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & B_2E_c & B_2 & B_2 \end{bmatrix} \quad \tilde{B} = \begin{bmatrix} B \\ B_1 \\ 0 \end{bmatrix} \tag{6.20}$$

$$\tilde{C}_1 = \begin{bmatrix} F_a & 0 & 0 \\ F_c & 0 & 0 \\ 0 & C_1 & 0 \\ D_2C & 0 & C_2 \end{bmatrix} \quad \tilde{D}_{11} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & D_2E_c & D_2 & D_2 \end{bmatrix} \quad \tilde{D}_{12} = \begin{bmatrix} 0 \\ 0 \\ D_1 \\ 0 \end{bmatrix} \tag{6.21}$$

$$\tilde{C}_2 = \begin{bmatrix} C & 0 & 0 \end{bmatrix} \quad \tilde{D}_{21} = \begin{bmatrix} 0 & E_c & I & I \end{bmatrix} \tag{6.22}$$

$$\Delta = \text{blockdiag}(Q_a, Q_c, \Delta_1, \Delta_2) \tag{6.23}$$

This state space description now describes the open loop system. A block diagram of the system is shown in figure 6-7. In the figure, $P(s)$ is the open loop system described by (6.16)–(6.18).

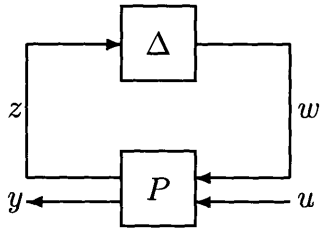

Figure 6-7: The open loop system including the uncertainties.

The control methodology we will use is an $\mathcal{H}_2$ design. The goal of the compensator is to minimize the $\mathcal{H}_2$ norm between w and z, as shown in figure 6-7. It is important to note that the compensator is not actually trying to achieve the specified robust performance. However, this is a design which is both easy to compute, and also results in a compensator that is the same order as the plant. Since we consider the parameter uncertainty as noise entering the system, the compensator is sensitized to this uncertainty. The robust performance analysis indicates that this method produces compensators which do well.

It is important to notice that for the $\mathcal{H}_2$ problem to be well posed, we need $D_2 = 0$. Since this is not the case with our particular weight on performance, we need to modify our weights. A 20 dB/decade rolloff was added to the weighting function $W_2(s)$ by including an extra pole at $s = -10000$. Since this pole is at such a high frequency, it will have very little effect on the system. However, it enables us to use the desired control methodology.

Several other design methodologies were attempted. It was found that an $\mathcal{H}_\infty$ design had similar performance to the $\mathcal{H}_2$ design. This similarity has been noted by other researchers, such as [38]. Also, see [17] for more on $\mathcal{H}_2$ and $\mathcal{H}_\infty$ designs. It is possible to improve the robustness properties of the closed loop system by using a design such as D–K synthesis [52]. However, the order of the compensator can often increase dramatically, causing the computational burden to become unreasonable. On the other hand, the $\mathcal{H}_2$ design can achieve a reasonable performance level with a reasonable amount of computation. This version of the $\mathcal{H}_2$ problem is very similar in nature to the sensitivity weighted LQG controller derived in [22], [23]. The sensitivity weighted LQG controller was shown to compare favorably with other robust control techniques, both in simulation and in closed loop implementation.

6.3.2 Example of Input Design

Our first example of the input design will use the same noise bound as shown in figure 6-4. As opposed to the previous examples, however, the initial data points will not be chosen necessarily near the peak of each mode. Instead, 5 data points are chosen in the vicinity of each mode, but covering more of the frequency region. The initial data is shown in figure 6-8.

Since the initial data points are not where the signal to noise ratio[2] is highest, the initial parameter intervals will be much larger. This is done to highlight the ability of the input design algorithm to find appropriate inputs when there is a large amount of uncertainty. The resulting parameter intervals are shown in appendix B. This

[2]By signal to noise ratio, we mean the ratio of the magnitude of the noise free system to the bound on the noise.

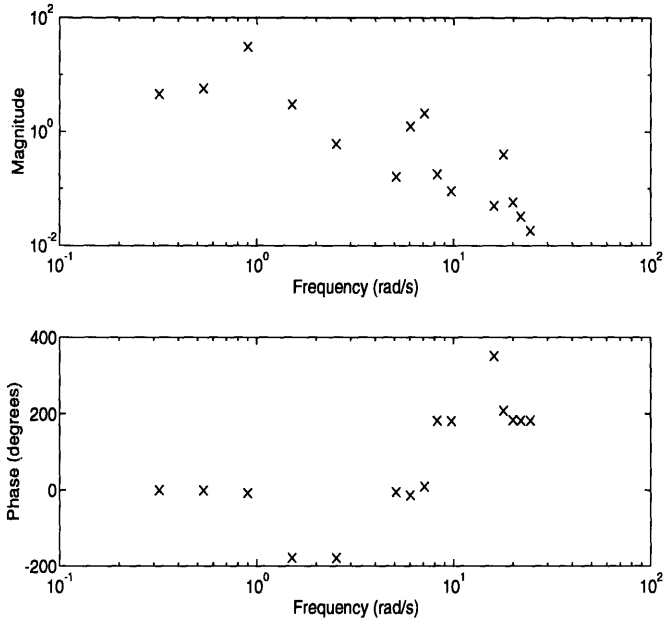

Figure 6-8: Initial data points used in the input design algorithm.

appendix lists both the parameter bounds, and the resulting μ–sensitivities for each iteration in the input design procedure.

It should be stressed that the data in figure 6-8, together with the a priori information on the parameters in our model and on the noise, as outlined in section 6.1, constitute the entire knowledge of the system at the beginning of the iterative algorithm. The reason that this is a difficult design problem is that we are using so few data points. Using the a priori knowledge, and the specified parameterization of the system, we can determine a rough estimate of the system, and generate new inputs to reduce the uncertainty quickly. This is the advantage of the approach presented in this research.

To highlight the uncertainty arising from only using these 15 data points, several values of the parameters are chosen from within the parameter intervals resulting from the identification. We will plot the transfer function for each set of parameters. This will give an indication of the amount of uncertainty from a transfer function perspective, and demonstrate the improvement due to new data points as the algorithm proceeds.

Table 6-6 lists the choices for these parameters. The actual uncertainty in the frequency domain due to the parameter intervals is not necessarily bounded by the transfer functions resulting from these parameter choices. These transfer functions represent a small sample of all of the possible transfer functions which result from this parametric uncertainty.

	a_{11}	a_{01}	b_{01}	a_{12}	a_{02}	b_{02}	a_{13}	a_{03}	b_{03}
Transfer Function 1	-1	1	-1	-1	1	-1	-1	1	-1
Transfer Function 2	1	-1	1	1	-1	1	1	-1	1
Transfer Function 3	-1	-1	1	-1	-1	1	-1	-1	-1
Transfer Function 4	1	-1	-1	-1	-1	-1	1	-1	1
Transfer Function 5	1	1	1	1	1	-1	1	-1	-1
Transfer Function 6	0	-1	-1	0	-1	-1	0	-1	-1
Transfer Function 7	-.5	0	0	-.5	0	0	-.5	0	0
Transfer Function 8	.5	0	0	.5	0	0	.5	0	0
Transfer Function 9	0	0	0	0	0	0	0	0	0

Table 6-6: Parameter values used to generate sample transfer functions. The value used for parameter θ_j is $\hat{\theta}_j + x(\overline{\theta}_j - \hat{\theta}_j)$, where $\hat{\theta}_j = .5(\overline{\theta}_j + \underline{\theta}_j)$, and x is the value in the table for θ_j.

The transfer functions for these choices of the parameters after the initial identification are shown in figure 6-9. We see that there is a large amount of uncertainty present. The DC value ranges from about 2.15 to 9. There is a wide variation in the pole and zero locations. This is highlighted in figure 6-10, where the poles and zeros of these transfer functions are plotted. We see that it is possible to have nonminimum phase zeros, as well as zeros which are purely real.

To understand the uncertainty which is most important to reduce, we need to understand how the compensator interacts with the system. To visualize this, the $\mathcal{H}_2$ compensator was designed for the initial model. In figure 6-11, the nominal loop transfer function is plotted (i.e. the nominal value of the plant, which occurs at the midpoints of the uncertainty intervals, together with the compensator). Since this is an $\mathcal{H}_2$ design, we know that the nominal closed loop system is stable. Thus, the system with the uncertainty is guaranteed to be stable if the loop transfer function

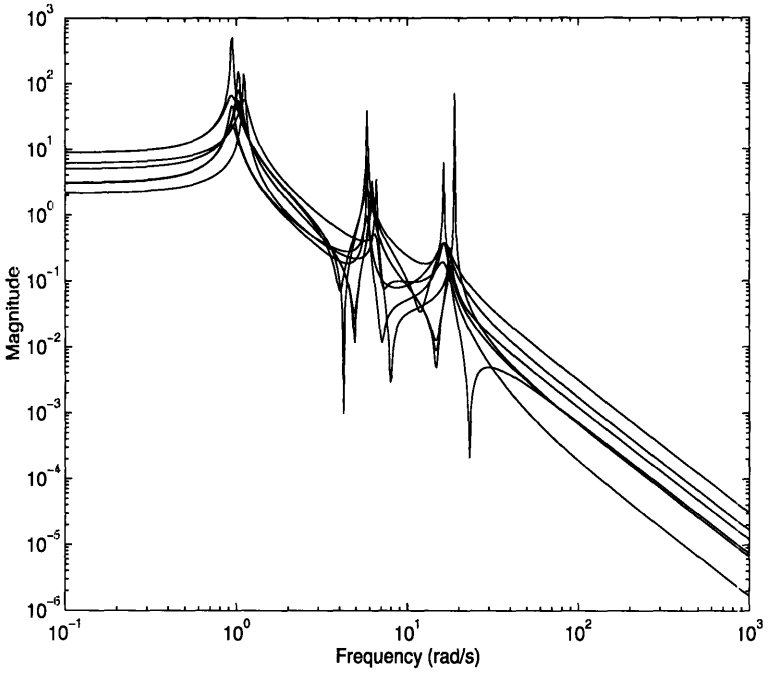

Figure 6-9: Transfer functions after identification using only fifteen initial data points.

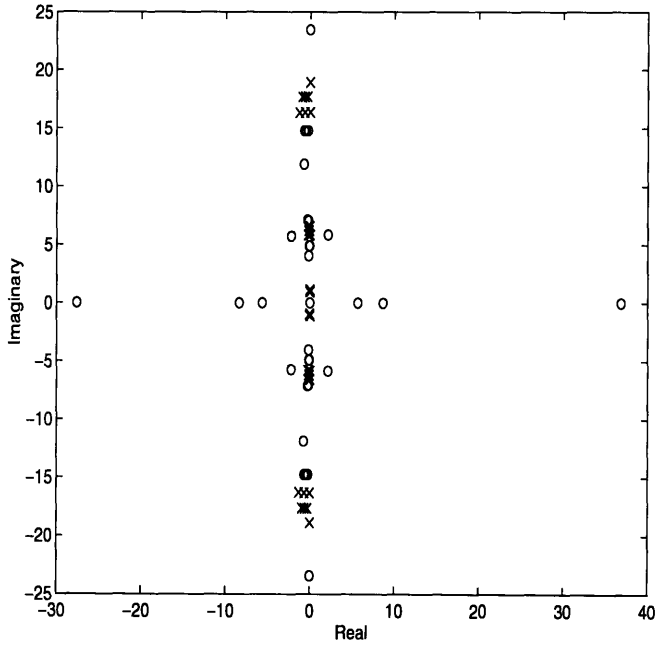

Figure 6-10: Open loop pole–zero pattern after identification using initial data points.

does not encircle the critical point, where the magnitude is 1 and the phase is 180 degrees.[3]

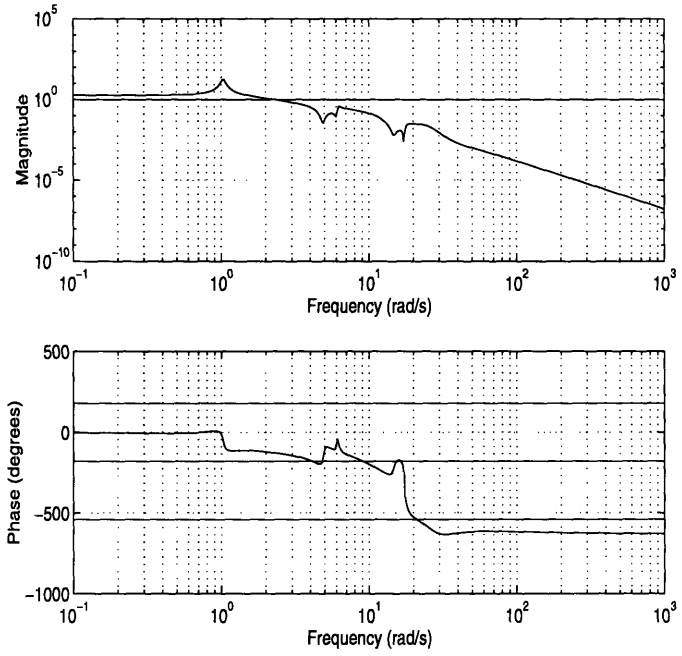

Figure 6-11: Nominal loop transfer function $-GK$ for the initial model.

We see from figure 6-11 that the compensator is phase stabilizing the first mode of the system and gain stabilizing the other modes. Thus, it is important that the phase remains away from 180 degrees until after the magnitude is less than 1. However, in order to meet the performance goals, the compensator needs to have authority over the frequency range where we desire sensitivity reduction. Thus, it is important to have an accurate model of the system between the first two modes, near crossover. We also need to have a good estimate of the gain of the system. We will see that the input design methodology tries to pick inputs which help get an accurate model of these aspects of the transfer function.

The following procedure is used for the input design procedure. At each iteration, all of the available data is used to determine the parameter intervals. We wish to guarantee the same bound on the ratio of conservatism as in the previous examples;

[3]The actual compensator was designed using positive feedback. Thus, the true critical point in the complex plane is at $s = 1$, and not $s = -1$. In order to adhere to standard notation, we have plotted $-GK$, so that the critical point is shifted back to $s = -1$.

namely $\gamma < 1.1$. However, we will only partition the parameter space if the required number of partitions is less than 200.

We will limit our choice of inputs to unit amplitude sinusoids at frequencies between .1 rad/sec and 1000 rad/sec. It is assumed that the data is observed in steady state, and that any residual transient is considered part of the noise. Thus the output consists of the magnitude and phase at a particular frequency. To limit ourselves to a finite number of frequencies from which to choose the sinusoid, we initially allow ourselves to choose from 500 different frequencies, logarithmically spaced between .1 rad/sec and 1000 rad/sec. To increase the number of frequencies from which we can choose the sinusoid, at the k^{th} iteration we will allow ourselves to choose from $500 + 25k$ different frequencies, logarithmically spaced between .1 rad/sec and 1000 rad/sec.

A summary of the results generated by the iterative algorithm is shown in figures 6-12 and 6-13. In figure 6-12, the value of the peak of $\overline{\mu}$ is plotted for each iteration. Below each point, the "optimal" frequency calculated for the next input is shown. We see that $\overline{\mu}$ does, in fact, decrease at each iteration, until $\overline{\mu} < 1$. We also see that the algorithm generates input frequencies near the natural frequency of one of the modes in our system. This is because the signal to noise ratio is highest near the natural frequency, and thus at this frequency we can learn more about all of the parameters of a particular mode.

In figure 6-13, $\overline{\mu}$ is plotted as a function of frequency. It is interesting to note that $\overline{\mu}$ achieves its peak near the second zero of the system. This is expected, because it is right near this zero (at $\omega \approx 4.4$ rad/sec.) where the loop transfer function is rolling off. However, the algorithm does not specify sinusoids in the frequency range where $\overline{\mu}$ reaches its peak. Since the signal to noise ratio is low near this zero, we would not learn much about the parameters by applying a sinusoid near the zero frequency. So although we seek an input to lower $\overline{\mu}$ near the zero, this can best be done by applying an input at a different frequency to overcome signal to noise problems.

The ultimate performance achieved by the algorithm is certainly dependent upon the control methodology. As an indication of the possible performance for this ex-

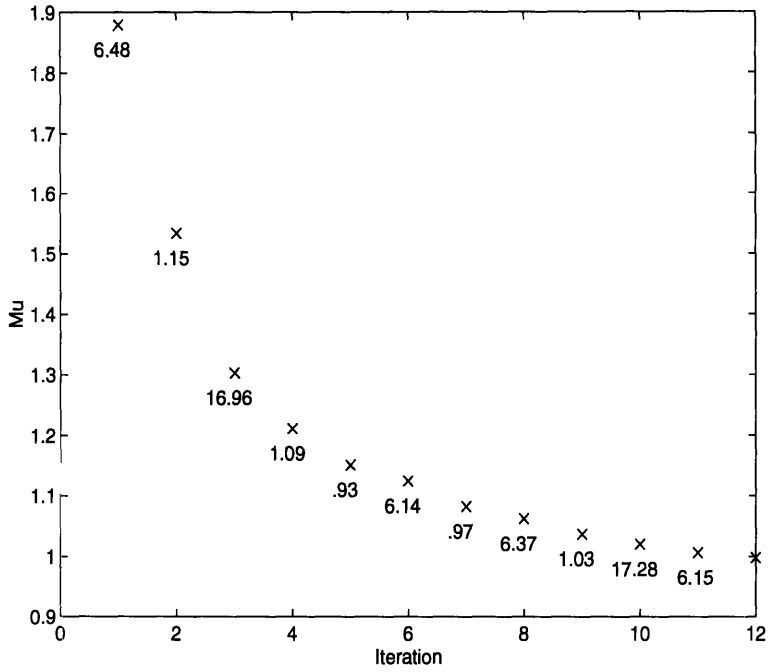

Figure 6-12: The decrease in $\overline{\mu}$ as the algorithm proceeds. The frequency for the next input is shown next to the value of $\overline{\mu}$ for each iteration.

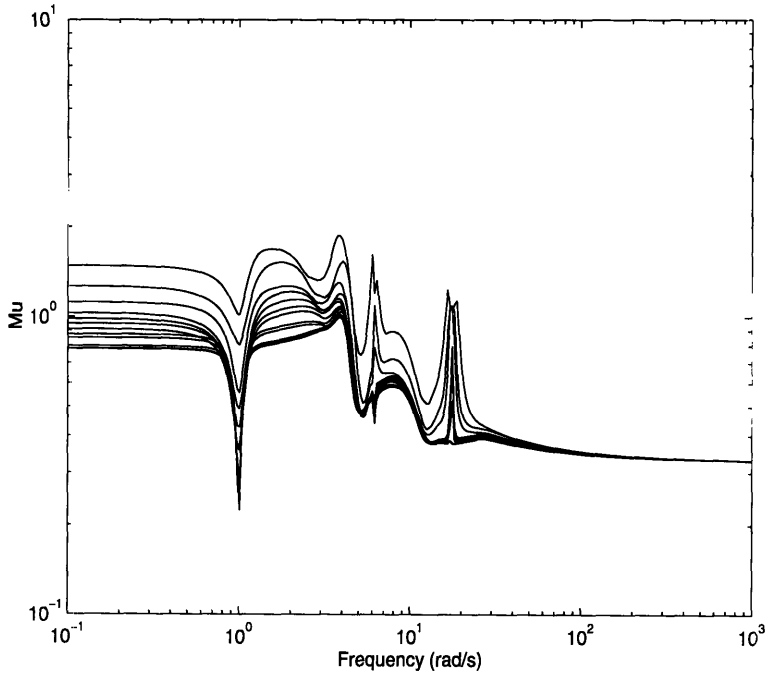

Figure 6-13: A plot of $\overline{\mu}$ as a function of frequency for each iteration.

ample, an $\mathcal{H}_2$ compensator was designed on the nominal system, with no parametric uncertainty. The robust performance was then calculated assuming that the only uncertainties were due to the unmodelled dynamics, and the robust performance specification. The resulting values of $\overline{\mu}$ was approximately .9.

We can thus conclude that the $\mathcal{H}_2$ control methodology is an appropriate choice for this particular problem. If the value of $\overline{\mu}$ was greater than 1, we would have been less willing to use an $\mathcal{H}_2$ methodology. (It is possible with an $\mathcal{H}_2$ methodology to have a lower value of $\overline{\mu}$ if we include a small amount of parameter uncertainty, but $\overline{\mu} > 1$ is an indication that we should try other methodologies). On the other hand, if $\overline{\mu}$ was much less than 1, we could conclude that this control methodology could achieve a more stringent performance goal. A value of .9 indicates that we have a reasonable performance goal, which most likely will be achieved once the parametric uncertainty is reduced.

The goal of the iterative algorithm is to reduce $\overline{\mu}$ where it reaches its peak. In this case, we wish to reduce $\overline{\mu}$ near the first zero (at $\omega \approx 4.4$) of the open loop system. All of the parameters affect the transfer function in this frequency region, and so the algorithm must decide which parameters are affecting it the most, and which input will improve $\overline{\mu}$ the most.

In fact, the three most important parameters, as determined by the μ–sensitivities, are the residues of the three modes (recall that $b_{i1} = 0$ for all of the modes). The μ–senstivity for b_{01} is .806; for b_{02} it is .754, and for b_{03} it is .192. This can be seen from table B-1 in the appendix. The analysis of the effect of the inputs on the uncertainty show that we expect to be able to get more information from the second mode than the first. This can be see in figure 6-14. In this figure, we have plotted the estimated change in $\overline{\mu}$ (calculated in step 7 of algorithm 5.1) for each of the 500 unit amplitude sinusoids from which the algorithm will choose the next input. Thus, each frequency point represents a different input. Note that there is only a small number of frequencies where the estimated improvement is nonzero.

We see from figure 6-14 that we expect the most improvement in $\overline{\mu}$ if we apply an input sinusoid near the natural frequency of one of the first three modes of the system.

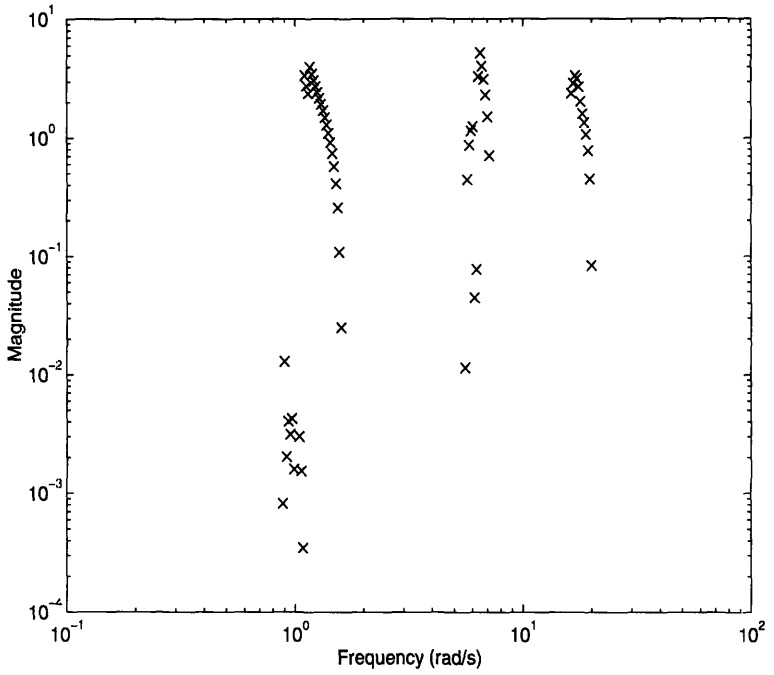

Figure 6-14: The estimated improvement in $\overline{\mu}$ for each of the unit amplitude sinusoids at the first iteration.

It is important to note that we do not actually expect an improvement as large as indicated by figure 6-14. The large values are due to the first order approximation of the effect of the change in uncertainty on $\overline{\mu}$ (i.e. the μ–sensitivities). Because the expected improvement in the parameter intervals is large, the estimated improvement in $\overline{\mu}$ is large. The primary use of the information in this figure is to determine which inputs are effective in improving performance, and to provide a ranking of these inputs.

Based upon figure 6-14, we see the algorithm has determined that there are many input frequencies which would be helpful in improving $\overline{\mu}$. The input frequency which is ranked the highest is 6.48 rad/sec. This is therefore the frequency chosen for the next experiment.

Let us examine the improvement after adding this one frequency point. Figure 6-15 shows the transfer functions for the choice of parameters listed in table 6-6, with the corresponding pole–zero pattern in figure 6-16. We see that there is a great improvement in all aspects of the transfer function, especially at frequencies near the second mode. However, there is still a large amount of uncertainty.

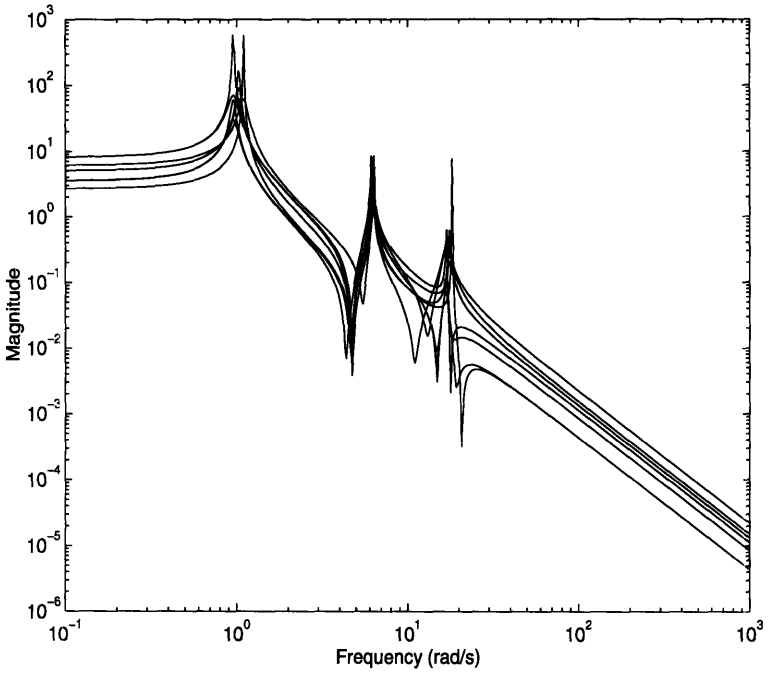

Figure 6-15: Transfer functions after applying a sinusoid at 6.48 rad/sec.

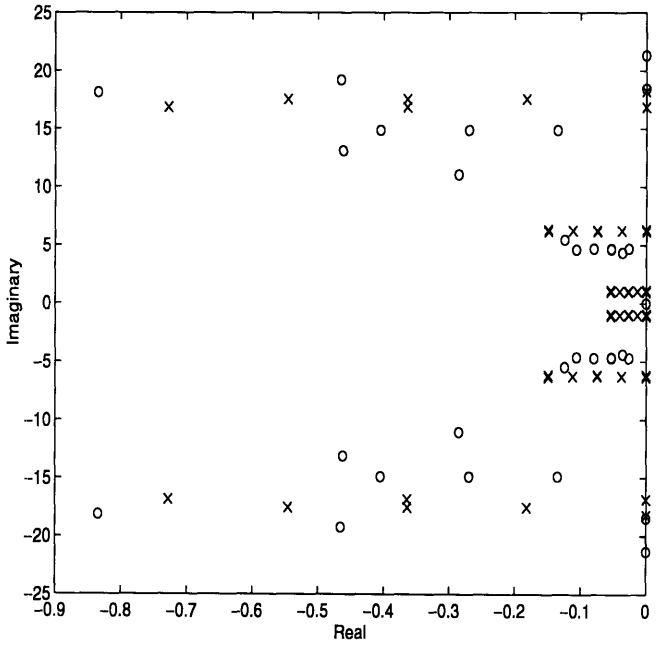

Figure 6-16: Open loop pole–zero pattern after applying a sinusoid at 6.48 rad/sec.

Looking at the new μ–sensitivities, we see that it is still the residue terms that are the most important for improving $\overline{\mu}$. From table B-2, the value of the μ–sensitivity for b_{01} is .850; for b_{02} it is .351; and for b_{03} it is .107. We see that b_{01} is at least twice as important as any of the other parameters. It is also interesting to note that although there is a wide variation in the natural frequency and damping ratio of the third mode, the μ–sensitivities for a_{03} and a_{13} are small (see table B-2). This is because the important part of this mode for decreasing the peak of the $\overline{\mu}$ plot is the DC value.

Figure 6-17 shows the estimated improvement in $\overline{\mu}$ for each of the possible choices for the next input. We see the estimated improvement in $\overline{\mu}$ is much larger if we choose an input frequency near the first mode. The algorithm therefore chooses an input sinusoid at 1.15 rad/sec.

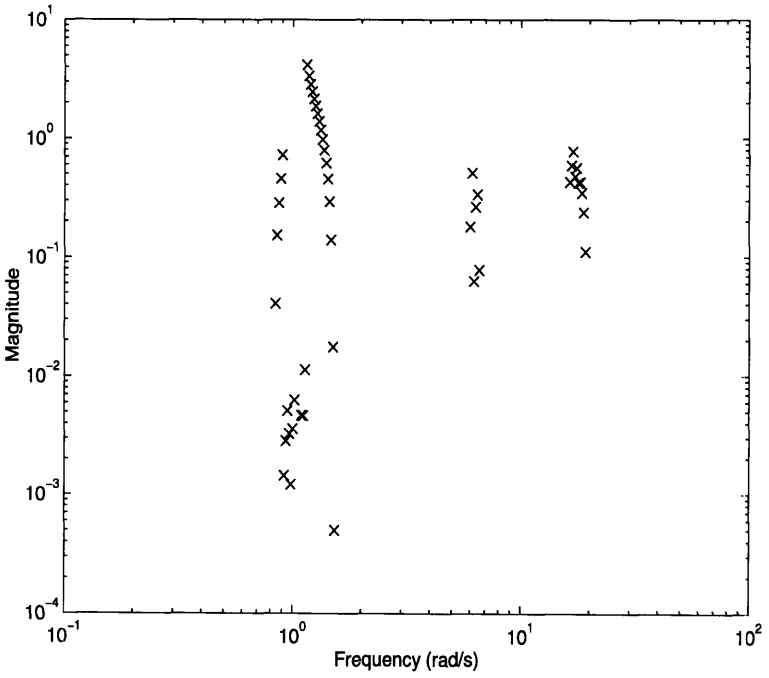

Figure 6-17: The estimated improvement in $\overline{\mu}$ for each of the unit amplitude sinusoids at the second iteration.

We now apply the new input. Figure 6-18 shows the transfer functions for the parameters selected in table 6-6. We see that there is much less uncertainty at both the first pole, and the first zero. Also, we have much less uncertainty in the DC gain of the system. However, there is an interesting phenomenon which appears in this figure. There is enough uncertainty in the parameters to cause a pole–zero cancellation at

the third mode for some parameter values. This is also seen in figure 6-19, where the poles and zeros are plotted for these transfer functions.

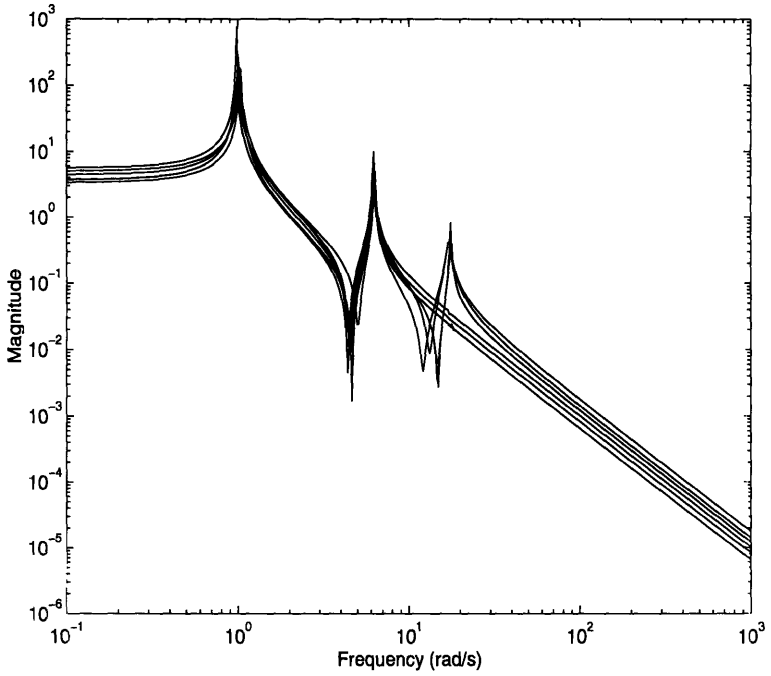

Figure 6-18: Transfer functions after applying a sinusoid at 1.15 rad/sec.

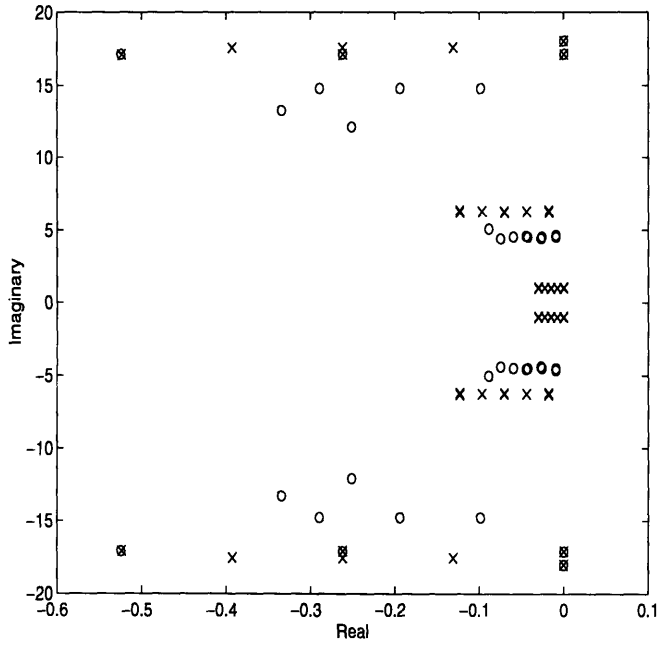

Figure 6-19: Open loop pole–zero pattern after applying a sinusoid at 1.15 rad/sec.

Another interesting thing to notice is that some of the zeros are still nonminimum phase. Although this is hard to tell from figure 6-19, some of the zeros actually have a real part which is approximately 10^{-5}. Although this was not apparent in figure 6-16, there have been values of the parameters which created nonminimum phase zeros for all previous iterations. The values of the parameters plotted in figure 6-16 did not happen to contain a nonminimum phase zero, as we are not guaranteed to bound the ranges of possible poles and zeros with our sample parameter values.

To explain the next optimal input at a frequency of 16.96 rad/sec, we again look at the μ–sensitivities and the effect of the inputs on the uncertainty. Once again, it is the residue terms which are the most important to identify accurately according to the μ–sensitivities. As seen in table B-3, the μ–sensitivity for b_{01} is .479; for b_{02} it is .280; and for b_{03} it is .094. However, in figure 6-20 we see that the algorithm has ranked an input frequency near the third mode highest. The reason for this is that there is so much uncertainty in this mode (see table B-3), the estimated improvement in $\overline{\mu}$ is greater when we apply an input which attempts to improve these parameters even though the μ–sensitivities are smaller. The algorithm therefore specifies an input near the natural frequency of this third mode.

We see that this input has indeed improved the transfer function estimates. It has certainly improved the location of the second zero and the third pole, as there is no longer a pole–zero cancellation, at least for the parameter values chosen. This is seen in figure 6-21. Also, in figure 6-22, we can now clearly see the pole–zero pattern. Since the system is lightly damped, the natural frequency of each pole or zero is approximately equal to its imaginary component; it can therefore be estimated from the value in the vertical direction in the pole–zero plot. We see that for all the poles and zeros (except perhaps the second zero), the frequency is known fairly accurately. However, the damping ratios are not well known.

This pattern of input design continues according to the values in figure 6-12. Notice that as we add more inputs, the improvement in $\overline{\mu}$ decreases. This is to be expected, because it becomes less likely that we will gain a large amount of information from any of the parameters. After adding 12 new data points, we have achieved

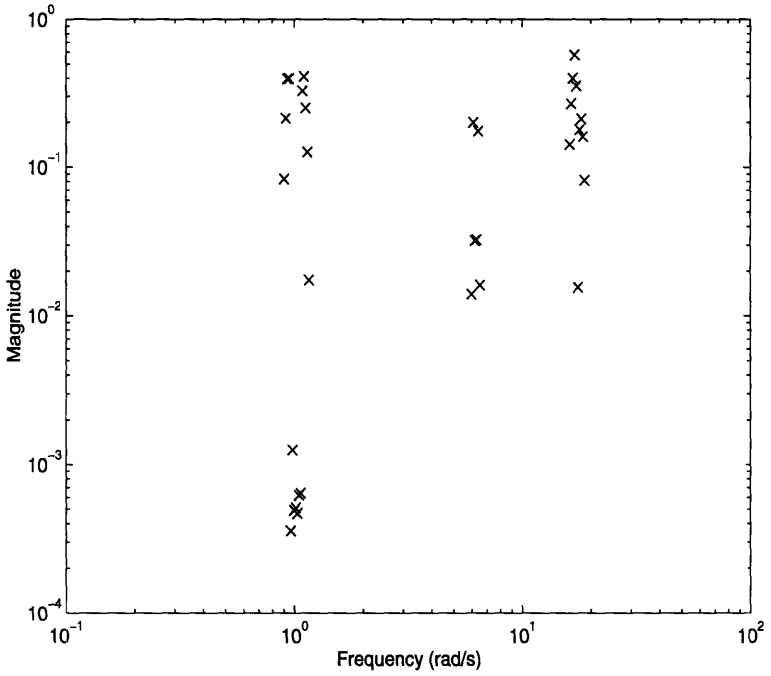

Figure 6-20: The estimated improvement in $\overline{\mu}$ for each of the unit amplitude sinusoids at the third iteration.

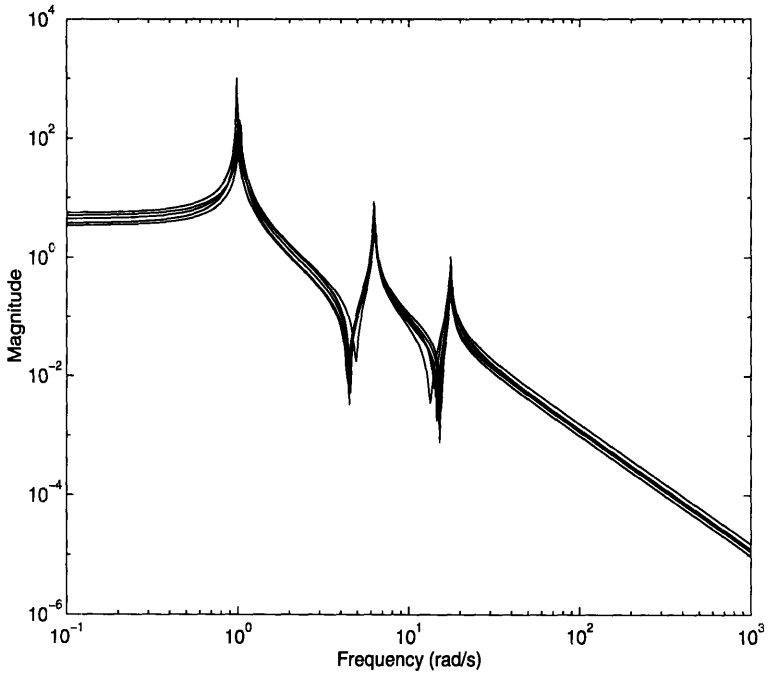

Figure 6-21: Transfer functions after applying a sinusoid at 16.96 rad/sec.

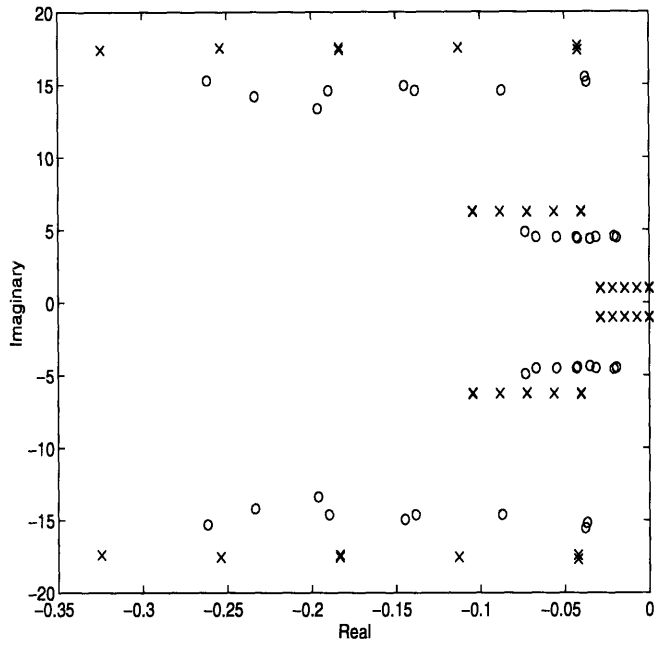

Figure 6-22: Open loop pole–zero pattern after applying a sinusoid at 16.96 rad/sec.

$\overline{\mu} < 1$. The final uncertainty can be seen by plotting the transfer functions and pole–zero pattern for the values of the parameters in table 6-6. These are shown in figure 6-23 and 6-24 respectively. We see that the uncertainty in the first two poles and the first zero have been significantly reduced. Also, the DC value is very well known, ranging from about 3.94 to 4.23. There is still some uncertainty in the damping of the third pole and the second zero, but these do not have as large an impact upon our control design.

Finally, let us examine how the compensator has changed as the algorithm progressed. The magnitude of the compensators for the first few iterations, as well as the magnitude of the final compensator, are shown in figure 6-25. All of the compensators have the characteristics of a typical $\mathcal{H}_2$ design. The compensators try to invert the plant, and insert the desired dynamics. Thus we see that the compensator has zeros at approximately the same locations as the poles of the plant, and poles at approximately the same location as the zeros of the plant. However, due to the uncertainty in the system, exact pole zero cancellations are not possible. As the uncertainty in the system is reduced, we have better knowledge of the location of the poles and zeros in

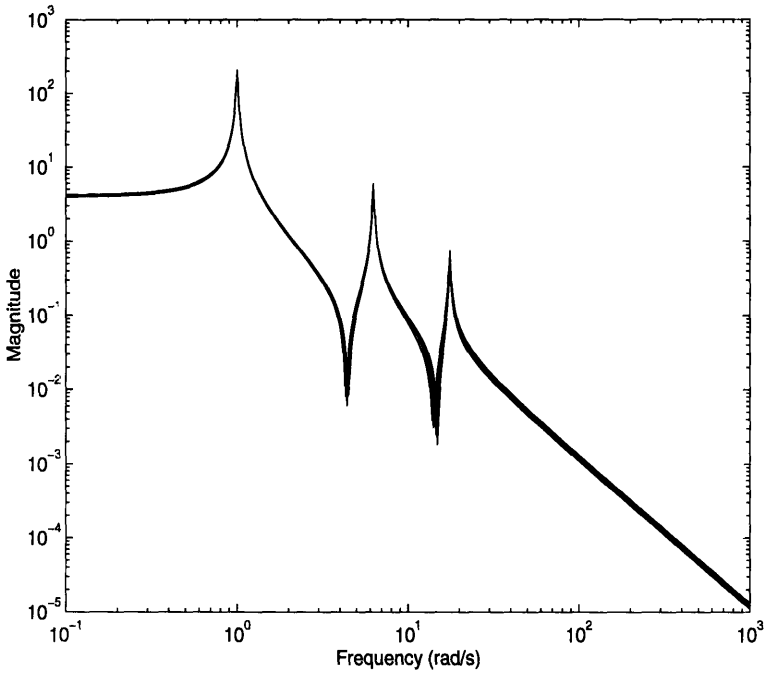

Figure 6-23: Final variation in transfer functions.

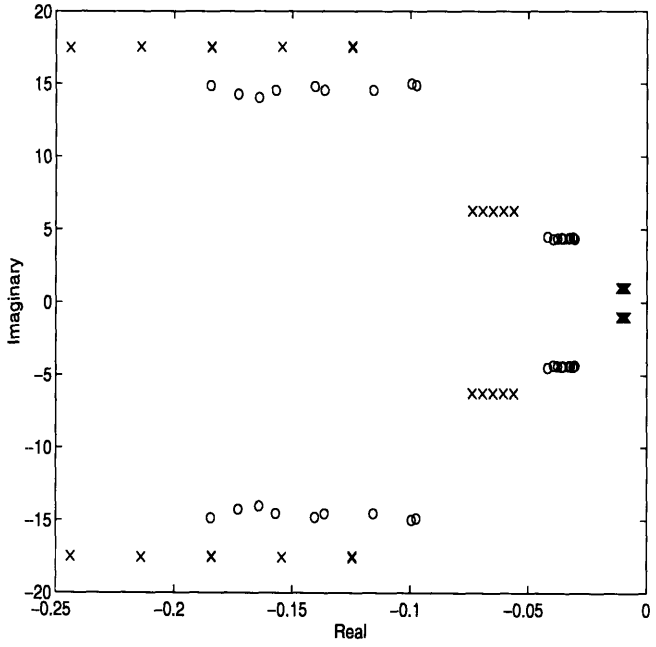

Figure 6-24: Final variation in open loop pole–zero pattern.

the system. As a result, the poles and zeros of the compensator become more lightly damped.

We also see that as the uncertainty decreases, the gain of the compensator increases. With less uncertainty in the system, the compensator can exert more control authority. This larger gain helps achieve the desired reduction in the sensitivity at low frequencies. We can therefore conclude that the improvement in performance is caused, in part, by higher authority control designs. Although the improvement in the parameter intervals helps improve performance even without changing the compensator, the iterative algorithm takes advantage of the tighter bounds to design higher authority compensators.

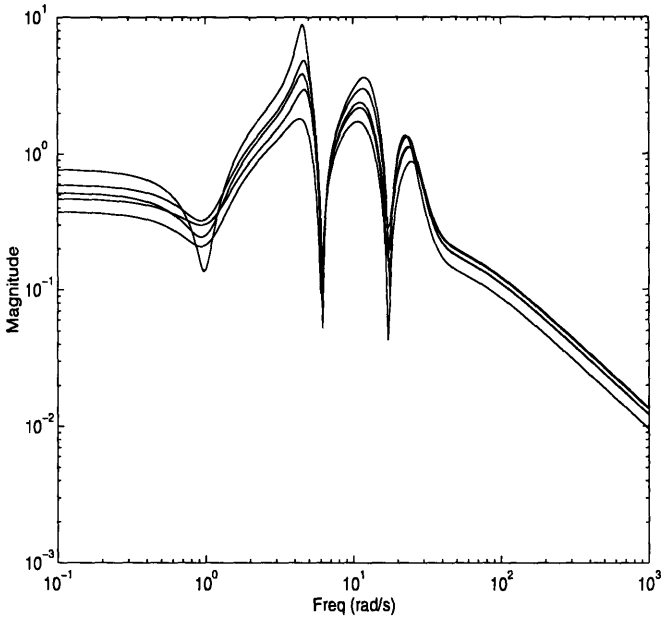

Figure 6-25: The magnitude of the compensator over several iterations.

6.3.3 Input Design with a SNR of 10

In the first example of the input design, we saw that the inputs were always chosen near the natural frequency of one of the modes in our model. This was because the signal to noise ratio is highest here, and therefore more information could be obtained. In this example, we will explore what happens when the signal to noise ratio is the

same at all frequencies.

For this example, we will use the same a priori information as in the previous example except for the bound on the noise. Here, the frequency dependent bound on the noise was chosen as 10% of the magnitude of the true system at each frequency. Therefore we have the same signal to noise ratio at each frequency.

The results of the input design algorithm are shown in figures 6-26 and 6-27. We see that initially the value of $\overline{\mu}$ is lower than in the previous example. This is because the initial data points are spread throughout the frequency spectrum. In the previous example, most of these points had signal to noise ratios which were lower than 10. Thus, initially, we have a more accurate model. The actual parameter values are shown in tables B-13 to B-35.

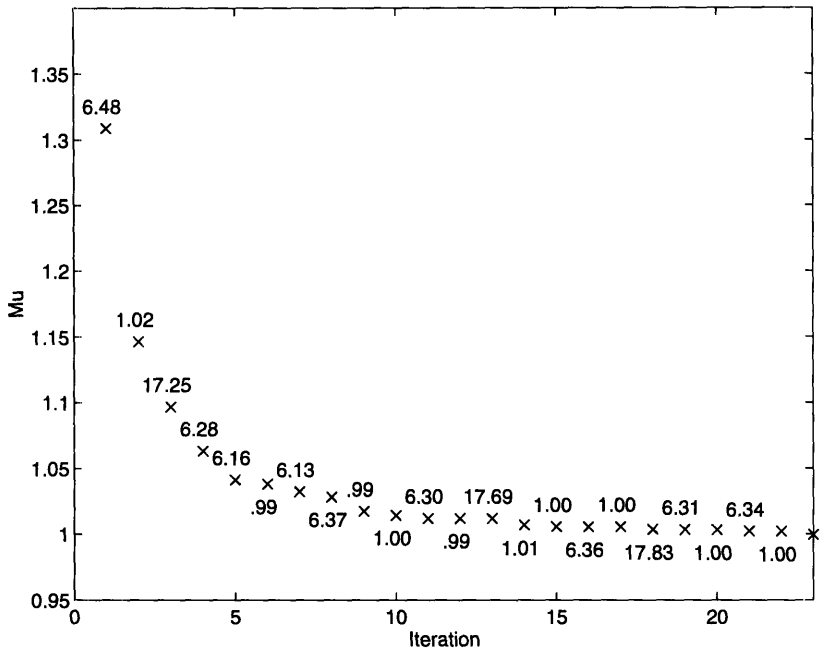

Figure 6-26: The decrease in $\overline{\mu}$ as the algorithm proceeds. The frequency for the next input is shown next to the value of $\overline{\mu}$ for each iteration.

In figure 6-27, we have plotted $\overline{\mu}$ as a function of frequency for all of the iterations. We see that although we do decrease the peak of the $\overline{\mu}$ plot at each iteration, we do not necessarily reduce $\overline{\mu}$ at every frequency. Since our performance was defined as the peak of the $\overline{\mu}$ plot, this is not a concern. This is, in fact, to be expected from any

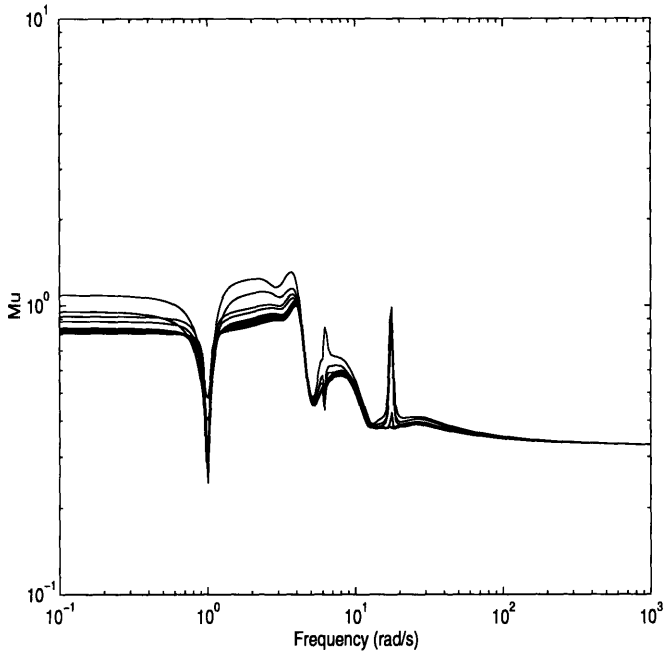

Figure 6-27: A plot of $\overline{\mu}$ as a function of frequency for each iteration.

methodology which attempts to reduce the peak of the $\overline{\mu}$ plot.

Looking at figure 6-26, we see that although we start at a lower value of $\overline{\mu}$, it actually takes more iterations to achieve $\overline{\mu} < 1$. This again is due to the signal to noise ratio. In the previous example, the inputs were chosen at frequencies where the signal to noise ratio was larger than 10. Since the bound on the noise is larger in this example at the input frequencies chosen, the identification algorithm is not able to get as tight a bound on the parameters.

Even though the signal to noise ratio was set to be the same at all frequencies, the inputs were still chosen near the natural frequencies of the modes. This is due to the other sources of noise in the system, namely the unmodelled dynamics and the model mismatch. Since these types of errors tend to vary slowly over frequency, the effective signal to noise ratio was larger at the natural frequencies of the modes. Thus, we still have more information at the natural frequencies. Only when the bound on the additive noise is higher at the natural frequencies than elsewhere will the algorithm specify inputs away from the natural frequencies.

6.3.4 Two Examples with Larger Noise at the Natural Frequencies

We now consider two examples where the signal to noise ratio is greatly reduced near the natural frequencies of the modes in our model. The bound on the noise as a function of frequency is shown in figure 6-28. We see that the noise is much larger near the natural frequencies of the modes than near the zeros. We would therefore expect that the "optimal" input frequencies generated by the algorithm will not be near the true natural frequencies.

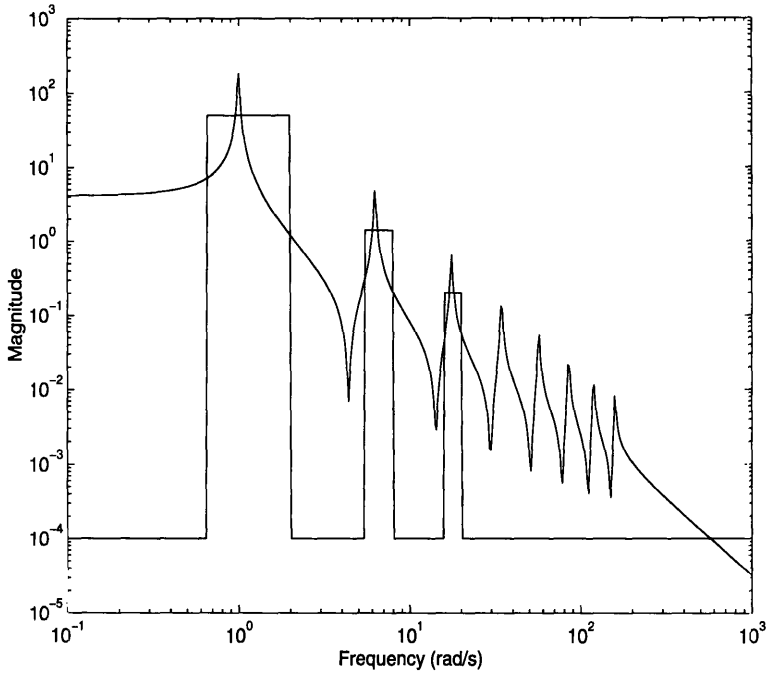

Figure 6-28: Noise free system, and bound on the additive noise as a function of frequency for the third example.

Since the signal to noise ratio is poor near the natural frequencies, for our initial frequency points we will choose 15 data points logarithmically spaced between .3 rad/sec and 30 rad/sec. Other than this choice of initial data points and the bound on the noise, the procedure for these examples is the same as in the previous examples.

For the first of these examples, we use the noise bound in figure 6-28, as well as the bound on the unmodelled dynamics in figure 6-3. The results are shown in figure 6-29 and 6-30. We see that in this example, we did not achieve $\overline{\mu} < 1$ within the 10

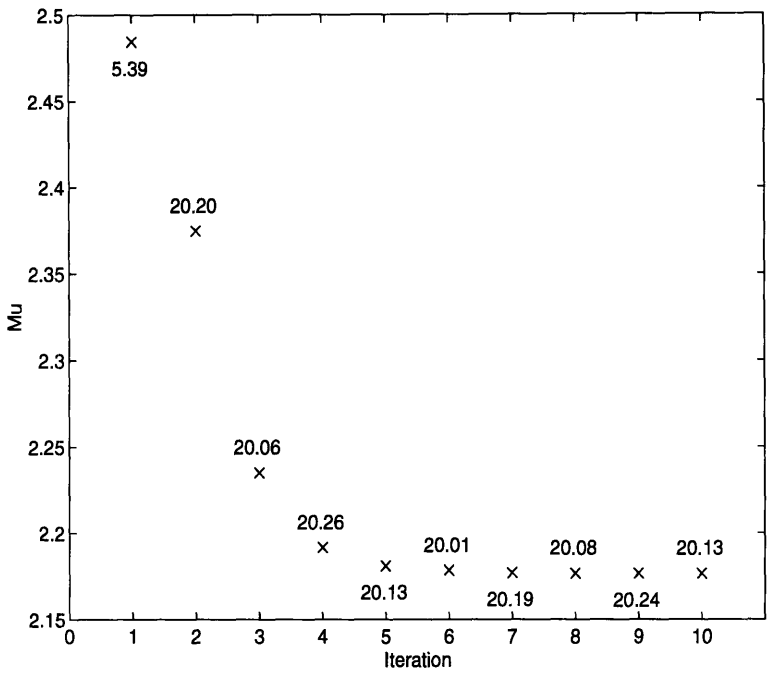

Figure 6-29: The decrease in $\overline{\mu}$ as the algorithm proceeds. The frequency for the next input is shown next to the value of $\overline{\mu}$ for each iteration.

iterations plotted. In fact, $\overline{\mu}$ did not decrease appreciably after many more iterations.

To understand why we were not able to achieve $\overline{\mu} < 1$, we have to understand how the inputs are being chosen. Due to the large noise near the natural frequencies, the sensitivity analysis indicates that there is little information to be obtained at these frequencies. From figure 6-28, we would expect there to be a lot of information where the bound on the noise becomes small. However, recall that there are other sources of noise which must be considered. Added to this noise bound is the bound on unmodelled dynamics. So although there appears to be a high signal to noise ratio near the zeros of the system, the unmodelled dynamics cause this signal to noise ratio to be lower than we originally thought.

There is also another source of noise in the system. This is the error due to model mismatch. Because our initial model is so poor, this error can be large. In the previous examples, this error was quickly reduced because our inputs were chosen near the natural frequencies, where these effects are reduced. For this particular example, these effects are large at the input frequencies chosen. We therefore have a

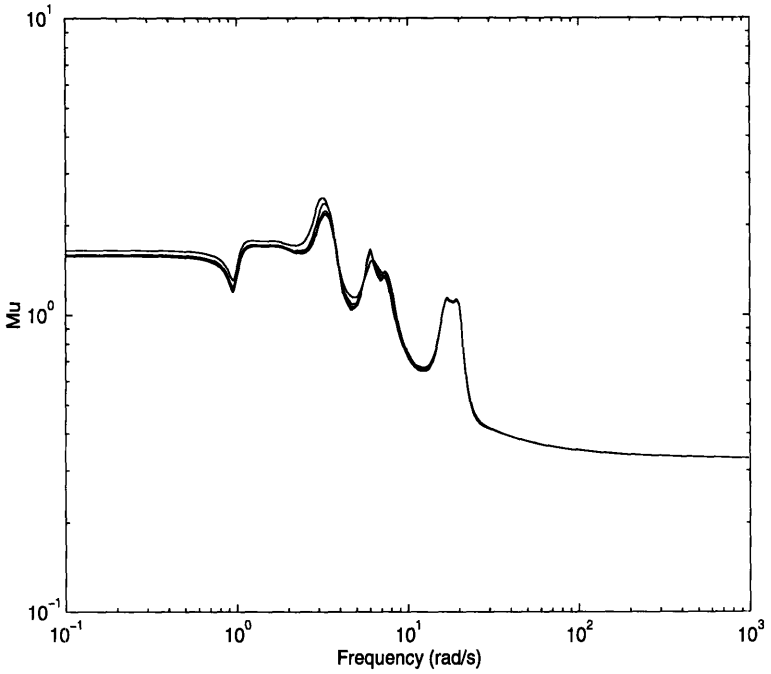

Figure 6-30: A plot of $\overline{\mu}$ as a function of frequency for each iteration.

large amount of noise in the system, even at the zeros.

Let us examine the choice for the input frequencies in more detail. The first input was chosen at 5.39 rad/sec. This is right at the border of where the additive noise bound increases. Thus, we are trying to get more information about the second mode, and to do this we choose an input as close to the natural frequency of the second mode as we can, while still avoiding the large additive noise.

The next input is chosen at 20.20 rad/sec. We are now trying to identify the parameters of the third mode, especially the residue. The choice for this input frequency is made for the same reason as the first choice.

After these first two inputs, the analysis still indicates that the best input is just above 20 rad/sec., where the additive noise bound drops. Since the set of frequencies from which we choose our input changes at each iteration, the frequency chosen varies by a small amount, but is always a little over 20 rad/sec. In this case, we are incorrectly assuming that the noise is dominated by a stochastic component. In fact, the major source of noise is the model mismatch. So although the expected improvement in $\overline{\mu}$ is greatest at these frequencies, this analysis used an incorrect

assumption on the noise. We therefore become caught in a trap of continuing to try the same frequency region when there is little information available.

In order to understand the effects of the unmodelled disturbances on the parameter bounds, we will now repeat the same example. However, this time we will remove the unmodelled dynamics from the true system. The noise bound used in the identification algorithm will therefore consist only of the additive noise bound; there is no reason to include the bound on unmodelled dynamics since we have removed them from the system. However, the control design and analysis will be done exactly the same way, using the weight $W_1(s)$ as part of the robust stability criterion.

The results are shown in figure 6-31 and 6-32. The iterations begin very similar to before. In fact, we pick exactly the same inputs. However, after the fourth iteration, the value of $\overline{\mu}$ suddenly drops. We have managed to get enough information in these four choices of inputs to vastly improve the parameter bounds.

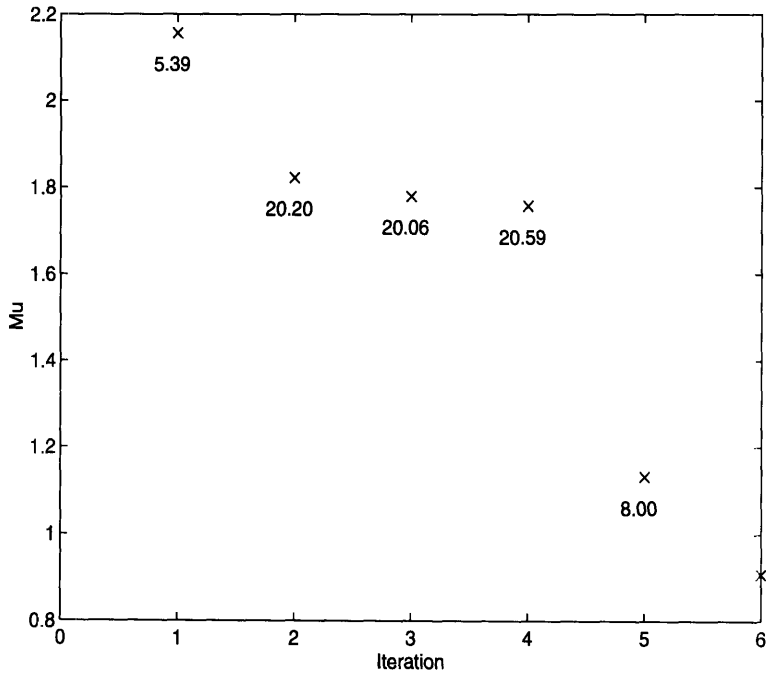

Figure 6-31: The decrease in $\overline{\mu}$ as the algorithm proceeds. The frequency for the next input is shown next to the value of $\overline{\mu}$ for each iteration.

To understand what has happened, we need to understand how the uncertainty is affecting the system. The parameter bounds and μ–sensitivites are listed in appendix

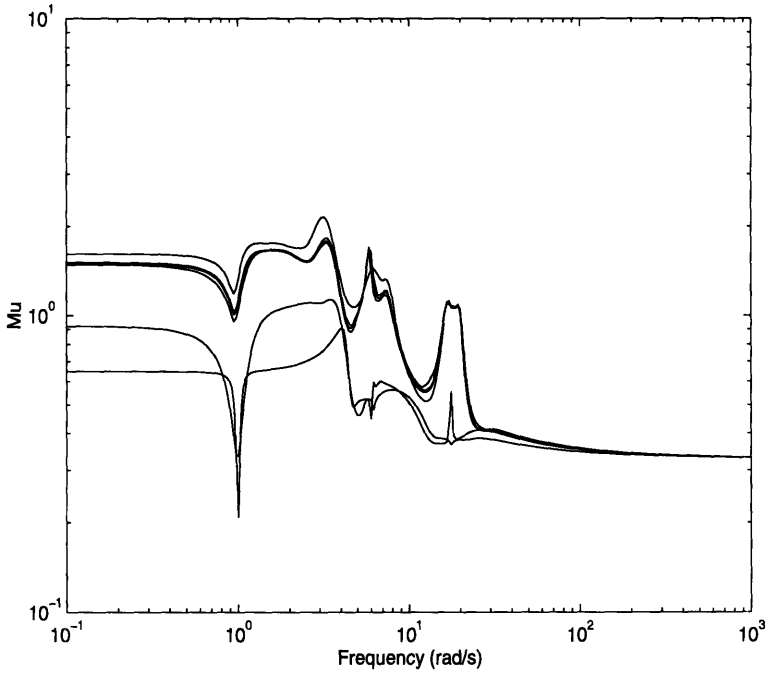

Figure 6-32: A plot of $\overline{\mu}$ as a function of frequency for each iteration.

B. We see in table B-46 that after the initial identification, there is a large uncertainty in the residues of all of the modes. The largest μ–sensitivity is the one for the residue of the second mode, b_{02}. The reason that this parameter is having such a large impact on the system is that there is such a large interval range. In fact, we don't even know the sign of b_{02}. The same is true for b_{03}.

We see from table B-47 that the input has had the desired effect. The interval range for b_{02} is much smaller now, although we still don't know the sign. The reason we were able to learn so much more in this example as opposed to the previous example is that the noise bound has been decreased; we are no longer adding the bound on the unmodelled dynamics to the additive noise bound. Notice that since the interval ranges have reduced by a large amount, we now have a tighter bound on the model mismatch. So the effective noise in the system has been significantly reduced.

The algorithm specifies the next input 20.20 rad/sec. As seen in table B-47, although the μ–sensitivity is highest for b_{02}, we expect a greater improvement in $\overline{\mu}$ if we apply an input to get more information from the parameters of the third mode.

For instance, the current knowledge on b_{03} is that it can range from -10.4 to 18.3. The input is chosen to reduce this uncertainty, and is effective in doing so. It also reduces the model mismatch affecting the first two modes. Thus, we are improving all the parameter bounds with this input.

The next input is chosen at 20.06 rad/sec, for a similar reason. The effect is that the uncertainty in the third mode has been significantly reduced. This has caused the model mismatch affecting the second mode to be reduced. The effective noise bound at the second mode is now small enough that we can determine the sign of b_{02}. This in turn greatly reduces the model mismatch caused by the second mode.

The result is that once the model mismatch has been reduced, the original data points become effectively noise free. The large amount of noise originally associated with these data points was mainly caused by model mismatch. Once we have reduced this mismatch, we can identify the system very accurately. Indeed, after 6 iterations, the bounds on all of the parameters are very small.

6.4 Summary

This chapter has demonstrated the effectiveness of the algorithms presented in this thesis. The first two examples showed that the identification algorithm can determine accurate interval ranges for the parameters from a small number of data points. These points were chosen where the signal to noise ratio was highest.

Several examples of the input design methodology were shown. The algorithm was quickly able to reduce the parametric uncertainty to where we have met our robust performance goals. The inputs were chosen where the signal to noise ratio was highest, which is typically at the natural frequencies of the structure. This approach to input design was therefore shown to be effective for the examples presented.

We also examined the case when the main source of noise was model mismatch. In this case, the assumptions of the input design methodology were violated. This is not a problem if we are able to choose inputs which reduced the model mismatch. Otherwise we may not be able to achieve the performance goals.

Chapter 7

Conclusions

7.1 Summary and Contributions

This work has been concerned with the identification of flexible structures. We chose a parameterization of our system which has been demonstrated in previous work to be numerically stable. We then wanted to determine a nominal model of the system, plus a description of the uncertainty which is appropriate for robust control.

We began by looking at how we could determine a model of our system from a set of input–output data. This data is assumed to be corrupted frequency domain data, with a known bound on the noise. We wanted to determine an interval range for each parameter in our model such that we are guaranteed the true model is in the resulting model set. We constructed an algorithm such that this guarantee can be made subject to the validity of our a priori assumptions.

The novelty of the resulting algorithm was the ability to use a model which is not linear in the unknown parameters. The desired optimization problems are therefore nonlinear. The algorithm avoids this difficulty by a form of "successive linearization." The system is split into a set of one mode problems. Each of the one mode problems is then solved by embedding the nonlinear optimization problem into a convex space. We can then use convex programming to solve the resulting problem.

We have examined the conservatism of this algorithm as opposed to a method which could solve the original nonlinear optimization problems. There are two sources

of conservatism. The first source is from splitting the problem into a set of one mode problems. This conservatism can be easily measured, and should be checked to ensure it becomes small for the problem under consideration. The second source of conservatism arises from embedding our nonlinear optimization problems into a convex space. It was shown how to adapt the identification algorithm to make this conservatism smaller than any specified tolerance by solving a finite number of optimization problems.

We then examined the problem of how to choose an input to apply to the system. Instead of choosing an input based solely on some measure of the parameter uncertainty, we examined how the inputs impact the resulting closed loop performance. The sensitivity of the performance to the inputs was estimated, and the input chosen was the one which had the highest sensitivity.

To calculate this sensitivity, two new quantities were introduced. The first is the sensitivity of the performance to the size of the uncertainties in the system. Using the standard upper bound to mixed μ as our measure of robust performance, we developed a computationally efficient method to calculate this sensitivity. The computation required was an eigenvalue decomposition. Since $\overline{\mu}$ is a function of the closed loop system, a control design methodology must be included in the algorithm. The philosophy of this work was to allow any desired methodology.

The second quantity we needed to calculate was the expected effect of the inputs on the uncertainties. A computationally efficient method to estimate this quantity was described. Using a certainty equivalence assumption, we were able to simulate the system based upon our nominal model to determine the expected output for each input. We then estimated the improvement in the parameter bounds for each of these input–output pairs. This method required little knowledge of the noise entering the system.

We then combined the identification and input design methodologies to create an iterative algorithm. Some basic convergence properties of this algorithm were examined. To guarantee that the robust performance was monotonically improving, some limitations on the control methodology were imposed, and we showed how to

adapt any control methodology to satisfy these requirements. We also examined the conditions under which the specified algorithm would converge to where we have met the robust performance goals. The limitations of the algorithm were discussed.

Several examples were then presented. We first demonstrated the ability of the identification algorithm to determine accurate parameter intervals from a small set of data. The need to use physical insight in parameterizing the system was highlighted. We then examined the ability of the iterative algorithm for identification, control design, and input design to quickly improve the performance in our system. We began with only 15 data points. In most of the examples, we were able to achieve the robust performance goals by adding only a small number of new data points. The ability of the input design algorithm to incorporate all of the available information to determine the most appropriate input was demonstrated.

7.2 Future Work

The algorithms developed in this work were aimed specifically at lightly damped structures. We assumed that all of the modes of the system were lightly damped, and that there was some separation between the modes. One area of future research would be to extend the results presented here to a more general class of systems. For systems with damped modes, we may be able to use a parameterization which is not as numerically stable. Thus, we would identify the parameters of more than one mode at a time. Other parameterizations may be needed for systems with closely spaced or repeated modes. We expect a tradeoff between the numerical stability of the parameterization and the number of modes we identify at one time.

The identification algorithm was based upon a set of a priori information. This information includes a bound on the noise, and a rough estimate of specific parameters. We did not examine the effect of having a small set of data points which violated the a priori assumptions. We would need a method to reject data points when appropriate. This would typically happen when the algorithm determines that there does not exist a set of parameters which is compatible with the data and the a priori assumptions.

We would either need to change our assumptions, or remove a small number of data points.

Another one of the underlying assumptions in this work is that intervals are the best way to describe parameter uncertainty for the purpose of control. The set of parameters consistent with our data might better be described by using a more complicated shape such as a polygon, or some other description which can provide a tighter bound on the set. Several of these methods have been investigated by other researchers, as indicated earlier. However, it remains an open question how we would design or analyze a control system with such a description of the uncertainty. An alternate method to determine the measure of robust performance may be needed.

The philosophy of the iterative input design algorithm presented here was to allow any control design methodologies, and to use as little knowledge of the noise as possible. By using a specific control methodology, or by having more information on the noise, we may be able to make stronger convergence arguments. For instance, it may be possible to estimate the probability distribution of the noise, and as a result achieve more accurate estimates of the effect of the inputs on the uncertainty. Of course, we need to be concerned with the resulting computational burden this extra computation may impose.

Another possible extension is to multi–input multi–output systems. There is no conceptual change in extending the results presented here to MIMO systems. One could identify the parameters of each input–output pair separately, and combine the results. No change is necessary in the input design algorithm, other than that we would need to consider the inputs from a MIMO viewpoint. The difficulty is that we would need to create a state space model of the system including the uncertainty. It is much harder to do this from a pole–residue model than it is for single–input single–output systems. A discussion of some of the relevant issues can be found in appendix A.

Appendix A

Parameterization of MIMO Systems

One of the possible extensions to the algorithms presented in this work is to multi–input multi–output systems. In this appendix, we will show that compared to single–input single–output systems, it is much more difficult to transform a model of a MIMO system which includes parametric uncertainty into a state space representation. A state space representation is desired because most robust control analysis and synthesis tools require such a representation of the system.

We begin by assuming we have a second order MIMO system, with no uncertainty. In the frequency domain, we would have the following representation.

$$G(s) = \frac{B_1 s + B_0}{s^2 + a_1 s + a_0} \tag{A.1}$$

Since this is a multi–input multi–output system, B_1 and B_0 are matrices.

For notational convenience, we will define $\psi(s)$ as the characteristic polynomial.

$$\psi(s) = s^2 + a_1 s + a_0 \tag{A.2}$$

We would like to determine when we can write this system in a state space representation with only two states. If the system has two nonrepeated poles, and no pole zero

cancellations, a state space representation can be found with only two states. Such a representation can always be transformed into the following state space matrices:

$$A = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} \qquad B = \begin{bmatrix} b_1^T \\ \tilde{b}_0^T \end{bmatrix} \tag{A.3}$$

$$C = \begin{bmatrix} c_0 & c_1 \end{bmatrix} \tag{A.4}$$

For notational convenience, we will define $b_0 = \tilde{b}_0 + a_1 b_1$. In doing so, we have not lost any generality in our representation. The state space representation is now described by the following equations.

$$A = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} \qquad B = \begin{bmatrix} b_1^T \\ b_0^T - a_1 b_1^T \end{bmatrix} \tag{A.5}$$

$$C = \begin{bmatrix} c_0 & c_1 \end{bmatrix} \tag{A.6}$$

The question we would like to address is when can our system (A.1) be put into this state space representation. In other words, when can we say the system (A.1) has two nonrepeated poles? Whenever we can make this statement, (A.5)–(A.6) is an appropriate state space representation.

As an example, consider the case when $B_1 = 0$. As we will see, if we have

$$G(s) = \frac{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}{s^2 + a_1 s + a_0} \tag{A.7}$$

then any state space description for this system must have at least four states. However, if we have

$$G(s) = \frac{\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}}{s^2 + a_1 s + a_0} \tag{A.8}$$

then we can create a state space description with two states.

The following result was shown in [21]. The proof is presented here for completeness.

Lemma A.1 *The system*

$$G(s) = \frac{B_1 s + B_0}{s^2 + a_1 s + a_0}$$

has a state space description given by (A.5)-(A.6) if and only if the following equalities hold.

$$B_1 s + B_0 = c(s)b(s) \ mod \ \psi(s) \tag{A.9}$$

$$c(s) = c_1 s + c_0 \tag{A.10}$$

$$b(s) = b_1^T s + b_0^T \tag{A.11}$$

where "mod $\psi(s)$" represents the residue after division by $s^2 + a_1 s + a_0$.

Proof: Straightforward algebra shows that

$$C(sI - A)^{-1}B = \begin{bmatrix} c_0 & c_1 \end{bmatrix} \begin{bmatrix} s & -1 \\ a_0 & s + a_1 \end{bmatrix}^{-1} \begin{bmatrix} b_1^T \\ b_0^T - a_1 b_1^T \end{bmatrix} \tag{A.12}$$

$$= \frac{c_0 b_1^T s + c_0 b_0^T - a_0 c_1 b_1^T + s c_1 b_0^T - s a_1 c_1 b_1^T}{s^2 + a_1 s + a_0} \tag{A.13}$$

Thus, this is a representation for our system if and only if

$$B_1 s + B_0 = (c_0 b_1^T + c_1 b_0^T - a_1 c_1 b_1^T)s + (c_0 b_0^T - a_0 c_1 b_1^T) \tag{A.14}$$

$$= (c_0 b_1^T + c_1 b_0^T - a_1 c_1 b_1^T)s + (c_0 b_0^T - a_0 c_1 b_1^T) \tag{A.15}$$

$$+ c_1 b_1^T \psi(s) \bmod \psi(s)$$

$$= c_1 b_1^T s^2 + (c_0 b_1^T + c_1 b_0^T)s + c_0 b_0^T \bmod \psi(s) \tag{A.16}$$

$$= (c_1 s + c_0)(b_1^T s + b_0^T) \bmod \psi(s) \tag{A.17}$$

■

This first result gives a characterization of the types of systems which can be transformed into the desired state space representation. Essentially, it characterizes

the second order transfer functions which can be represented as a system with two nonrepeated poles. A more useful characterization is given in the following lemma.

Lemma A.2 *The system*

$$G(s) = \frac{B_1 s + B_0}{s^2 + a_1 s + a_0}$$

has a state space description given by (A.5)-(A.6) if and only if

$$B_1 s + B_0 \Big|_{\psi(s)=0} \text{ is rank 1} \tag{A.18}$$

Proof: First, assume that (A.5)–(A.6) is an appropriate state space representation. Then we know from lemma A.1

$$B_1 s + B_0 = c(s)b(s) \bmod \psi(s) \tag{A.19}$$

where $c(s)$ and $b(s)$ were defined in lemma A.1.

For all s, $c(s)b(s)$ is rank 1. Multiplying this out, and collecting terms, we see the following quantity has rank 1.

$$(c_0 b_1^T + c_1 b_0^T - a_1 c_1 b_1^T)s + (c_0 b_0^T - a_0 c_1 b_1^T) + c_1 b_1^T \psi(s) \tag{A.20}$$

The first two terms of this expression equal $c(s)b(s) \bmod \psi(s)$. We therefore know that

$$c(s)b(s) \bmod \psi(s) \quad + c_1 b_1^T \psi(s) \text{ is rank 1} \tag{A.21}$$

Therefore, we have

$$B_1 s + B_0 + c_1 b_1^T \psi(s) \text{ is rank 1 } \forall s \tag{A.22}$$

The result follows by evaluating at $\psi(s) = 0$.

Conversely, let us assume that $B_1 s + B_0 \Big|_{\psi(s)=0}$ is rank 1. From [21], we can always write this residue in terms of a sum of dyads as follows.

$$B_1 s + B_0 = \sum_{i=1}^{h} c_i(s) b_i(s) \bmod \psi(s) \tag{A.23}$$

$$c_i(s) = c_{1i}s + c_{0i} \tag{A.24}$$

$$b_i(s) = b_{1i}^T s + b_{0i}^T \tag{A.25}$$

We therefore have the following quantity as being rank 1.

$$\sum_{i=1}^{h} c_i(s)b_i(s) \bmod \psi(s) \Big|_{\psi(s)=0} \tag{A.26}$$

Following the same steps as before, we can conclude that

$$\sum_{i=1}^{h} c_i(s)b_i(s) \Big|_{\psi(s)=0} \quad \text{is rank 1} \tag{A.27}$$

Without loss in generality, we can assume that

$$c_i(s)b_i(s) \Big|_{\psi(s)=0} = 0 \qquad i = 2, ..., h \tag{A.28}$$

Therefore we can write

$$B_1 s + B_0 = c_1(s)b_1(s) \bmod \psi(s) \tag{A.29}$$

Using lemma A.1 completes the proof. ■

The conclusion from this lemma is that if our frequency domain description has two nonrepeated poles, then the residue evaluated at the location of the poles must be rank 1. In general, if the residue is rank h at the location of the poles, then these poles are repeated h times. We would therefore need a state space description which has $2h$ states instead of 2 states.

Let us examine what this means for describing parametric uncertainty. For simplicity, let us assume that we have a second order system whose poles are known exactly. Furthermore, let us assume that the residue consists only of a constant term. We thus have the following system.

$$G(s) = \frac{B_0 + \tilde{B}_0}{s^2 + a_1 s + a_0} \tag{A.30}$$

Here, B_0 is the nominal value of the residue, and $\tilde{B}_0$ is the uncertainty. Let us assume that we know the pole is nonrepeated. Thus, B_0 must be rank 1. Furthermore, we need to have $B_0 + \tilde{B}_0$ be rank 1 for all possible values of the uncertainty. If it was not rank 1, then the uncertainty would be increasing the order of the system!

In order to guarantee that the residue remains rank 1 in the face of uncertainty, we would need to write

$$B_0 + \tilde{B}_0 = (c_0 + \Delta_1 \tilde{c}_0)(b_0 + \Delta_2 \tilde{b}_0) \tag{A.31}$$

Here, Δ_1 and Δ_2 are the uncertainty blocks, with $\tilde{c}_0$ and $\tilde{b}_0$ representing the direction the uncertainty enters the residue.

The difficulty with this representation is that it is not easy to transform this description of the uncertainty into parameter intervals. In fact, requiring this form for the uncertainty is a nonconvex constraint. The problem compounds when there is a first order term in the residue, and the location of the poles is uncertain. We would need to guarantee that the residue remains rank 1 at the location of the poles in the face of uncertainty in both the residues and the pole locations.

The conclusion is that the identification of parametric uncertainty becomes much more difficult for MIMO systems. We would need either to use alternate parameterizations, or to be otherwise able to guarantee that the uncertainty does not increase the order of the system. It remains an open question on how this can be done from a frequency domain perspective.

Appendix B

Parameter Bounds and Sensitivities

In this appendix, the parameter intervals resulting from the iterative input design algorithm are listed. For each input, the lower and upper bounds are listed. Also listed are the calculated μ–sensitivites.

B.1 First example of Input Design

In this section, we list the parameter bounds and μ–sensitivities for the example of section 6.3.2. It is interesting to note that when we apply an input to improve the parameter intervals of one mode, all the parameter intervals can decrease. For instance, the lower bound to a_{12} is greatly improved after applying an input at 1.15 rad/sec. The reason is that the model mismatch from mode 1 was large, causing the effective noise bound at mode 2 to be high. Once we apply an input at 1.15 rad/sec, this model mismatch decreases. The noise bound at mode 2 decreases, allowing a more accurate identification of the parameters in this mode.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	2.0911×10^{-9}	1.2140×10^{-1}	3.5078×10^{-2}
a_{01}	8.8827×10^{-1}	1.2333	2.7621×10^{-2}
b_{01}	2.7028	7.6251	8.0642×10^{-1}
a_{12}	4.5511×10^{-9}	5.3030×10^{-1}	3.2849×10^{-3}
a_{02}	3.3851×10^{1}	4.3162×10^{1}	1.7235×10^{-2}
b_{02}	-1.4572	7.6061	7.5447×10^{-1}
a_{13}	1.6299×10^{-9}	2.5572	6.2945×10^{-4}
a_{03}	2.6725×10^{2}	3.5747×10^{2}	7.1178×10^{-3}
b_{03}	-7.8492	1.5549×10^{1}	1.9178×10^{-1}

Table B-1: The lower bounds, upper bounds, and μ–sensitivities after initial data points.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	2.3388×10^{-9}	1.0803×10^{-1}	3.1556×10^{-2}
a_{01}	9.1603×10^{-1}	1.2083	2.6370×10^{-2}
b_{01}	3.1475	7.3021	8.5018×10^{-1}
a_{12}	5.2126×10^{-9}	2.9791×10^{-1}	1.5692×10^{-2}
a_{02}	3.8214×10^{1}	4.0165×10^{1}	3.3262×10^{-2}
b_{02}	2.6454	5.3482	3.5126×10^{-1}
a_{13}	2.4323×10^{-9}	1.4552	6.0251×10^{-5}
a_{03}	2.8427×10^{2}	3.3174×10^{2}	5.9931×10^{-4}
b_{03}	-1.4172	9.3018	1.0711×10^{-1}

Table B-2: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 6.48 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	4.2424×10^{-9}	5.9748×10^{-2}	1.7408×10^{-2}
a_{01}	9.6593×10^{-1}	1.0661	9.8363×10^{-3}
b_{01}	3.4690	5.2415	4.7898×10^{-1}
a_{12}	3.5359×10^{-2}	2.4620×10^{-1}	1.2230×10^{-2}
a_{02}	3.8567×10^{1}	3.9892×10^{1}	2.6848×10^{-2}
b_{02}	3.0733	4.9074	2.7978×10^{-1}
a_{13}	3.1971×10^{-9}	1.0474	1.0600×10^{-4}
a_{03}	2.9249×10^{2}	3.2448×10^{2}	1.0762×10^{-3}
b_{03}	-1.1083×10^{-2}	7.6161	9.3782×10^{-2}

Table B-3: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 1.15 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	4.7365×10^{-9}	5.7917×10^{-2}	3.0749×10^{-2}
a_{01}	9.6762×10^{-1}	1.0645	2.0218×10^{-2}
b_{01}	3.4987	5.2119	3.0615×10^{-1}
a_{12}	8.0641×10^{-2}	2.0884×10^{-1}	6.8280×10^{-4}
a_{02}	3.8819×10^{1}	3.9646×10^{1}	1.6809×10^{-3}
b_{02}	3.4239	4.5696	1.8212×10^{-2}
a_{13}	8.4559×10^{-2}	6.4925×10^{-1}	4.7136×10^{-5}
a_{03}	3.0265×10^{2}	3.1263×10^{2}	3.2356×10^{-4}
b_{03}	2.7391	5.4602	5.0543×10^{-3}

Table B-4: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 16.96 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	5.0456×10^{-9}	5.2191×10^{-2}	2.2463×10^{-2}
a_{01}	9.7113×10^{-1}	1.0310	5.4597×10^{-3}
b_{01}	3.5514	4.6660	2.9543×10^{-1}
a_{12}	8.2541×10^{-2}	2.0460×10^{-1}	1.1113×10^{-2}
a_{02}	3.8851×10^{1}	3.9638×10^{1}	1.5936×10^{-2}
b_{02}	3.4482	4.5374	1.5601×10^{-1}
a_{13}	1.0453×10^{-1}	6.2602×10^{-1}	2.9560×10^{-4}
a_{03}	3.0305×10^{2}	3.1231×10^{2}	1.1117×10^{-3}
b_{03}	2.8288	5.3569	3.0237×10^{-2}

Table B-5: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 1.09 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	4.8585×10^{-3}	4.3214×10^{-2}	1.6367×10^{-2}
a_{01}	9.8127×10^{-1}	1.0182	3.4144×10^{-3}
b_{01}	3.7303	4.5836	2.3284×10^{-1}
a_{12}	8.3685×10^{-2}	2.0285×10^{-1}	1.0891×10^{-2}
a_{02}	3.8857×10^{1}	3.9626×10^{1}	1.6077×10^{-2}
b_{02}	3.4608	4.5270	1.5747×10^{-1}
a_{13}	1.0870×10^{-1}	6.2251×10^{-1}	2.9174×10^{-4}
a_{03}	3.0311×10^{2}	3.1223×10^{2}	1.1266×10^{-3}
b_{03}	2.8493	5.3391	3.0713×10^{-2}

Table B-6: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at .93 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	4.9140×10^{-3}	4.3165×10^{-2}	1.7787×10^{-2}
a_{01}	9.8135×10^{-1}	1.0181	3.1973×10^{-3}
b_{01}	3.7321	4.5823	2.2599×10^{-1}
a_{12}	9.7232×10^{-2}	1.5475×10^{-1}	6.7511×10^{-3}
a_{02}	3.9032×10^{1}	3.9412×10^{1}	8.8697×10^{-3}
b_{02}	3.5933	4.4337	1.3021×10^{-1}
a_{13}	1.1183×10^{-1}	6.1975×10^{-1}	3.3715×10^{-4}
a_{03}	3.0316×10^{2}	3.1217×10^{2}	1.1204×10^{-3}
b_{03}	2.8650	5.3251	3.0814×10^{-2}

Table B-7: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 6.14 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.7257×10^{-2}	2.7399×10^{-2}	4.6538×10^{-3}
a_{01}	9.9722×10^{-1}	1.0104	1.2098×10^{-3}
b_{01}	3.8041	4.3815	1.7016×10^{-1}
a_{12}	9.7941×10^{-2}	1.5407×10^{-1}	6.7495×10^{-3}
a_{02}	3.9037×10^{1}	3.9408×10^{1}	9.6238×10^{-3}
b_{02}	3.6063	4.4279	1.4150×10^{-1}
a_{13}	1.1511×10^{-1}	6.1565×10^{-1}	3.3976×10^{-4}
a_{03}	3.0324×10^{2}	3.1212×10^{2}	1.2269×10^{-3}
b_{03}	2.8807	5.3061	3.3811×10^{-2}

Table B-8: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at .97 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.7257×10^{-2}	2.7399×10^{-2}	3.7977×10^{-3}
a_{01}	9.9726×10^{-1}	1.0103	1.4364×10^{-3}
b_{01}	3.8063	4.3808	2.1780×10^{-1}
a_{12}	1.1192×10^{-1}	1.4985×10^{-1}	4.9294×10^{-3}
a_{02}	3.9149×10^{1}	3.9369×10^{1}	9.1058×10^{-3}
b_{02}	3.8021	4.4279	1.5669×10^{-1}
a_{13}	1.1843×10^{-1}	6.1378×10^{-1}	3.2123×10^{-4}
a_{03}	3.0327×10^{2}	3.1205×10^{2}	1.6896×10^{-3}
b_{03}	2.8977	5.2952	4.6565×10^{-2}

Table B-9: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 6.37 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.7307×10^{-2}	2.2188×10^{-2}	2.4443×10^{-3}
a_{01}	9.9733×10^{-1}	1.0032	5.9608×10^{-4}
b_{01}	3.8091	4.0513	8.4121×10^{-2}
a_{12}	1.1197×10^{-1}	1.4896×10^{-1}	5.5060×10^{-3}
a_{02}	3.9154×10^{1}	3.9368×10^{1}	6.9287×10^{-3}
b_{02}	3.8110	4.4200	1.2416×10^{-1}
a_{13}	1.9632×10^{-1}	5.7746×10^{-1}	3.3100×10^{-4}
a_{03}	3.0427×10^{2}	3.1104×10^{2}	1.2031×10^{-3}
b_{03}	3.1714	5.0556	3.1107×10^{-2}

Table B-10: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 1.03 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.7313×10^{-2}	2.2183×10^{-2}	2.5130×10^{-3}
a_{01}	9.9733×10^{-1}	1.0032	5.6864×10^{-4}
b_{01}	3.8095	4.0511	8.0198×10^{-2}
a_{12}	1.1227×10^{-1}	1.4896×10^{-1}	5.6354×10^{-3}
a_{02}	3.9155×10^{1}	3.9366×10^{1}	6.5573×10^{-3}
b_{02}	3.8124	4.4200	1.1862×10^{-1}
a_{13}	2.4905×10^{-1}	4.8879×10^{-1}	2.2847×10^{-4}
a_{03}	3.0600×10^{2}	3.0975×10^{2}	6.7900×10^{-4}
b_{03}	3.3665	4.7721	2.2302×10^{-2}

Table B-11: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 17.28 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.7316×10^{-2}	2.2181×10^{-2}	2.5646×10^{-3}
a_{01}	9.9734×10^{-1}	1.0032	5.5364×10^{-4}
b_{01}	3.8097	4.0510	7.8029×10^{-2}
a_{12}	1.1241×10^{-1}	1.4755×10^{-1}	5.5185×10^{-3}
a_{02}	3.9183×10^{1}	3.9365×10^{1}	5.5386×10^{-3}
b_{02}	3.8157	4.3398	9.9833×10^{-2}
a_{13}	2.4905×10^{-1}	4.8749×10^{-1}	2.3178×10^{-4}
a_{03}	3.0602×10^{2}	3.0974×10^{2}	6.5673×10^{-4}
b_{03}	3.3676	4.7721	2.1723×10^{-2}

Table B-12: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 6.15 rad/sec.

B.2 Example with SNR of 10

In this section, we list the parameter bounds and μ–sensitivities for the example of section 6.3.3, where the bound on the additive noise was set at 10% of the true system.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	2.3512×10^{-9}	5.2417×10^{-2}	1.8664×10^{-2}
a_{01}	9.6671×10^{-1}	1.0554	8.6960×10^{-3}
b_{01}	3.7179	4.6707	2.5264×10^{-1}
a_{12}	1.0730×10^{-9}	2.6803×10^{-1}	1.1883×10^{-2}
a_{02}	3.7486×10^{1}	4.0497×10^{1}	3.7285×10^{-2}
b_{02}	2.3832	5.0955	3.3311×10^{-1}
a_{13}	1.9081×10^{-9}	9.1498×10^{-1}	2.9885×10^{-4}
a_{03}	2.9682×10^{2}	3.1816×10^{2}	1.9234×10^{-3}
b_{03}	2.0176	6.3354	4.6831×10^{-2}

Table B-13: The lower bounds, upper bounds, and μ–sensitivities after initial data points.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	3.3388×10^{-9}	5.0199×10^{-2}	2.0357×10^{-2}
a_{01}	9.7153×10^{-1}	1.0466	7.6136×10^{-3}
b_{01}	3.7957	4.4936	2.1219×10^{-1}
a_{12}	8.0302×10^{-2}	2.0086×10^{-1}	1.0630×10^{-2}
a_{02}	3.8845×10^{1}	3.9627×10^{1}	1.8067×10^{-2}
b_{02}	3.4522	4.4892	1.6969×10^{-1}
a_{13}	3.3509×10^{-9}	7.5742×10^{-1}	3.9745×10^{-4}
a_{03}	2.9990×10^{2}	3.1483×10^{2}	1.9475×10^{-3}
b_{03}	2.6989	5.7228	4.1063×10^{-2}

Table B-14: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 6.48 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.2638×10^{-2}	2.2547×10^{-2}	3.8579×10^{-3}
a_{01}	9.9382×10^{-1}	1.0055	1.1820×10^{-3}
b_{01}	3.8504	4.3295	1.5146×10^{-1}
a_{12}	8.2148×10^{-2}	1.9882×10^{-1}	1.0181×10^{-2}
a_{02}	3.8864×10^{1}	3.9616×10^{1}	1.8117×10^{-2}
b_{02}	3.4721	4.4526	1.6760×10^{-1}
a_{13}	3.8572×10^{-2}	7.1225×10^{-1}	3.5837×10^{-4}
a_{03}	3.0073×10^{2}	3.1406×10^{2}	1.8609×10^{-3}
b_{03}	2.8265	5.5758	3.9101×10^{-2}

Table B-15: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 1.02 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.2741×10^{-2}	2.2497×10^{-2}	4.1621×10^{-3}
a_{01}	9.9387×10^{-1}	1.0055	1.0757×10^{-3}
b_{01}	3.8520	4.3277	1.3736×10^{-1}
a_{12}	8.3899×10^{-2}	1.9664×10^{-1}	1.0803×10^{-2}
a_{02}	3.8868×10^{1}	3.9608×10^{1}	1.6318×10^{-2}
b_{02}	3.4819	4.4206	1.4703×10^{-1}
a_{13}	2.2377×10^{-1}	5.0279×10^{-1}	1.8799×10^{-4}
a_{03}	3.0571×10^{2}	3.1008×10^{2}	6.5173×10^{-4}
b_{03}	3.2925	4.8550	2.0611×10^{-2}

Table B-16: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 17.25 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.2741×10^{-2}	2.2484×10^{-2}	3.4581×10^{-3}
a_{01}	9.9389×10^{-1}	1.0055	1.2088×10^{-3}
b_{01}	3.8547	4.3263	1.7231×10^{-1}
a_{12}	9.9619×10^{-2}	1.5192×10^{-1}	6.0625×10^{-3}
a_{02}	3.9173×10^{1}	3.9384×10^{1}	7.7356×10^{-3}
b_{02}	3.4976	4.4111	2.1979×10^{-1}
a_{13}	2.2377×10^{-1}	5.0007×10^{-1}	1.9698×10^{-4}
a_{03}	3.0571×10^{2}	3.1008×10^{2}	9.2966×10^{-4}
b_{03}	3.2925	4.8533	2.9421×10^{-2}

Table B-17: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 6.28 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.2741×10^{-2}	2.2480×10^{-2}	3.5663×10^{-3}
a_{01}	9.9390×10^{-1}	1.0055	1.2046×10^{-3}
b_{01}	3.8554	4.3259	1.7139×10^{-1}
a_{12}	1.0382×10^{-1}	1.5192×10^{-1}	5.7956×10^{-3}
a_{02}	3.9173×10^{1}	3.9379×10^{1}	7.5990×10^{-3}
b_{02}	3.5260	4.4111	2.1256×10^{-1}
a_{13}	2.2399×10^{-1}	4.9793×10^{-1}	2.0175×10^{-4}
a_{03}	3.0572×10^{2}	3.1007×10^{2}	9.2345×10^{-4}
b_{03}	3.2983	4.8533	2.9225×10^{-2}

Table B-18: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 6.16 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.5197×10^{-2}	2.2376×10^{-2}	2.6152×10^{-3}
a_{01}	9.9595×10^{-1}	1.0030	7.2796×10^{-4}
b_{01}	3.8689	4.3255	1.6550×10^{-1}
a_{12}	1.0382×10^{-1}	1.5192×10^{-1}	5.7544×10^{-3}
a_{02}	3.9173×10^{1}	3.9378×10^{1}	7.5538×10^{-3}
b_{02}	3.5269	4.4082	2.1078×10^{-1}
a_{13}	2.2407×10^{-1}	4.9761×10^{-1}	1.9990×10^{-4}
a_{03}	3.0572×10^{2}	3.1007×10^{2}	9.1750×10^{-4}
b_{03}	3.2989	4.8527	2.9083×10^{-2}

Table B-19: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at .99 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.5197×10^{-2}	2.2353×10^{-2}	2.6910×10^{-3}
a_{01}	9.9595×10^{-1}	1.0030	7.1987×10^{-4}
b_{01}	3.8690	4.3253	1.6353×10^{-1}
a_{12}	1.0382×10^{-1}	1.4934×10^{-1}	5.6285×10^{-3}
a_{02}	3.9194×10^{1}	3.9378×10^{1}	6.6948×10^{-3}
b_{02}	3.5269	4.3642	1.9817×10^{-1}
a_{13}	2.2429×10^{-1}	4.9738×10^{-1}	2.0598×10^{-4}
a_{03}	3.0573×10^{2}	3.1006×10^{2}	9.0556×10^{-4}
b_{03}	3.3002	4.8514	2.8716×10^{-2}

Table B-20: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 6.13 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.5197×10^{-2}	2.2353×10^{-2}	2.7357×10^{-3}
a_{01}	9.9595×10^{-1}	1.0030	7.3517×10^{-4}
b_{01}	3.8713	4.3241	1.6569×10^{-1}
a_{12}	1.0464×10^{-1}	1.4926×10^{-1}	5.7716×10^{-3}
a_{02}	3.9195×10^{1}	3.9378×10^{1}	7.0247×10^{-3}
b_{02}	3.6376	4.3633	1.7531×10^{-1}
a_{13}	2.2478×10^{-1}	4.9656×10^{-1}	2.0803×10^{-4}
a_{03}	3.0573×10^{2}	3.1005×10^{2}	9.1881×10^{-4}
b_{03}	3.3030	4.8473	2.9182×10^{-2}

Table B-21: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 6.37 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.5971×10^{-2}	2.1649×10^{-2}	2.1606×10^{-3}
a_{01}	9.9821×10^{-1}	1.0030	4.9678×10^{-4}
b_{01}	3.8728	4.3234	1.6430×10^{-1}
a_{12}	1.0464×10^{-1}	1.4925×10^{-1}	5.7385×10^{-3}
a_{02}	3.9195×10^{1}	3.9378×10^{1}	6.9985×10^{-3}
b_{02}	3.6376	4.3627	1.7459×10^{-1}
a_{13}	2.2478×10^{-1}	4.9652×10^{-1}	2.0680×10^{-4}
a_{03}	3.0573×10^{2}	3.1005×10^{2}	9.1483×10^{-4}
b_{03}	3.3031	4.8471	2.9078×10^{-2}

Table B-22: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at .99 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.7312×10^{-2}	2.1530×10^{-2}	1.5998×10^{-3}
a_{01}	9.9821×10^{-1}	1.0018	3.7120×10^{-4}
b_{01}	3.8729	4.3221	1.6347×10^{-1}
a_{12}	1.0464×10^{-1}	1.4924×10^{-1}	5.7152×10^{-3}
a_{02}	3.9195×10^{1}	3.9378×10^{1}	6.9841×10^{-3}
b_{02}	3.6376	4.3627	1.7426×10^{-1}
a_{13}	2.2479×10^{-1}	4.9651×10^{-1}	2.0593×10^{-4}
a_{03}	3.0573×10^{2}	3.1005×10^{2}	9.1268×10^{-4}
b_{03}	3.3032	4.8470	2.9024×10^{-2}

Table B-23: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 1.00 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.7312×10^{-2}	2.1522×10^{-2}	1.5967×10^{-3}
a_{01}	9.9823×10^{-1}	1.0018	3.6857×10^{-4}
b_{01}	3.8729	4.3221	1.6343×10^{-1}
a_{12}	1.0464×10^{-1}	1.4924×10^{-1}	5.7149×10^{-3}
a_{02}	3.9195×10^{1}	3.9378×10^{1}	6.9834×10^{-3}
b_{02}	3.6376	4.3624	1.7417×10^{-1}
a_{13}	2.2479×10^{-1}	4.9650×10^{-1}	2.0592×10^{-4}
a_{03}	3.0573×10^{2}	3.1005×10^{2}	9.1257×10^{-4}
b_{03}	3.3032	4.8470	2.9021×10^{-2}

Table B-24: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 6.31 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.7312×10^{-2}	2.1520×10^{-2}	1.5963×10^{-3}
a_{01}	9.9823×10^{-1}	1.0018	3.6849×10^{-4}
b_{01}	3.8729	4.3221	1.6343×10^{-1}
a_{12}	1.0464×10^{-1}	1.4924×10^{-1}	5.7149×10^{-3}
a_{02}	3.9195×10^{1}	3.9378×10^{1}	6.9834×10^{-3}
b_{02}	3.6376	4.3624	1.7417×10^{-1}
a_{13}	2.2479×10^{-1}	4.9650×10^{-1}	2.0592×10^{-4}
a_{03}	3.0573×10^{2}	3.1005×10^{2}	9.1256×10^{-4}
b_{03}	3.3032	4.8470	2.9021×10^{-2}

Table B-25: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at .99 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.7312×10^{-2}	2.1520×10^{-2}	1.6355×10^{-3}
a_{01}	9.9823×10^{-1}	1.0018	3.6148×10^{-4}
b_{01}	3.8730	4.3221	1.6029×10^{-1}
a_{12}	1.0464×10^{-1}	1.4924×10^{-1}	5.8623×10^{-3}
a_{02}	3.9195×10^{1}	3.9378×10^{1}	6.8555×10^{-3}
b_{02}	3.6377	4.3621	1.7090×10^{-1}
a_{13}	2.4425×10^{-1}	4.5244×10^{-1}	1.6247×10^{-4}
a_{03}	3.0637×10^{2}	3.0927×10^{2}	6.0473×10^{-4}
b_{03}	3.3072	4.8412	2.8373×10^{-2}

Table B-26: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 17.69 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.7312×10^{-2}	2.1520×10^{-2}	1.6325×10^{-3}
a_{01}	9.9879×10^{-1}	1.0018	3.0382×10^{-4}
b_{01}	3.8731	4.3221	1.5996×10^{-1}
a_{12}	1.0464×10^{-1}	1.4923×10^{-1}	5.8501×10^{-3}
a_{02}	3.9195×10^{1}	3.9378×10^{1}	6.8422×10^{-3}
b_{02}	3.6378	4.3617	1.7047×10^{-1}
a_{13}	2.4429×10^{-1}	4.5048×10^{-1}	1.6074×10^{-4}
a_{03}	3.0638×10^{2}	3.0927×10^{2}	6.0260×10^{-4}
b_{03}	3.3112	4.8403	2.8231×10^{-2}

Table B-27: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 1.01 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.7392×10^{-2}	2.1520×10^{-2}	1.6016×10^{-3}
a_{01}	9.9879×10^{-1}	1.0018	3.0349×10^{-4}
b_{01}	3.8731	4.3221	1.5992×10^{-1}
a_{12}	1.0464×10^{-1}	1.4923×10^{-1}	5.8508×10^{-3}
a_{02}	3.9195×10^{1}	3.9378×10^{1}	6.8414×10^{-3}
b_{02}	3.6378	4.3616	1.7043×10^{-1}
a_{13}	2.4430×10^{-1}	4.5022×10^{-1}	1.6056×10^{-4}
a_{03}	3.0638×10^{2}	3.0927×10^{2}	6.0193×10^{-4}
b_{03}	3.3115	4.8393	2.8203×10^{-2}

Table B-28: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 1.00 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.7392×10^{-2}	2.1520×10^{-2}	1.6016×10^{-3}
a_{01}	9.9879×10^{-1}	1.0018	3.0343×10^{-4}
b_{01}	3.8731	4.3221	1.5991×10^{-1}
a_{12}	1.0464×10^{-1}	1.4923×10^{-1}	5.8508×10^{-3}
a_{02}	3.9195×10^{1}	3.9378×10^{1}	6.8413×10^{-3}
b_{02}	3.6378	4.3616	1.7042×10^{-1}
a_{13}	2.4431×10^{-1}	4.5018×10^{-1}	1.6054×10^{-4}
a_{03}	3.0638×10^{2}	3.0927×10^{2}	6.0183×10^{-4}
b_{03}	3.3115	4.8391	2.8198×10^{-2}

Table B-29: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 6.36 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.8371×10^{-2}	2.1520×10^{-2}	1.2183×10^{-3}
a_{01}	9.9953×10^{-1}	1.0018	2.2717×10^{-4}
b_{01}	3.8731	4.3221	1.5947×10^{-1}
a_{12}	1.0464×10^{-1}	1.4923×10^{-1}	5.8287×10^{-3}
a_{02}	3.9195×10^{1}	3.9378×10^{1}	6.8220×10^{-3}
b_{02}	3.6378	4.3615	1.6997×10^{-1}
a_{13}	2.4431×10^{-1}	4.5017×10^{-1}	1.5989×10^{-4}
a_{03}	3.0638×10^{2}	3.0927×10^{2}	5.9999×10^{-4}
b_{03}	3.3115	4.8391	2.8125×10^{-2}

Table B-30: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 1.00 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.8371×10^{-2}	2.1520×10^{-2}	1.2180×10^{-3}
a_{01}	9.9953×10^{-1}	1.0018	2.2692×10^{-4}
b_{01}	3.8731	4.3221	1.5946×10^{-1}
a_{12}	1.0464×10^{-1}	1.4923×10^{-1}	5.8273×10^{-3}
a_{02}	3.9195×10^{1}	3.9378×10^{1}	6.8215×10^{-3}
b_{02}	3.6378	4.3615	1.6996×10^{-1}
a_{13}	2.4431×10^{-1}	4.5017×10^{-1}	1.5985×10^{-4}
a_{03}	3.0638×10^{2}	3.0927×10^{2}	5.9992×10^{-4}
b_{03}	3.3115	4.8085	2.7562×10^{-2}

Table B-31: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 17.83 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.8371×10^{-2}	2.1520×10^{-2}	1.2319×10^{-3}
a_{01}	9.9953×10^{-1}	1.0018	2.2588×10^{-4}
b_{01}	3.8731	4.3221	1.5874×10^{-1}
a_{12}	1.0633×10^{-1}	1.4923×10^{-1}	5.6700×10^{-3}
a_{02}	3.9195×10^{1}	3.9378×10^{1}	6.7973×10^{-3}
b_{02}	3.6379	4.3604	1.6901×10^{-1}
a_{13}	2.4431×10^{-1}	4.5017×10^{-1}	1.6167×10^{-4}
a_{03}	3.0638×10^{2}	3.0927×10^{2}	5.9723×10^{-4}
b_{03}	3.3116	4.8085	2.7439×10^{-2}

Table B-32: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 6.31 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.8371×10^{-2}	2.1448×10^{-2}	1.2018×10^{-3}
a_{01}	9.9954×10^{-1}	1.0015	1.9917×10^{-4}
b_{01}	3.8731	4.3196	1.5807×10^{-1}
a_{12}	1.0633×10^{-1}	1.4922×10^{-1}	5.6677×10^{-3}
a_{02}	3.9196×10^{1}	3.9378×10^{1}	6.7722×10^{-3}
b_{02}	3.6410	4.3604	1.6853×10^{-1}
a_{13}	2.4432×10^{-1}	4.5015×10^{-1}	1.6147×10^{-4}
a_{03}	3.0638×10^{2}	3.0927×10^{2}	5.9790×10^{-4}
b_{03}	3.3117	4.8083	2.7474×10^{-2}

Table B-33: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 1.00 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.8371×10^{-2}	2.1437×10^{-2}	1.1975×10^{-3}
a_{01}	9.9954×10^{-1}	1.0015	1.9849×10^{-4}
b_{01}	3.8731	4.3194	1.5802×10^{-1}
a_{12}	1.0633×10^{-1}	1.4922×10^{-1}	5.6674×10^{-3}
a_{02}	3.9196×10^{1}	3.9378×10^{1}	6.7725×10^{-3}
b_{02}	3.6410	4.3604	1.6852×10^{-1}
a_{13}	2.4432×10^{-1}	4.5014×10^{-1}	1.6146×10^{-4}
a_{03}	3.0638×10^{2}	3.0927×10^{2}	5.9794×10^{-4}
b_{03}	3.3117	4.8083	2.7476×10^{-2}

Table B-34: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 6.34 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.8557×10^{-2}	2.1398×10^{-2}	1.1037×10^{-3}
a_{01}	9.9955×10^{-1}	1.0006	1.0807×10^{-4}
b_{01}	3.8743	4.3185	1.5690×10^{-1}
a_{12}	1.0633×10^{-1}	1.4922×10^{-1}	5.6356×10^{-3}
a_{02}	3.9196×10^{1}	3.9378×10^{1}	6.7545×10^{-3}
b_{02}	3.6411	4.3601	1.6806×10^{-1}
a_{13}	2.4433×10^{-1}	4.5013×10^{-1}	1.6052×10^{-4}
a_{03}	3.0638×10^{2}	3.0927×10^{2}	5.9622×10^{-4}
b_{03}	3.3118	4.8082	2.7411×10^{-2}

Table B-35: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 1.00 rad/sec.

B.3 Two Examples with Larger Noise at the Natural Frequencies

This section contains the parameter bounds for the two examples where the additive noise bound was much larger at the natural frequencies of the system than near the zeros. Each example will be put into its own subsection.

B.3.1 Parameter Bounds When We Include Unmodelled Dynamics

a_{11}	5.0508×10^{-9}	1.2619×10^{-1}	3.2196×10^{-2}
a_{01}	7.8212×10^{-1}	1.1802	2.9418×10^{-2}
b_{01}	2.6351	4.9086	3.5682×10^{-1}
a_{12}	4.1845×10^{-9}	7.3355	6.7197×10^{-3}
a_{02}	2.9266×10^{1}	5.3809×10^{1}	1.2886×10^{-1}
b_{02}	-1.0883×10^{1}	1.8948×10^{1}	1.8993
a_{13}	1.5471×10^{-9}	1.9815×10^{1}	6.9523×10^{-3}
a_{03}	2.6725×10^{2}	3.9262×10^{2}	2.5262×10^{-2}
b_{03}	-3.1001×10^{1}	4.0654×10^{1}	3.8948×10^{-1}

Table B-36: The lower bounds, upper bounds, and μ–sensitivities after initial data points.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	6.1708×10^{-9}	1.0166×10^{-1}	2.7544×10^{-2}
a_{01}	7.8212×10^{-1}	1.1802	3.0132×10^{-2}
b_{01}	2.8966	4.9086	3.2816×10^{-1}
a_{12}	5.1640×10^{-9}	4.5244	9.5091×10^{-3}
a_{02}	2.9266×10^{1}	5.3809×10^{1}	9.0020×10^{-2}
b_{02}	-8.0258	1.6058×10^{1}	1.7048
a_{13}	2.2385×10^{-9}	1.9815×10^{1}	8.4559×10^{-3}
a_{03}	2.6725×10^{2}	3.9262×10^{2}	2.9910×10^{-2}
b_{03}	-3.0892×10^{1}	4.0654×10^{1}	4.2559×10^{-1}

Table B-37: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 5.39 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	6.9136×10^{-9}	1.0166×10^{-1}	1.5784×10^{-2}
a_{01}	7.8318×10^{-1}	1.1802	3.3215×10^{-2}
b_{01}	2.9055	4.9086	3.8742×10^{-1}
a_{12}	5.6791×10^{-9}	3.8323	8.5892×10^{-3}
a_{02}	2.9266×10^{1}	5.3809×10^{1}	5.6664×10^{-2}
b_{02}	-6.1032	1.4042×10^{1}	1.6023
a_{13}	3.3435×10^{-9}	1.9815×10^{1}	3.0732×10^{-3}
a_{03}	2.6725×10^{2}	3.9262×10^{2}	3.5972×10^{-2}
b_{03}	-2.9544×10^{1}	3.7529×10^{1}	4.7792×10^{-1}

Table B-38: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 20.20 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	7.4044×10^{-9}	1.0166×10^{-1}	1.6781×10^{-2}
a_{01}	7.8318×10^{-1}	1.1802	3.3992×10^{-2}
b_{01}	2.9055	4.9086	3.9566×10^{-1}
a_{12}	6.9503×10^{-9}	3.6342	9.8070×10^{-3}
a_{02}	2.9266×10^{1}	5.3809×10^{1}	4.5380×10^{-2}
b_{02}	-5.5028	1.3441×10^{1}	1.5038
a_{13}	3.9267×10^{-9}	1.9815×10^{1}	3.3474×10^{-3}
a_{03}	2.6725×10^{2}	3.9262×10^{2}	3.6036×10^{-2}
b_{03}	-2.8216×10^{1}	3.6204×10^{1}	4.7118×10^{-1}

Table B-39: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 20.06 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	7.6702×10^{-9}	1.0166×10^{-1}	1.7117×10^{-2}
a_{01}	7.8318×10^{-1}	1.1802	3.4017×10^{-2}
b_{01}	2.9055	4.9086	3.9572×10^{-1}
a_{12}	7.8217×10^{-9}	3.5860	1.0018×10^{-2}
a_{02}	2.9266×10^{1}	5.3809×10^{1}	4.3094×10^{-2}
b_{02}	-5.3566	1.3295×10^{1}	1.4773
a_{13}	5.0084×10^{-9}	1.9815×10^{1}	3.3618×10^{-3}
a_{03}	2.6725×10^{2}	3.9262×10^{2}	3.5870×10^{-2}
b_{03}	-2.7881×10^{1}	3.5872×10^{1}	4.6708×10^{-1}

Table B-40: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 20.26 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	8.5047×10^{-9}	1.0166×10^{-1}	1.7066×10^{-2}
a_{01}	7.8318×10^{-1}	1.1802	3.4183×10^{-2}
b_{01}	2.9055	4.9086	3.9762×10^{-1}
a_{12}	8.3153×10^{-9}	3.5746	1.0113×10^{-2}
a_{02}	2.9266×10^{1}	5.3809×10^{1}	4.2167×10^{-2}
b_{02}	-5.3220	1.3261×10^{1}	1.4684
a_{13}	6.2511×10^{-9}	1.9815×10^{1}	3.4197×10^{-3}
a_{03}	2.6725×10^{2}	3.9262×10^{2}	3.5965×10^{-2}
b_{03}	-2.7800×10^{1}	3.5792×10^{1}	4.6815×10^{-1}

Table B-41: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 20.13 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	8.9517×10^{-9}	1.0166×10^{-1}	3.8805×10^{-3}
a_{01}	7.8318×10^{-1}	1.1802	2.8042×10^{-3}
b_{01}	2.9055	4.9086	4.3852×10^{-2}
a_{12}	1.0256×10^{-8}	3.5683	1.0100×10^{-2}
a_{02}	2.9266×10^{1}	5.3809×10^{1}	2.1814×10^{-2}
b_{02}	-5.3032	1.3241×10^{1}	4.6106×10^{-1}
a_{13}	7.4262×10^{-9}	1.9815×10^{1}	1.4074×10^{-3}
a_{03}	2.6725×10^{2}	3.9262×10^{2}	5.5167×10^{-4}
b_{03}	-2.7746×10^{1}	3.5733×10^{1}	6.1149×10^{-2}

Table B-42: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 20.01 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.0212×10^{-8}	1.0166×10^{-1}	1.7105×10^{-2}
a_{01}	7.8318×10^{-1}	1.1802	3.4208×10^{-2}
b_{01}	2.9055	4.9086	3.9787×10^{-1}
a_{12}	1.1589×10^{-8}	3.5665	1.0149×10^{-2}
a_{02}	2.9266×10^{1}	5.3809×10^{1}	4.1762×10^{-2}
b_{02}	-5.2977	1.3236×10^{1}	1.4635
a_{13}	1.0332×10^{-8}	1.9815×10^{1}	3.4286×10^{-3}
a_{03}	2.6725×10^{2}	3.9262×10^{2}	3.5944×10^{-2}
b_{03}	-2.7733×10^{1}	3.5721×10^{1}	4.6753×10^{-1}

Table B-43: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 20.19 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.1165×10^{-8}	1.0166×10^{-1}	1.7108×10^{-2}
a_{01}	7.8318×10^{-1}	1.1802	3.4208×10^{-2}
b_{01}	2.9055	4.9086	3.9787×10^{-1}
a_{12}	1.3060×10^{-8}	3.5661	1.0150×10^{-2}
a_{02}	2.9266×10^{1}	5.3809×10^{1}	4.1741×10^{-2}
b_{02}	-5.2964	1.3235×10^{1}	1.4633
a_{13}	1.0821×10^{-8}	1.9815×10^{1}	3.4285×10^{-3}
a_{03}	2.6725×10^{2}	3.9262×10^{2}	3.5942×10^{-2}
b_{03}	-2.7730×10^{1}	3.5718×10^{1}	4.6749×10^{-1}

Table B-44: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 20.08 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.3766×10^{-8}	1.0166×10^{-1}	1.6298×10^{-2}
a_{01}	7.8318×10^{-1}	1.1802	3.4510×10^{-2}
b_{01}	2.9055	4.9086	4.0207×10^{-1}
a_{12}	1.4261×10^{-8}	3.5660	1.2284×10^{-2}
a_{02}	2.9266×10^{1}	5.3809×10^{1}	4.7944×10^{-2}
b_{02}	-5.2961	1.3234×10^{1}	1.4774
a_{13}	1.1804×10^{-8}	1.9815×10^{1}	3.6818×10^{-3}
a_{03}	2.6725×10^{2}	3.9262×10^{2}	3.5635×10^{-2}
b_{03}	-2.7729×10^{1}	3.5717×10^{1}	4.7107×10^{-1}

Table B-45: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 20.24 rad/sec.

B.3.2 Parameter Bounds Without Unmodelled Dynamics

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	3.1956×10^{-9}	1.1943×10^{-1}	3.1644×10^{-2}
a_{01}	7.8212×10^{-1}	1.1643	3.1166×10^{-2}
b_{01}	2.6588	4.8290	3.7177×10^{-1}
a_{12}	1.9103×10^{-9}	7.3355	1.5401×10^{-2}
a_{02}	2.9266×10^{1}	5.3809×10^{1}	7.6127×10^{-2}
b_{02}	-6.5011	1.4219×10^{1}	1.3977
a_{13}	2.1504×10^{-9}	1.9815×10^{1}	3.7307×10^{-3}
a_{03}	2.6725×10^{2}	3.9262×10^{2}	1.5129×10^{-2}
b_{03}	-1.5923×10^{1}	2.4232×10^{1}	2.3878×10^{-1}

Table B-46: The lower bounds, upper bounds, and μ–sensitivities after initial data points.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	3.8321×10^{-9}	9.4964×10^{-2}	3.4497×10^{-2}
a_{01}	7.9160×10^{-1}	1.1643	3.6277×10^{-2}
b_{01}	2.9445	4.8290	3.9563×10^{-1}
a_{12}	2.2790×10^{-9}	1.9675	1.2413×10^{-2}
a_{02}	2.9266×10^{1}	5.3809×10^{1}	4.8223×10^{-2}
b_{02}	-7.4727×10^{-1}	8.6883	5.2710×10^{-1}
a_{13}	2.8567×10^{-9}	1.9815×10^{1}	4.5500×10^{-3}
a_{03}	2.6725×10^{2}	3.9262×10^{2}	1.6697×10^{-2}
b_{03}	-1.0409×10^{1}	1.8337×10^{1}	2.3550×10^{-1}

Table B-47: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 5.39 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	4.8511×10^{-9}	9.4964×10^{-2}	3.5576×10^{-2}
a_{01}	7.9584×10^{-1}	1.1643	3.5858×10^{-2}
b_{01}	2.9680	4.8290	3.8898×10^{-1}
a_{12}	2.9977×10^{-9}	1.7313	1.3774×10^{-2}
a_{02}	2.9266×10^{1}	5.3345×10^{1}	5.7457×10^{-2}
b_{02}	-1.2706×10^{-1}	8.0656	4.4850×10^{-1}
a_{13}	3.1977×10^{-9}	1.8793×10^{1}	4.3339×10^{-3}
a_{03}	2.6725×10^{2}	3.9262×10^{2}	1.5594×10^{-2}
b_{03}	-9.5700	1.7485×10^{1}	2.2171×10^{-1}

Table B-48: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 20.20 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	5.2870×10^{-9}	9.4964×10^{-2}	3.6839×10^{-2}
a_{01}	7.9644×10^{-1}	1.1643	3.7678×10^{-2}
b_{01}	2.9712	4.8290	4.0794×10^{-1}
a_{12}	4.3215×10^{-9}	1.6543	1.4581×10^{-2}
a_{02}	2.9266×10^{1}	5.2758×10^{1}	6.2664×10^{-2}
b_{02}	3.9033×10^{-2}	7.9090	4.5652×10^{-1}
a_{13}	3.6083×10^{-9}	9.9455	6.5449×10^{-4}
a_{03}	2.6725×10^{2}	3.9262×10^{2}	3.3478×10^{-3}
b_{03}	-4.3364	1.2440×10^{1}	1.2977×10^{-1}

Table B-49: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 20.06 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	7.6697×10^{-3}	3.2217×10^{-2}	1.4624×10^{-2}
a_{01}	9.6456×10^{-1}	1.0350	5.2783×10^{-3}
b_{01}	3.8189	4.1791	7.0881×10^{-2}
a_{12}	2.9575×10^{-2}	2.2043×10^{-1}	1.5552×10^{-2}
a_{02}	3.8456×10^{1}	4.0094×10^{1}	1.7279×10^{-2}
b_{02}	3.7546	4.2459	3.8259×10^{-2}
a_{13}	1.4728×10^{-1}	5.5959×10^{-1}	3.0119×10^{-4}
a_{03}	3.0439×10^{2}	3.1150×10^{2}	6.6139×10^{-4}
b_{03}	3.8235	4.1762	2.5579×10^{-3}

Table B-50: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 20.59 rad/sec.

Parameter	Lower Bound	Upper Bound	μ–sensitivity
a_{11}	1.9637×10^{-2}	2.0362×10^{-2}	4.8277×10^{-4}
a_{01}	9.9943×10^{-1}	1.0006	1.1769×10^{-4}
b_{01}	3.9970	4.0030	2.3353×10^{-3}
a_{12}	1.2063×10^{-1}	1.3023×10^{-1}	2.6925×10^{-3}
a_{02}	3.9239×10^{1}	3.9306×10^{1}	3.2423×10^{-3}
b_{02}	3.9895	4.0106	5.7286×10^{-3}
a_{13}	3.2752×10^{-1}	3.7931×10^{-1}	9.2254×10^{-5}
a_{03}	3.0754×10^{2}	3.0840×10^{2}	2.4756×10^{-4}
b_{03}	3.9788	4.0225	9.1500×10^{-4}

Table B-51: The lower bounds, upper bounds, and μ–sensitivities after applying a sinusoid at 8.00 rad/sec.

References

[1] Akaike, H., "Stochastic Theory of Minimal Realization," *IEEE Trans. on Auto. Control*, Vol. 19, No. 6, December 1974, pp. 667–674.

[2] Akçay, H., Gu, G., and Khargonekar, P., "Identification in $\mathcal{H}_\infty$ with Nonuniformly Spaced Frequency Response Measurements," *Proc. 1992 American Control Conference, Chicago, IL*, June 1992, pp. 246–250.

[3] Balas, G. J., Doyle, J. C., Glover, K., Packard, A. K., and Smith, R. S., "The μ Analysis and Synthesis Toolbox," 1991. MathWorks and MUSYN.

[4] Bayard, D. S., Yam, Y., and Mettler, E., "A Criterion for Joint Optimization of Identification and Robust Control," *IEEE Trans. on Auto. Control*, Vol. 37, No. 7, July 1992, pp. 986–991.

[5] Belforte, G., Bona, B., and Cerone, V., "Parameter Estimation Algorithms for a Set-Membership Description of Uncertainty," *Automatica*, Vol. 26, No. 5, 1990, pp. 887–898.

[6] Blevins, R. D., *Formulas for Natural Frequency and Mode Shape*, Robert E. Krieger Publishing Co., Malabar, Florida, 1984.

[7] Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V., *Linear Matrix Inequalities in System and Control Theory*, Vol. 15 of *SIAM Studies in Applied Mathematics*, SIAM, 1994.

[8] Braatz, R. D. and Morari, M., "μ-Sensitivities as an Aid for Robust Identification," *Proc. 1991 American Control Conference, Boston, MA*, June 1991, pp. 231–236.

[9] Chen, J., Gu, G., and Nett, C. N., "Worst Case Identification of Continuous Time Systems via Interpolation," *Proc. 1993 American Control Conference, San Fransisco, CA*, June 1993, pp. 1544–1548.

[10] Chen, J., Nett, C. N., and Fan, M. K. H., "Worst-Case System Identification in $\mathcal{H}_\infty$: Validation of Apriori Information, Essentially Optimal Algorithms, and Error Bounds," *Proc. 1992 American Control Conference, Chicago, IL*, June 1992, pp. 251–257.

[11] Cheung, M., Yurkovich, S., and Passino, K. M., "An Optimal Volume Ellipsoid Algorithm for Parameter Set Estimation," *IEEE Trans. on Auto. Control*, Vol. 38, No. 8, August 1993, pp. 1292–1296.

[12] Deller, J. R., "Set Membership Identification in Digital Signal Processing," *IEEE ASSP Magazine*, Vol. 6, No. 4, October 1989, pp. 4–20.

[13] Douglas, J., "*Linear Quadratic Control for Systems with Structured Uncertainty.*" SM thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May 1991. SERC report #12-91.

[14] Douglas, J. and Athans, M., "Application of Set-Membership Identification to Lightly Damped Systems," *Proc. 1995 American Control Conference, Seattle, WA*, 1995.

[15] Douglas, J. and Athans, M., "The Calculation of μ–Sensitivities," *Proc. 1995 American Control Conference, Seattle, WA*, 1995.

[16] Doyle, J. C., "Structured Uncertainty in Control System Design,," *Proceedings of the 24^{th} Conference on Decision and Control*, 1985, pp. 260–265.

[17] Doyle, J. C., Glover, K., Khargonekar, P., and Francis, B., "State-Space Solutions to Standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Control Problems," *IEEE Trans. on Auto. Control*, Vol. 34, No. 8, August 1989, pp. 831–847.

[18] Fogel, E. and Huang, Y. F., "On the Value of Information in System Identification - Bounded Noise Case," *Automatica*, Vol. 18, No. 2, 1982, pp. 229–238.

[19] Freudenberg, J. S., Looze, D. P., and Cruz, J. B., "Robustness Analysis Using Singular Value Sensitivities," *Int. Journal of Control*, Vol. 35, No. 1, 1982, pp. 95 – 116.

[20] Gahinet, P. and Nemirovskii, A., "General-Purpose LMI Solvers with Benchmarks," *Proc. 1993 Conference on Decision and Control*, December 1993, pp. 3162–3165.

[21] Gilpin, K., "*Identification of a Lightly Damped Structure for Control/Structure Interaction.*" SM thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, August 1991. SERC report #11-91.

[22] Grocott, S. C. O., How, J. P., and Miller, D. W., "Comparison of Control Techniques for Robust Performance on Uncertain Structural Systems," January 1994. MIT SERC report #2-94.

[23] Grocott, S. C. O., How, J. P., and Miller, D. W., "A Comparison of Robust Control Techniques for Uncertain Structural Systems," *AIAA Guidance, Navigation, and Control Conference, Scottsdale, AZ*, August 1994, pp. 261–271.

[24] Gu, G., "Suboptimal Algorithms for Worst Case Identification and Model Validation," *Proc. 1993 Conference on Decision and Control, San Antonio, TX*, December 1993, pp. 539–544.

[25] Helmicki, A. J., Jacobson, C. A., and Nett, C. N., "Control Oriented System Identification: a Worst-Case/Deterministic Approach in $\mathcal{H}_\infty$," *IEEE Trans. on Auto. Control*, Vol. 36, October 1991, pp. 1163–1176.

[26] Helmicki, A. J., Jacobson, C. A., and Nett, C. N., "Least Squares Methods for $\mathcal{H}_\infty$ Control-Oriented System Identification," *Proc. 1992 American Control Conference, Chicago, IL*, June 1992, pp. 258–264.

[27] Ho, B. L. and Kalman, R. E., "Effective Construction of Linear State-Variable Models from Input/Output Function," *Regelungstechnik*, 1966, pp. 545–548.

[28] Jacques, R. N. and Miller, D. W., "Multivariable Model Identification from Frequency Response Data," *Proc. 1993 Conference on Decision and Control, San Antonio, TX*, December 1993, pp. 3046–3051.

[29] Juang, J. N. and Pappa, R. S., "An Eigensystem Realization Algorithm for Modal Parameter Identification and Model Reduction," *Journal of Guidance, Control and Dynamics*, Vol. 8, No. 5, Sept./Oct. 1985, pp. 620–627.

[30] Juang, J. N. and Pappa, R. S., "Effects of Noise on Modal Parameters Identified by the Eigensystem Realization Algorithm," *Journal of Guidance, Control and Dynamics*, Vol. 9, No. 3, May/June 1986, pp. 294–303.

[31] Juang, J. N., Phan, M., Horta, L. G., and Longman, R. W., "Identification of Observer/Kalman Filter Markov Parameters: Theory and Experiments," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 2, March/April 1993, pp. 320–329.

[32] Kato, T., *Perturbation Theory for Linear Operators*, Springer-Verlag, 1976.

[33] Lamaire, R. O., Valavani, L., Athans, M., and Stein, G., "A Frequency-Domain Estimator for Use in Adaptive Control Systems," *Automatica*, Vol. 27, No. 1, 1991, pp. 23–38.

[34] Liu, K. and Miller, D. W., "Time Domain State Space Identification of Structural Systems," *ASME Journal of Dynamic Systems, Meaurement and Control*, to appear.

[35] Liu, K. and Skelton, R. E., "Q-Markov Covariance Equivalent Realization and its Application to Flexible Structure Identification," *Journal of Guidance, Control and Dynamics*, Vol. 16, No. 2, March/April 1993, pp. 308–319.

[36] Ljung, L., *System Identification: Theory for the User*, Prentice Hall, 1987.

[37] Lopez-Toledo, A. A., "*Optimal Inputs for Identification of Stochastic Systems.*" PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, September 1974. Report ESL-R-566.

[38] Lublin, L. and Athans, M., "An Experimental Comparison of $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Designs for an Interferometer Testbed," in *Lecture Notes in Control and Information Sciences: Feedback Control, Nonlinear Systems, and Complexity* (Francis, B. A. and Tannenbaum, A. R., eds.), pp. 150–172, Springer, 1995.

[39] Mehra, R. K., "Optimal Input Signals for Parameter Estimation in Dynamic Systems - Survey and New Results," *IEEE Trans. on Auto. Control*, Vol. 19, No. 6, December 1974, pp. 753–768.

[40] Milanese, M. and Belforte, G., "Estimation Theory and Uncertainty Intervals Evaluation in Presence of Unknown But Bounded Errors: Linear Families of Models and Estimators," *IEEE Trans. on Auto. Control*, Vol. 27, No. 2, April 1982, pp. 408–414.

[41] Milanese, M. and Vicino, A., "Estimation Thoery for Nonlinear Models and Set Membership Uncertainty," *Automatica*, Vol. 27, No. 2, 1991, pp. 403–408.

[42] Milanese, M. and Vicino, A., "Optimal Estimation Theory for Dynamic Systems with Set Membership Uncertainty: An Overview," *Automatica*, Vol. 27, No. 6, 1991, pp. 997–1009.

[43] Nemirovskii, A. and Gahinet, P., "The Projective Method for Solving Linear Matrix Inequalities," *Proc. 1994 American Control Conference*, June 1994, pp. 840–844.

[44] Norton, J. P., "Identification of Parameter Bounds for ARMAX Models from Records with Bounded Noise," *Int. J. Control*, Vol. 45, No. 2, 1987, pp. 375–390.

[45] Piet-Lahanier, H. and Walter, E., "Characterization of Non-Connected Parameter Uncertainty Regions," *Mathematics and Computers in Simulation*, Vol. 32, 1990, pp. 553–560.

[46] Pronzato, L. and Walter, E., "Minimal Volume Ellipsoids," *Int. J. of Adaptive Control and Signal Processing*, Vol. 8, 1994, pp. 15–30.

[47] Rudin, W., *Principles of Mathematical Analysis*, McGraw–Hill, third ed., 1976.

[48] Schrama, R. J., "Accurate Identification for Control: The Necessity of an Iterative Scheme," *IEEE Trans. on Auto. Control*, Vol. 37, No. 7, July 1992, pp. 991–994.

[49] Schweppe, F. C., *Uncertain Dynamic Systems*, Prentice-Hall, 1973.

[50] Sidman, M. D., DeAngelis, F. E., and C.Verghese, G., "Parametric System Identification on Logarithmic Frequency Response Data," *IEEE Trans. on Auto. Control*, Vol. 36, No. 9, September 1991, pp. 1065–1070.

[51] Silverman, L. M., "Realization of Linear Dynamical Systems," *IEEE Trans. on Auto. Control*, Vol. 16, December 1971, pp. 554–567.

[52] Stein, G. and Doyle, J. C., "Beyond Singular Values and Loop Shapes," *Journal of Guidance, Control and Dynamics*, Vol. 14, No. 1, Jan./Feb. 1991, pp. 5–16.

[53] Sun, J., "Multiple Eigenvalue Sensitivity Analysis," *Linear Algebra and Its Applications*, Vol. 137/138, 1990, pp. 183–211.

[54] Walter, E. and Piet-Lahanier, H., "Estimation of Parameter Bounds from Bounded-Error Data: A Survey," *Mathematics and Computers in Simulation*, Vol. 32, 1990, pp. 449–468.

[55] Williamson, D., *Digital Control and Implementation: Finite Wordlength Considerations*, Prentice Hall, 1991.

[56] Young, P. M. and Doyle, J. C., "Properties of the Mixed μ Problem and its Bounds," *IEEE Trans. on Auto. Control*, to appear.

[57] Young, P. M., Newlin, M. P., and Doyle, J. C., "Let's Get Real," in *Robust Control Theory* (Francis, B. and Khargonekar, P., eds.), pp. 143–173, Springer-Verlag, 1995.

[58] Young, P. M., Newlin, M. P., and Doyle, J. C., "Computing Bound for the Mixed μ Problem," *Int. Journal Robust and Nonlinear Control*, to appear.

[59] Zafiriou, E. and Morari, M., "Design of the IMC Filter By Using the Structured Singular Value Approach," *Proc. 1986 American Control Conference, Seattle, WA*, June 1986, pp. 1–6.