Gain Scheduling for Geometrically Nonlinear Flexible Space Structures

by

Jeremy Hoyt Yung

S.B., Aeronautics and Astronautics
S.B., Electrical Engineering
Massachusetts Institute of Technology, 1993

S.M., Aeronautics and Astronautics
Massachusetts Institute of Technology, 1996

SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
IN PARTIAL FULFILLMENT OF THE DEGREE OF

DOCTORATE OF PHILOSOPHY
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2001

Signature of Author . . . . . .
Department of Aeronautics and Astronautics
September 28, 2001

Certified by . . . . . .
David W. Miller, Associate Professor
Thesis Committee Chairman

Certified by . . . . . .
Carl A. Blaurock, Ph. D.
Thesis Committee Member

Certified by . . . . . .
Edward Crawley, Professor and Department Head
Thesis Committee Member

Certified by . . . . . .
Marthinus C. van Schoor, Ph. D.
Thesis Committee Member

Certified by . . . . . .
Eric Feron, Associate Professor
Minor Advisor

Accepted by . . . . . .
Professor Wallace Vander Velde
Chairman, Department Graduate Committee

Gain Scheduling for Geometrically Nonlinear Flexible Space Structures

by

JEREMY HOYT YUNG

Submitted to the Department of Aeronautics and Astronautics
on September 28, 2001 in partial fulfillment of the
requirements for the degree of Doctorate of Philosophy
at the Massachusetts Institute of Technology

ABSTRACT

A gain-scheduling approach for the control of geometrically nonlinear structures is developed. The objective is to improve performance over current linear design techniques that are applied to the same control problem. The approach is applicable to a variety of structures that have complex dynamics with slow variations such as flexible robotic arms and space structures with gimballing solar arrays.

The modeling approach is motivated by the lack of *in situ* test data available for design of 0-g controllers. A Linear Fractional form allows the nonlinear and uncertain aspects of the structure to be modeled independently. The geometric nonlinearity is modeled using a feedback description of structural coupling. The uncertainty model is based on a physical parameter description, so that an experimentally identified 1-g parametric uncertainty model can be extrapolated to 0-g.

The control approach is motivated by the success of linear control design synthesis and analysis techniques for space structures. Graphical heuristics for linear control design using Linear Quadratic Gaussian (LQG) and Sensitivity Weighted LQG techniques are introduced. A procedure to realize reduced-order gain-scheduled controllers from a family of linear state-space controllers is developed. A nonlinear analysis framework suitable for the slow variations of geometrically nonlinear structures is also presented. The realization procedure and nonlinear analysis is combined with the graphical linear design heuristics to form an iterative gain scheduled design process.

The complete gain scheduling approach is applied to the MIT/MACE-II experiment flown on the International Space Station. Gain scheduled controller designs are shown to provide improved performance and robustness over a Multiple Model linear controller design.

Thesis Supervisor:
Professor David W. Miller
Department of Aeronautics and Astronautics

ACKNOWLEDGEMENTS

I respectfully and gratefully acknowledge just some of those who made this work possible:

Professor David Miller, for giving me the opportunity of a lifetime in working on MACE; Dr. Carl Blaurock, who dragged me towards the end of the tunnel when there was no light; Professor Edward Crawley, my first mentor at MIT; Dr. Marthinus van Schoor and Professor Eric Feron, who graciously afforded me their time and wisdom; and Professor Jonathan How and my good friend Dr. Mauro Atalla for having the patience to slog through my thesis.

My friends over the many years in the Lab without whom life would have been unbearable, especially Mark and Amy Andersson, Eric Prechtl, Yool Kim, Alex Makarenko, Timothy Glenn, Patrick Trapa, Alice Liu and Laila Elias. And SharonLeah Brown, who has fiercely protected her brood from those who might otherwise take advantage of them.

My friends outside the Lab who have shown me a life away from work these last few years, Jeannie Chin, Anne Hui, and Shane Shih. Jane Ngo, for listening and sharing. Nhai Cao, and Bernard and Agnes Chen for taking me in when I had no place to go.

My mother and father, and my sister, who have always been there for me and patiently waited all these years for me to finish at MIT.

And most of all, my wife Susan. Who was there to suffer my deepest and darkest moments. It's over. I'm done.

Chapter 1

INTRODUCTION

This thesis will develop and demonstrate Controls/Structures Interaction (CSI) control approaches for Time Varying (TV) systems. CSI problems arise when control detrimentally interacts with flexibility in a system. Such interaction is caused by mismodeling or lack of consideration of flexibility. The standard approach to CSI is to limit the control bandwidth to below the frequencies of structural vibration. However, as required precision increases, and structural frequencies decrease, this approach will unduly and perhaps unacceptably limit performance. Overcoming this problem requires a multidisciplinary approach combining modeling and model updating, system identification, uncertainty model development, linear systems analysis, and robust control synthesis.

Time variations in the controlled system can arise from configuration changes. Examples are lightweight robotics which undergo large changes in geometry and flexible spacecraft solar arrays which rotate with respect to the spacecraft to track the sun. Such motions are typically slow, and the velocity dependent terms in the equations of motion can be ignored. These types of systems will be referred to as *geometrically nonlinear*, indicating that the nonlinearity is primarily a function of the geometry or *setpoint*.

CSI control approaches require a detailed model of nominal dynamic behavior, as well as an uncertainty description. Standard approaches dictate that *in situ* test data be used for the development of such control design models. However, *in situ* (0-g) test data is not

generally available for space-based systems prior to flight. A modeling approach that can incorporate 1-g test data is thus required to develop 0-g models for CSI control.

The thesis outlines an approach for the modeling and control of geometrically nonlinear structures. A modeling framework that captures both the variations of the geometrically nonlinear system and uncertainties is developed to produce a control design model. Proven linear synthesis and analysis tools are incorporated into a integrated design process for gain scheduled controller design. The design process is simplified by the introduction of graphical heuristics for the design of linear controllers.

The introduction is divided into a summary of the research objectives; a literature review of relevant modeling and control research; an introduction to MIT/MACE-II, the experimental focus of the thesis; a motivation and overview of the approach used in the thesis; and finally a thesis outline.

1.1 Research Objectives

The overall objective of the thesis is to develop and demonstrate a framework for configuration dependent control of a geometrically nonlinear system. There are three specific objectives that must be satisfied in order to satisfy the overall objective:

- Develop a framework for modeling a geometrically nonlinear system. This model includes an error model for the design of a robust controller, and must be done without *in situ* test data.
- Develop a framework for designing robust gain scheduled controllers for a geometrically nonlinear system. These controllers are designed to achieve better performance than linear time invariant (LTI) controllers for the same geometrically nonlinear system.
- Implement the modeling and control framework on an experimental testbed. This testbed is the Middeck Active Control Experiment Reflight (MIT/MACE-II), which was flown on STS-106 for deployment on the International Space Station in September 2000.

1.2 Literature Review

1.2.1 Modeling of Geometrically Nonlinear Systems

Much of the previous relevant work concerning the modeling of geometrically nonlinear structures falls under the broad category of modeling for flexible multi-body/multi-link dynamics.

Lagrangian methods [Meirovitch, 1986] employ a system energy function that is written in terms of a set of assumed degrees of freedom (DOF). The Lagrangian equation can then be used to determine the equations of motion. [Kelkar and Alberts, 1991] develop closed form symbolic equations for a planar model of a flexible two-link manipulator. Similar derivations are performed by [D'Eleuterio and Sincarsin 1991] for general variable-geometry truss structures, and [Mordfin and Sivakumar, 2000] for planar flexible multi-link systems arranged in chain topologies. Lagrangian methods have the advantage of accuracy; [Mordfin and Sivakumar, 2000] use such methods as truth models to validate the choice of shapefunctions for assumed modes. Since the energy function is generally straightforward to formulate, the method is extremely flexible and can describe complex systems. However, the accuracy of the approach is dependent on the set of assumed DOF that are chosen. A reduced set of DOF will not be able to capture the system energy correctly, while a large set of DOF may result in equations of motion that are extremely complex. For realistic systems, such as three-dimensional manipulators, it is often the case that a set of assumed DOF that is large enough to capture the system energy results in a set of equations of motion that are prohibitively expensively to simulate.

An alternative approach to analytical modeling of flexible multi-body dynamics uses *Kane's Equations*, which is a method to solve for inertial forces from expressions for the inertial velocities [Oakley and Cannon Jr., 1990]. The highly formalized structure of the solution process makes it suitable for modeling complex multi-link structures [Mueller-Karger and Townsend, 1997]. For realistic geometrically nonlinear systems such as three-dimensional manipulators, however, both Kane's Equations and Lagrangian methods tend

to produce equations of motion that are too complex for useful simulation, and intractable for the purposes of robust control.

Simplified modeling packages, such as Dymola, [Elmqvist, 1994], and DYNAST, [Mann, 2001] make use of simple blocks such as rigid bodies, elastic elements and nonlinear components that are combined to simulate physical systems. The block interconnection nature of these packages facilitates rapid prototyping. The resulting models are used for simulation purposes, but the high-level modeling process can obscure the underlying physical structure, which in turn makes model parameter tuning difficult.

There is a large body of knowledge in *Finite Element Modeling* techniques [Bathe, 1982], which are a direct extension of Lagrangian energy methods. The system energy function is spatially discretized using a simplified set of assumed modes. Finite element methods generally require a high level of discretization in order to capture the strain distribution correctly; consequently the models are of unnecessarily large order and must be truncated for control design. Component modeling techniques have also been developed to apply finite element techniques to linear components which are then combined for a multi-body analysis [Geradin et al., 1989].

1.2.2 Model Updating and Uncertainty Modeling

[Mottershead and Friswell, 1993] present a detailed survey of model updating for finite element based linear models using experimental data. The substantial body of literature on this topic can be divided according to the type of experimental data that is available for updating. Many techniques utilize detailed eigenvector data, which entails an elaborate data collection process that is impractical for space-based systems. Techniques that update the modal based on eigenvalue information still require partial eigenvector information in order to correlate experimental and analytical mode shapes. [Visser and Imregun, 1991] explore techniques of model updating that incorporate data from experimental frequency responses.

Uncertainty modeling is closely related to model updating. [Crawley et al., 1995, and Ingham and Crawley, 2001] measure modal parameter variations due to laboratory suspension effects, disassembly/re-assembly, 0-g/1-g effects and other phenomena related to deployable space structures. [Campbell, 1996] uses the Discrete Extended Kalman Filter to estimate modal statistics (mean and average) from multiple experimental data runs. These are used both to update a nominal model and to develop an uncertainty model.

1.2.3 Nonlinear Control for Geometrically Nonlinear Systems

Nonlinear control has been well-studied in the literature and remains an active field of research. [Blaurock, 1997] performs a survey of time-varying control techniques. Several general categories emerge: *ad hoc* TV implementations of fixed gain controllers; adaptive control [Yurkovich et al., 1989]; feedback inversion; variable structure/sliding mode control [Slotine and Li, 1991]; and linear robust control for TV parametrically uncertain systems [Zhou et al., 1996]. In the same work, [Blaurock, 1997] performs a detailed comparison of control techniques practically applicable to the control of geometrically nonlinear structures (specifically a model of the Space Station Remote Manipulator System). These techniques included the aforementioned linear robust control and adaptive control, and also an approximation to Feedback Linearization [Khorrami et al., 1994], and a gain scheduled controller based on the output interpolation of multiple linear controllers. The results of the study conclude that gain scheduling can indeed be an effective control form for the control of geometrically nonlinear structures. The remainder of this review is devoted to gain scheduling.

A theme common to the various methods of gain scheduling in the literature is the division of the nonlinear control problem into linear sub-problems. This divide and conquer approach allows the application of well established linear design methods to the nonlinear control problem. Linear design methods are particularly attractive for two reasons; first, that there is an extensive body of research, and secondly, that linear design methods are well accepted in industry for the purposes of certification. This section presents a review

of the current gain scheduling techniques, in the context of selecting a technique for the control of geometrically nonlinear structures.

Stability proofs for slowly varying systems occur in the literature as early as the 1960's [Desoer, 1969, Hoppensteadt, 1966], and is extended in the 1990's [Khalil and Kokotovic, 1991]. These show that the stability of a family of linearized systems corresponding to a nonlinear plant implies stability of the nonlinear plant for slow variations within the neighborhood of the linearizations. Furthermore, for slow variations, the nonlinear system inherits the robustness properties of the linearized systems.

Classical gain scheduling generally involves the creation of a parameterized plant, and developing linearizations at multiple equilibrium points; this follows directly from the Linear Parameter Varying (LPV) form,

$$\begin{bmatrix} \dot{x} \\ y \end{bmatrix} = \begin{bmatrix} A(\Theta) & B(\Theta) \\ C(\Theta) & D(\Theta) \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}, \tag{1.1}$$

A corresponding linear controller is designed for each equilibrium point, and this family of linear controllers is then combined to form a gain scheduled nonlinear controller. Conventional approaches linearly interpolate the controller parameters; either the state-space structure [Hyde and Glover, 1993], or the gain, poles and zeros of the controller, [Nichols et al., 1993].

[Leith and Leithead, 1999b] propose velocity-based linearizations that extend the operating envelope of the nonlinear controller beyond the vicinity of the equilibrium points. Neural network based models of the nonlinear system have been used to design blended local controllers [Tanaka, Ikeda and Wang, 1998]. These approaches are all still related to classical gain scheduling in that a family of linear controllers is designed for multiple linearizations of a nonlinear plant.

There are several gain scheduling techniques that take advantage of the LPV form. These techniques typically have in common the direct synthesis of a nonlinear controller, as

opposed to the previous techniques which assemble the nonlinear controller from a family of linear controllers. In the late 1980s the analysis of gain scheduling for LPV systems was developed [Shamma and Athans, 1990], and there are currently several Lyapunov function approaches for the design of gain scheduled controllers. Placing a parameter dependence on the Lyapunov function can reduce the conservativeness of the controller; however this dramatically increases the complexity of the solution process. Basis function and piecewise linear approaches have been developed to make the solution process tractable [Wu et al., 1995 and Lim and How, 1997].

In certain cases LPV systems can be formulated as parameter dependent Linear Fractional Transformation (LFT) form. The LFT form restricts the parameter variance of a nonlinear system to a parameter feedback matrix. Systems that have parameter variance expressed as polynomials or rational functions can be transformed into an LFT description [Belcastro, 1998]. This LFT form can be used to design gain scheduled controllers using a small gain approach [Apkarian and Gahinet, 1995]. This small gain approach has been applied to an LFT description of a structural system with geometric nonlinearity [Blaurock, 1997].

1.3 Approach

The overall objective of this thesis requires the design of a configuration dependent controller for geometrically nonlinear systems. As presented in the literature review, there are several modeling and control techniques that may be applicable to this overall objective. However, the specific objectives of this thesis present a challenging set of requirements:

1. The modeling process must be able to accurately represent geometric nonlinearity as well as the modally dense lightly damped dynamics of a structural system.
2. The control design must be able to take advantage of the additional information that is provided by the knowledge of the geometric nonlinearity.
3. The modeling process must incorporate a model updating procedure, as well as provide an uncertainty description.
4. The updates to the model and the development of the uncertainty model must be performed without 0-g (*in situ*) test data.

5. The control synthesis and analysis process must be able to incorporate this uncertainty description so that a reasonable prediction of closed-loop behavior can be made.
6. The control design must be able to provide performance for nonanalogous or noncollocated control topologies, or where there is flexibility between any combination of control actuator, control sensor, disturbance source and performance metric.

No modeling and control framework that meets this set of requirements was revealed during the literature search. 1) and 3) severely restrict the use of many analytical modeling techniques; 2) precludes the use of linear robust control schemes that treat nonlinearity as a parametric uncertainty; 4) places many model updating procedures out of reach; 5) cannot be met by many nonlinear control schemes because of the exhaustive simulation required; and 6) creates phase losses that preclude many forms of nonlinear analysis.

The approach taken in this thesis to satisfy all of these requirements is summarized in Figure 1.1. The overall objective is to derive a gain scheduled controller, given an initial geometrically nonlinear system and pre-flight testing. This is the path (A-D) in Figure 1.1. The solution to the objective involves three major steps. The first step (A–B) consists of the modeling procedure, and extends the work of [Blaurock, 1997], where geo-

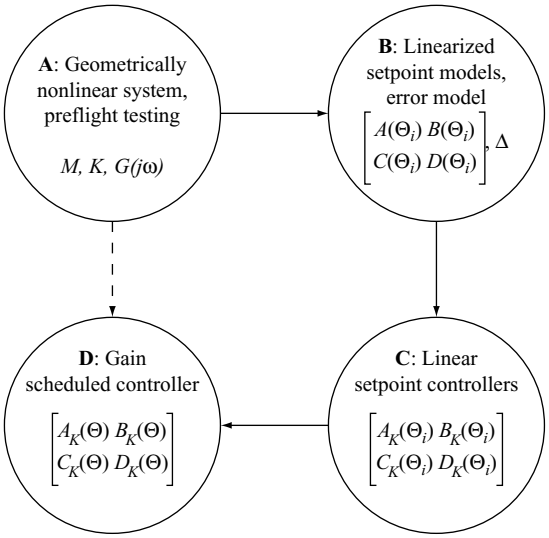

Figure 1.1 Thesis Approach.

metric nonlinearity is implemented as structural feedback gains between linear components. The dynamics of the uncoupled linear components are described with assumed modes. As part of the contribution of this thesis, the modeling framework is extended in [Blaurock et al., 1999] to incorporate linear Finite Element (FE) analysis results to facilitate the analysis of complex structures.

Robust control also requires a description of the associated uncertainties, or differences between the nominal model and the true plant. MACE-I demonstrated the effectiveness of a parametric error model, which describes the model error in terms of allowable variations in model parameters (such as stiffness and mass). In this work, a parametric error model that is valid for geometrically nonlinear flexible systems is developed.

The control synthesis methodology presented in this thesis is based on the design of linear controllers at different setpoints to provide reference points for a gain scheduled controller. The second step (B–C) of the thesis approach thus focuses on designing linear setpoint controllers for linearized setpoint models of the nonlinear system. The MACE-I design tools for the synthesis and analysis of high performance robust linear controllers are extended in this thesis to meet the requirements of the gain-scheduling design process. In particular, the individual linear control designs should be related in some sense. This thesis will develop the concept of *smoothness* in the scheduling variables, and apply it to the linear control design process.

H_2 control synthesis techniques such as LQG and SWLQG typically require several iterations of control design and analysis to achieve a desired level of performance and robustness. This is further complicated by the fact that there are numerous weighting matrices that must be selected in the LQG and SWLQG solution processes. This thesis also develops graphical heuristics to provide guidelines for the selection of weighting matrices for LQG and SWLQG control synthesis.

The final step (C–D) of the thesis approach is the realization of a gain-scheduled controller from a family of linear setpoint controllers. Several state-space transformation, reduc-

tion and interpolation procedures are developed to perform this realization. An analysis procedure suitable for a geometrically nonlinear closed loop system is also developed. These procedures are combined with the linear design tools to form an iterative design process for gain-scheduled controllers.

1.4 Introduction to MIT/MACE-II

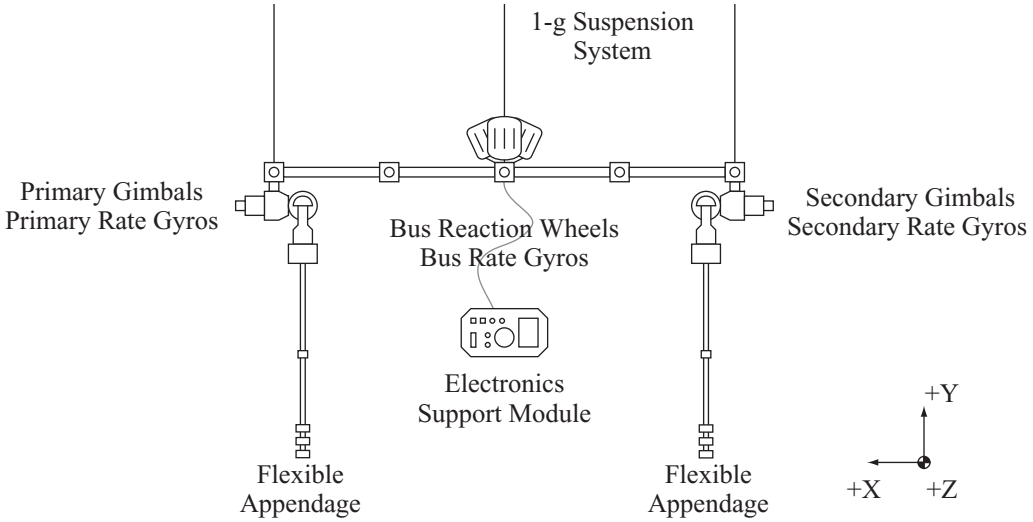

Figure 1.2 The MIT/MACE-II Experiment.

The experimental focus of this thesis is the MIT/MACE-II experiment. A functional diagram is shown in Figure 1.2. The primary structure is the Multi-Body Platform (MBP), a one-meter truss constructed of Lexan tubes with aluminum nodes. A three axis reaction wheel package is located at the center of the MBP. A three-axis rate gyro package is located at the reaction wheels to provide coarse attitude control capability. At each end of the MBP, a two-axis gimbal is mounted. These gimbals drive payload cans containing two-axis rate gyro packages. The geometric nonlinearity in the MBP is amplified using flexible appendages that are mounted to the outboard side of the payload cans. The electronics support module provides all of the real-time computing power, sensor conditioning and actuator amplification necessary to perform active control. A pneumatic suspension system with active stiffness and mass cancellation is used for 1-g testing; a photograph of this entire configuration is shown in Figure 1.3.

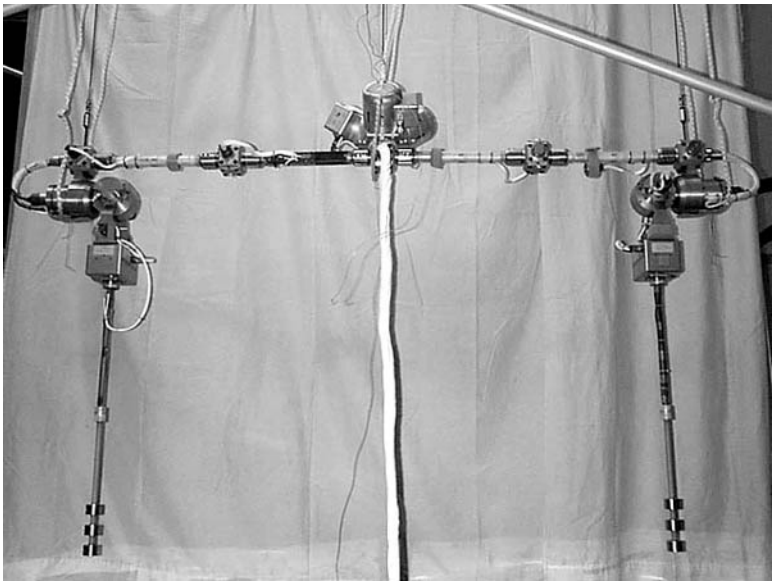

Figure 1.3 Photograph of the Multi-Body Platform with flexible appendages.

1.5 Thesis Outline

Figure 1.4 serves as a roadmap to the thesis. The modeling approach for geometrically nonlinear structures is developed in Chapter 2. This approach combines Finite Element modeling techniques, geometric nonlinearity and a physical parameter uncertainty model to create a control design model that is suitable for robust control techniques.

In Chapter 3, design tools for the synthesis and analysis of linear controllers are reviewed. These tools are developed for the purpose of developing setpoint controllers for the linearized setpoint models developed in Chapter 2. Graphical heuristics that assist in the selection of control design parameters for Linear Quadratic Gaussian (LQG) and Sensitivity Weighted LQG are introduced.

These linear control design techniques are absorbed into a gain scheduling design framework in Chapter 4. An approach is developed for interpolating between a discrete family of linear state-space setpoint controllers; this is incorporated into an algorithm for the design of high-performance robust gain scheduled controllers for geometrically nonlinear systems.

Chapter 5 describes the application of the modeling and control approach to the MIT/ MACE-II flight experiment. Results show that the gain scheduled control methodology presented in this thesis can produce controllers that can achieve increased performance with improved robustness when compared to standard linear control design techniques.

Finally conclusions are presented in Chapter 6 including contributions and recommendations for further work.

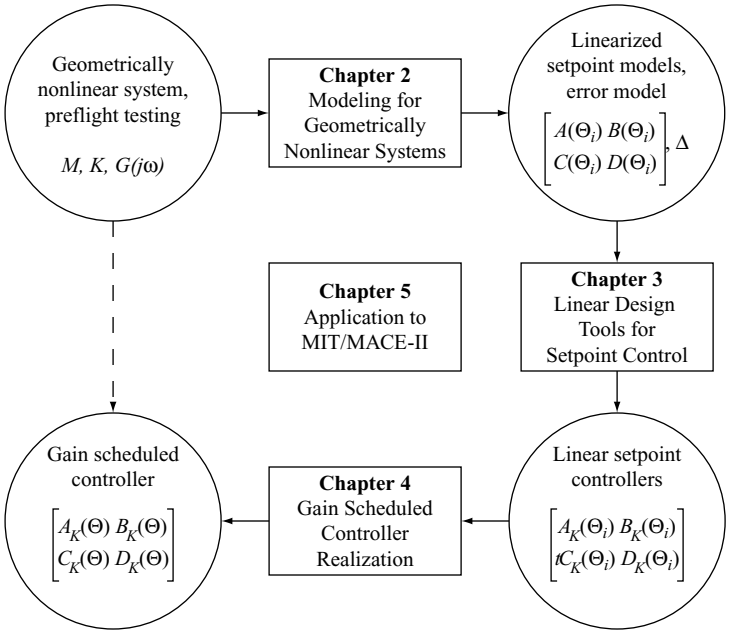

Figure 1.4 Thesis roadmap.

Chapter 2

MODELING OF GEOMETRICALLY NONLINEAR SYSTEMS

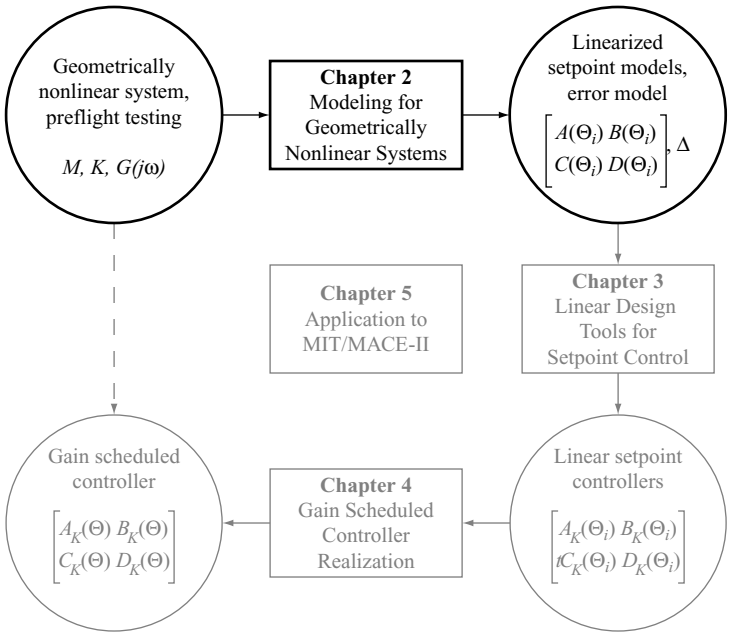

Figure 2.1 Modeling in gain scheduling framework. The inputs to the modeling step are physical measurements of the system, represented by mass and stiffness matrices M and K, and experimental data, $G(j\omega)$. The modeling step produces linearized models $[A(\Theta_i), B(\Theta_i), C(\Theta_i), D(\Theta_i)]$ as well as an error model, represented by Δ.

The overall objective of this thesis is to design robust controllers for geometrically nonlinear flexible structures. The first step towards meeting this objective is to develop a control design model (Figure 2.1). This model must meet several requirements in order to be suitable for the linear and gain-scheduled control design techniques developed in this thesis:

- The model must be capable of representing the large order dynamics that are necessary for the control of flexible structures.
- The model must account for geometric nonlinearity in the system. *Geometrically nonlinear* systems are defined as systems where velocity dependent terms in the equations of motion can be ignored.
- Discrepancies between the model and the actual system response must be accounted for as model uncertainties.
- The resulting nonlinear model must be linearized at multiple operating points, or *setpoints*, for control synthesis and analysis.

An additional constraint on the modeling process for space-based structures is the lack of *in situ* (0-g) test data. Prior to flight, only data from 1-g testing is available. Even though suspension systems can be used to simulate the *in situ* environment, gravitational effects such as sag and pre-stress must be accounted for when validating 1-g models with test data. Furthermore, uncertainty models developed in 1-g at a single linearized operating point must be extended to the full 0-g nonlinear operational range.

The solution to these requirements uses several proven modeling tools:

- Finite Element (FE) techniques are used to create the nominal model; this is a proven method for obtaining high fidelity structural models for control. The nominal model developed in Chapter 5 is based on the linear FE model developed for the MACE-I mission [Glaese, 1994, Miller et al., 1996].
- Geometric nonlinearity is incorporated using a feedback description of structural coupling through large angle joint rotations, as documented in [Blaurock, 1997] using assumed modes, and with Finite Element components [Blaurock et al., 1999].
- The development of the uncertainty models follows the MACE-I approach [Miller et al., 1996], where nominal modal parameters were updated with experimental data. The update process is then repeated across several datasets to obtain a modal parameter statistics database. This approach is extended to the geometrically nonlinear problem in this thesis by developed a physically based parameter description [Blaurock et al., 2001].

The modeling effort produces a control design model in Linear Fractional (LF) form. This is shown in Figure 2.2. Large order system dynamics are incorporated into the linear sys-

tem *P*, and nonlinear terms Θ and uncertain terms Δ included via a feedback matrix. The linear system is represented in state-space form,

$$P:\begin{bmatrix}\dot{x}\\ q\\ y\end{bmatrix} = \begin{bmatrix}A & B_p & B_u\\ C_q & D_{qp} & D_{qu}\\ C_y & D_{yp} & D_{yu}\end{bmatrix}\begin{bmatrix}x\\ p\\ u\end{bmatrix}, \tag{2.1}$$

and the nonlinear and uncertain terms are incorporated via the block diagonal feedback matrix,

$$p = \begin{bmatrix}\Theta & \\ & \Delta\end{bmatrix} q\,. \tag{2.2}$$

The system dynamics, *P*, are strictly linear and "certain", so that it can be manipulated and analyzed with linear algebra tools.

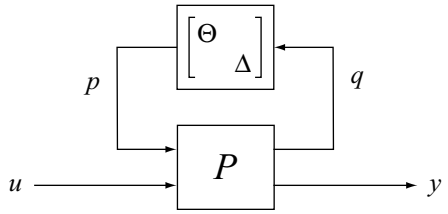

Figure 2.2 Linear Fractional representation of geometrically nonlinear uncertain system. The plant *P* is linear, and nonlinearity and uncertainty are incorporated as feedback matrices Θ and Δ respectively.

The modeling framework is presented as a flowchart in Figure 2.3, and covers the outline of this chapter. A nominal linear FE model is first created from physical measurements and then used for nonlinear modeling and model updating/uncertainty modeling. The nonlinear modeling uses the analysis results of the uncoupled FE model, which consists of the structure divided at the joints into linear subcomponents. Nonlinear inputs and outputs are defined and the nonlinear feedback matrix is determined in the nonlinear modeling step to create the 0-g nonlinear model. The nonlinear model is then validated using analyses of the 0-g linear coupled FE model at different setpoints.

The second major process consists of model updating and uncertainty model development. For this process, a 1-g linear FE model is required to match the pre-flight test environment; this model must include the suspension system and gravity effects. 1-g data is then used to update the nominal FE model, and also to create a parameter error model. The modeling effort thus results in a geometrically nonlinear model with a parametric uncertainty model.

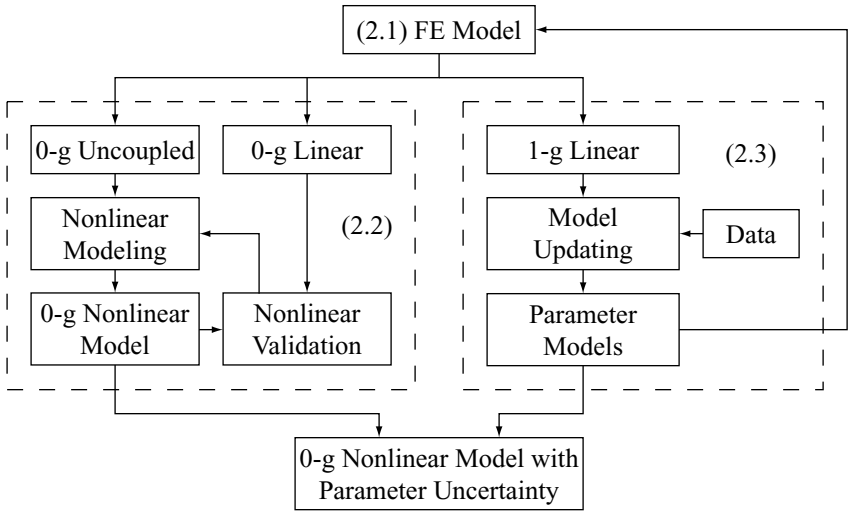

Figure 2.3 Modeling overview. Numbers in parentheses correspond to sections.

2.1 Finite Element and State-Space Model

The modeling process begins by creating a nominal FEM and state-space model. The nominal model is created from physical measurements of geometry and mass, and material properties such as the Young's modulus and mass density. The FE analysis results in frequencies and mode shapes,

$$\Omega = \text{diag}(\omega_i),\ \Phi = \begin{bmatrix} \phi_1 & \dots & \phi_L \end{bmatrix}. \tag{2.3}$$

where ω_j are the natural frequencies, ϕ_j are the corresponding mode shapes, and diag($\cdot$) is a matrix with the vector argument along the main diagonal and all off-diagonal terms zero.

The desired sensors and actuators are specified by the investigator for a given experiment. The sensor outputs can be described as a linear combination of the FE mode shapes:

$$y_d = \left[y_{d_i} \cdots y_{d_M}\right]^{\mathsf{T}} = \hat{C}_d q, y_r = \hat{C}_r \dot{q}, F_{modal} = \hat{B}_u, \tag{2.4}$$

where C_d, C_r and B_u are the displacement sensor, rate sensor observation matrices, and control influence matrix, respectively.

The nonlinear modeling process couples interface accelerations and forces between components. The model is thus augmented with joint interface accelerations a and forces f:

$$a = \hat{C}_a \ddot{q}, F_{interface} = \hat{B}_f F, \tag{2.5}$$

where C_a is the interface acceleration observation matrix and B_f is the interface force influence matrix.

Damping is described by proportional terms representing material damping, wire damping, etc., and by non-proportional terms such as joint friction,

$$D = \text{diag}\,(2\zeta_i \omega_i) + \sum_i c_i \phi_i \phi_i^{\mathsf{T}}, \tag{2.6}$$

where ζ_i is modal damping, and c_i is bearing friction damping. Due to the difficulty of analytically predicting damping, ζ_i and c_i are identified from experiment.

Modal and sensor/actuator information is incorporated into a state space model,

$$\begin{aligned} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} &= \begin{bmatrix} 0 & I \\ -\Omega^2 & D \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ \hat{B}_f & \hat{B}_u \end{bmatrix} \begin{bmatrix} f \\ u \end{bmatrix} \\ \begin{bmatrix} a \\ y_d \\ y_r \end{bmatrix} &= \begin{bmatrix} -\hat{C}_a \Omega^2 & \hat{C}_a D \\ \hat{C}_d & 0 \\ 0 & \hat{C}_r \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} \hat{D}_f & \hat{D}_u \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} f \\ u \end{bmatrix} \end{aligned}, \tag{2.7}$$

which can be written as a set of first order system matrices in the state $x = \left[q^T \; \dot{q}^T\right]^T$,

$$\begin{aligned} \dot{x} &= Ax + \begin{bmatrix} B_f & B_u \end{bmatrix} \begin{bmatrix} f \\ u \end{bmatrix} \\ \begin{bmatrix} a \\ y \end{bmatrix} &= \begin{bmatrix} C_a \\ C_y \end{bmatrix} x + \begin{bmatrix} D_{af} & D_{au} \\ D_{yf} & D_{yu} \end{bmatrix} \begin{bmatrix} f \\ u \end{bmatrix} \end{aligned} . \tag{2.8}$$

Servo loops, state space realizations for anti-aliasing filters, actuator and sensor dynamics, and computer time delays, are appended to this state-space model. The actuator and sensor matrices are scaled from physical units to counts, as seen through the control computer. With this scaling, controllers designed on the state-space model can be directly implemented in hardware.

Three different state-space models are produced, each requiring a separate FE analysis: an uncoupled 0-g model for nonlinear modeling; a linear 0-g for nonlinear validation; and a linear 1-g coupled model for updating and error modeling. Note that essentially the same input deck is used for the 0-g uncoupled, 0-g linear, and 1-g linear models. The use of sets of constraint elements in the FE software package makes it straightforward to connect and disconnect the linear subcomponents.

2.2 Nonlinear Modeling

The nonlinear modeling methodology of this work, shown in Figure 2.4, uses a feedback description of structural coupling through large angle joint rotations. The uncoupled model contains interface inputs and outputs that describe the forces and motions at the joint interfaces. The interface inputs and outputs are manipulated to create a feedback representation of the joint rotations. Geometric variations are parameterized by a global rotation matrix. The resulting system can be manipulated into various control design forms.

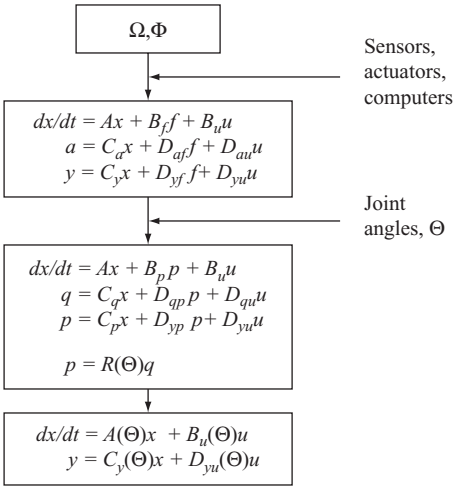

Figure 2.4 Nonlinear modeling methodology

2.2.1 Incorporation of Nonlinear Terms

Input/Output Inversion

The interface accelerations and forces, defined in the uncoupled model, represent coupling at the joints. In the coupled system, accelerations and forces across each joint are related as a function of the joint angles. The state-space model inputs and outputs can be manipulated to permit a description of the angle as a feedback interconnection (Figure 2.5). At each joint, accelerations and forces of the uncoupled nonlinear model are partitioned into complementary pairs:

$$a = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{N1} & a_{N2} \end{bmatrix}^T$$
$$f = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{N1} & f_{N2} \end{bmatrix}^T , \tag{2.9}$$

where (a_{11}, f_{11}), on one component, are defined at the same joint as (a_{12}, f_{12}) on the other component. One of the joint input/output pairs at each joint is inverted, so that the inputs and outputs of the new system are:

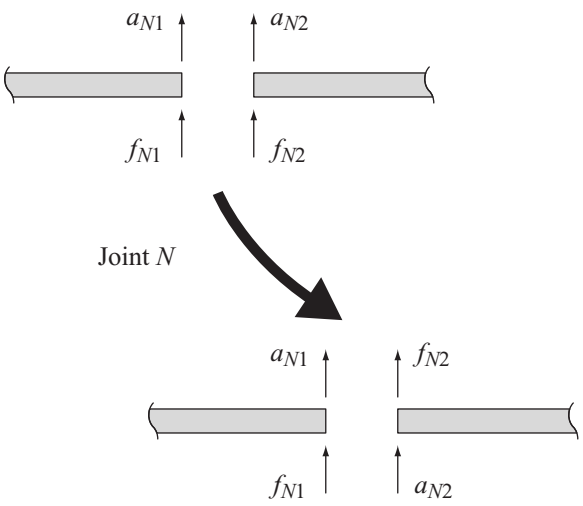

Figure 2.5 Input/output manipulation to transform state space model into feedback representation of geometric nonlinearity.

$$\begin{aligned} q &= \begin{bmatrix} a_{11} & f_{12} & \cdots & a_{N1} & f_{N2} \end{bmatrix}^{\mathsf{T}} \\ p &= \begin{bmatrix} f_{11} & a_{12} & \cdots & f_{N1} & a_{N2} \end{bmatrix}^{\mathsf{T}}. \end{aligned} \tag{2.10}$$

Physically this corresponds to constraining one side of the joint, so that accelerometers are commanded and reaction forces are sensed. The inversion is performed on the state space model by reordering and zero-padding the state space system matrices into the form

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}, \; M = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix}, \tag{2.11}$$

where the 2 subscripts correspond to the inputs/outputs to be inverted. The reordered system is transformed using the equation,

$$\hat{M} = M_{11} + M_{12} M_{22}^{-1} M_{21}, \tag{2.12}$$

which represents the dynamics of the constrained system. The inputs of the constrained system include control forces u, and the independent boundary forces and accelerations p.

The outputs include the sensors y as well as the dependent boundary forces and accelerations q.

Nonlinear model

The constrained system constitutes a linear description of the uncoupled component dynamics. The coupled system dynamics can be determined by specifying that the constrained variables at each joint (input forces and accelerations) are functions of the unconstrained variables (output forces and accelerations) through the rotating joint. Define a global rotation matrix which describes the n_r rotations at all joints,

$$R = \text{block-diag}\left(\begin{bmatrix} 0 & -R_1^{\mathsf{T}} \\ R_1 & 0 \end{bmatrix}, \ldots, \begin{bmatrix} 0 & -R_{n_r}^{\mathsf{T}} \\ R_{n_r} & 0 \end{bmatrix} \right), \tag{2.13}$$

where the block-diag operation places the matrix operands on the diagonal of the resulting matrix. For MIT/MACE-II, the experimental focus of this thesis, there are four rotation matrices R_i corresponding to each gimbal axis,

$$R_i = \begin{cases} \begin{bmatrix} 1 & 0 & 0 \\ 0 & r_i & 0 \\ 0 & 0 & r_i \end{bmatrix} & i = 1, 3 \text{ (X axes)} \\ \begin{bmatrix} r_i & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & r_i \end{bmatrix} & i = 2, 4 \text{ (Z axes)} \end{cases}, \tag{2.14}$$

with r_i as the joint rotation matrix,

$$r_i = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix}. \tag{2.15}$$

The input variables can now be described as a feedback of the output variables,

$$p = Rq\,. \tag{2.16}$$

This equation represents a Linear Fractional Transformation (LFT) through R on the linear system matrices. It can again be equated to a feedback of structural forces and accelerations through the time varying gain R.

Input/Output Reduction

The global rotation matrix R includes non-time-varying terms which relate translational accelerations and forces along the joint axis. For example, consider a gimbal that allows rotation of a single axis. The force and motion components along this axis do not rotate with the gimbal:

$$\ddot{x}_{11} = \ddot{x}_{12}, f_{x_{11}} = -f_{x_{12}}. \tag{2.17}$$

For the gimbal rotation matrices of Equation 2.14, the (1,1) term of the X axis rotation and the (2,2) term of the Z axis rotation are non-time-varying. These terms unnecessarily expand the model input/output dimensionality. The terms can be removed from the system matrices by reordering the interface I/O (p, q) and partitioning into fixed (p_1, q_1) and time-varying (p_2, q_2) elements

$$\begin{gathered} p = \begin{bmatrix} p_1 & p_2 \end{bmatrix}^{\mathsf{T}}, q = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^{\mathsf{T}} \\ p_1 = \operatorname{diag}\left(\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \right) q_1 = K q_1 \end{gathered}, \tag{2.18}$$

where K is the matrix that relates the known interface elements. The above expression is a Linear Fractional Transformation (LFT) on the system matrices.

2.2.2 Control Design Model

The final stage in the nonlinear modeling process is to assemble linearized models. This begins by diagonalizing the feedback gain matrix of Equation 2.16, which is initially skew-symmetric[1]. The feedback gain matrix can be diagonalized using a set of "pointing

matrices" $\{S_l, S_r\}$ that distribute the rotational terms of a diagonal Θ block into the correct elements of the global rotation matrix R:

$$R = S_l \Theta S_r, \tag{2.19}$$

with,

$$\Theta = \text{block-diag}(I_4 \cos\theta_i, I_4 \sin\theta_i, \ldots), i = 1 \ldots 4. \tag{2.20}$$

The matrices S_l and S_r are prepended and appended, respectively, to the system matrices, and Equation 2.20 represents the new Θ block feedback term in the nonlinear equations.

The process for obtaining linearized models is now relatively straightforward. The diagonal Θ block is determined at the desired setpoint as,

$$\Theta_0 = \text{block-diag}(I_4 \cos\theta_{0_i}, I_4 \sin\theta_{0_i}, \ldots), i = 1 \ldots 4, \tag{2.21}$$

and the LFT can be evaluated on the linear system matrices. A entire family of linearized systems can now be efficiently determined.

The feedback matrix can also be expressed as a function of a set of nominal angles and small perturbations $\theta_i = \theta_{0_i} + \delta\theta_i$. Small angle approximations are then used to derive a nominal Θ matrix, Θ_0, and a linear perturbation, $\delta\Theta$. With the system matrices expanded out by duplicating the uncertainty channels, an LFT through the nominal Θ matrix Θ_0 is performed. The resulting linearized system describes the sensitivity of the system to angle changes at the specified setpoint. This information can be used to sensitize an LQG controller to setpoint variations about the design setpoint.

1. . This diagonalization allows the use of structured uncertainty robust control approaches [Zhou et al., 1996], as well as H_∞-based gain scheduling [Apkarian and Gahinet, 1995].

2.3 Model Updating

The next major step in the modeling approach presented in this thesis is model updating and uncertainty modeling. Model updating is the process of correcting models, or adjusting model parameters, by processing dynamic response data from test measurements. The model updating procedure is used for two purposes in this thesis. The first is to fine-tune or update a nominal finite element model to match experimentally obtained frequency responses. The second purpose of model updating is uncertainty modeling, or creating an error model for robust control design based on experimental data.

It is important to note that the model updating procedure occurs independently of the nonlinear modeling described in the previous section. This is because the parameter variations and uncertainty models can be decoupled from the effects of the geometric nonlinearity. This reflected in the block-diagonal nature of the LF feedback matrix of Figure 2.2 and Equation 2.2.

The procedure described in this section can update both modal and physical parameters. Updates to modal parameters include those of frequency, damping and modal residues. Damping is of particular importance because it is difficult to analytically predict damping. Physical parameters such as mass, inertia and Young's moduli can also be updated. This is of particular importance in this thesis because the lack of *in situ* experimental data precludes the use of modal parameter updates.

Three high level choices must be made when a model update is desired. The first is to select a group of *parameters* for updating. These parameters can be modal, such as natural frequencies and damping, or they can be updates to the global mass and stiffness matrices. Good engineering insight must be used to determine a suitable set of update parameters. The second choice concerns the *cost* to be minimized. This can involve minimizing the error between measured and modeled eigenvectors, if detailed mode shape information is available. Alternatively the error between measured and modeled eigenvalues can serve as the cost if only natural frequencies can be determined. The final step in

the model update process is to choose a *search algorithm*. Depending on the nature of the cost and parameter selection, this can simply be a least-squares type fit, or it may require the use of an iterative nonlinear optimization routine.

2.3.1 Parameter Selection and Realization

The updating method developed in this thesis allows for the update of modal and physical parameters. Modal parameters include natural frequencies, damping ratios and residues. Damping is especially important because the analytical prediction of damping is difficult [Ingham and Crawley, 2001]. Physical parameters include mass, inertia and stiffness. The update of physical parameters is particularly important for a geometrically nonlinear system, where modal parameters do not directly map over to other setpoints.

The LF form is used to represent parameter changes,

$$\begin{aligned} \dot{x} &= Ax + B_p p + B_u u \\ q &= C_q x + D_{qp} p + D_{qu} u \\ y &= C_y x + D_{yp} p + D_{yu} u \end{aligned}, \tag{2.22}$$

so that the updated model is determined by applying the parameter update matrix Δ,

$$p = \Delta q . \tag{2.23}$$

As was the case with the nonlinear feedback matrix Θ, the parameter update matrix Δ can be chosen to be diagonal via a set of "pointing matrices",

$$\Delta = \text{diag}\ (\delta_i I_i),\ i = 1 \ldots n_p , \tag{2.24}$$

where n_p is the total number of independent update parameters. The remainder of this subsection details the realization of the modal and physical parameter updates into the LF form, given the nominal state-space model (Section 2.1),

$$A = \begin{bmatrix} 0 & I \\ -\Omega^2 & -Z\Omega \end{bmatrix}, B_u = \begin{bmatrix} 0 \\ \Phi_{act}^{\mathsf{T}} \end{bmatrix}, \begin{bmatrix} C_d \\ C_r \\ C_a \end{bmatrix} = \begin{bmatrix} \Phi_d & 0 \\ 0 & \Phi_r \\ -\Phi_a\Omega^2 & -2\Phi_a Z\Omega \end{bmatrix}, D_{yu} = \begin{bmatrix} 0 \\ 0 \\ \Phi_a\Phi_{act}^{\mathsf{T}} \end{bmatrix}, \quad (2.25)$$

so that there are input forces, B_u, and three different sensor groups; displacement, C_d, rate, C_r, and acceleration, C_a.

The realization of the parameter update inputs and outputs, B_p and C_q, proceeds like a FE assembly process. A parameter update input and output is realized for each element, and then the element realizations are then incorporated into the global parameter update inputs and outputs at the appropriate degrees of freedom. Where possible, parameter updates are realized as multiplicative variations. This improves the scaling of the gradients (which are necessary for the minimization routine), and also permits constraints to be applied as percentages. However, some parameters (e.g.: inertia terms) have a zero nominal value and therefore an additive variation must be realized.

Modal Parameter Updates

Frequency Update. The frequency term is located in the (2,1) partition of the A matrix. The parameter variation matrices B_{p_i} and C_{q_i} act as "pointing matrices" to pick out the appropriate element of the baseline A matrix,

$$B_{p_i} = \begin{bmatrix} 0 \\ 1_i \end{bmatrix}, C_{q_i} = \begin{bmatrix} 1_i & 0 \end{bmatrix}, p_\omega = \delta_i q_\omega, 1_i = [0 \ldots 1 \ldots 0]^{\mathsf{T}}, \quad (2.26)$$

where 1_i is a vector with the i^{th} element equal to 1 and zero everywhere else. The updated frequency for mode i is then,

$$\hat{\omega}_i = \omega_i \sqrt{1 + \delta_i}. \quad (2.27)$$

Note that the square root operation is necessary because the frequency term appears with a square in the (2,1) partition of the A matrix.

Damping Update. As with the frequency update, the parameter variation matrices act to pick out the corresponding term in the A matrix,

$$B_{p_i} = \begin{bmatrix} 0 \\ 1_i \end{bmatrix}, \; C_{q_i} = \begin{bmatrix} 0 & 1_i \end{bmatrix}, \; p_\zeta = \delta_i q_\zeta, \; 1_i = [0 \ldots 1 \ldots 0]^\mathsf{T}, \tag{2.28}$$

and the updated damping ratio is given by,

$$\hat{\zeta}_i = \frac{\zeta_i (1 + \delta_i)}{\hat{\omega}_i}. \tag{2.29}$$

Note that the frequency term is the updated frequency, $\hat{\omega}_i$. This must be used when the modal frequencies are being simultaneously updated.

Global Stiffness Matrix Update

The global stiffness matrix is updated with a differential stiffness matrix,

$$\hat{K} = K + \Delta K. \tag{2.30}$$

The differential stiffness matrix ΔK is the sum of individual element updates ΔK_i,

$$\Delta K = \sum_{i=1}^{n_p} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \Delta K_i & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.31}$$

Each element update ΔK_i is assumed linear in the update parameter, δ_i, and can then be decomposed as,

$$\Delta K_i = b_i \delta_i c_i. \tag{2.32}$$

This decomposition can be performed analytically for certain simple elements, but in general a Singular Value Decomposition (SVD) can be used to find b_i and c_i. The second order state space form for the updated model is,

$$\begin{aligned} M\ddot{x} + (K + \Delta K)x &= B_u u \\ y_d = C_d x, y_r = C_r \dot{x}, y_a &= C_a \ddot{x} \end{aligned} \tag{2.33}$$

This is transformed to modal coordinates, ψ, with the diagonal matrix of eigenvalues, Ω, and the matrix of eigenvectors, Φ,

$$\begin{aligned} I\ddot{\psi} + \Omega^2\psi &= -\Phi^T \Delta K \Phi \psi + \Phi^T B_u u \\ y_d = C_d \Phi \psi, y_r = C_r \Phi \dot{\psi}, y_a &= C_a \Phi \ddot{\psi} \end{aligned} \tag{2.34}$$

From Equation 2.31, the differential contribution to the modal stiffness, $\Phi^T \Delta K \Phi$, can be rewritten,

$$\Phi^T \Delta K \Phi = \Phi^T \left(\sum_{i=1}^{n_p} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \Delta K_i & 0 \\ 0 & 0 & 0 \end{bmatrix} \right) \Phi = \sum_{i=1}^{n_p} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \Phi_i^T \Delta K_i \Phi_i & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{2.35}$$

where Φ_i are the rows of the eigenvector matrix Φ corresponding to the element degrees of freedom for parameter *i*. Each elemental modal stiffness is decomposed according to Equation 2.32 and the summation written as a matrix decomposition,

$$\Phi^T \Delta K \Phi = \sum_{i=1}^{n_p} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \Phi_i^T b_i \delta_i c_i \Phi_i & 0 \\ 0 & 0 & 0 \end{bmatrix} = B_K \Delta_K C_K, \tag{2.36}$$

where,

$$B_K = \begin{bmatrix} 0 & & & \\ \Phi_1^T b_1 & 0 & & \\ & \Phi_2^T b_2 & \cdots & \\ 0 & & \Phi_N^T b_N & \\ & & & 0 \end{bmatrix}, \; C_K = \begin{bmatrix} 0 & c_1 \Phi_1 & & \\ & & c_2 \Phi_2 & 0 \\ & 0 & \vdots & \\ & & & c_N \Phi_N \; 0 \end{bmatrix}, \tag{2.37}$$

Following the definition of Φ_i, the elements of B_K and C_K are populated according to the appropriate global degrees of freedom. The parameter updates are thus aggregrated into a single parameter update matrix as,

$$\Delta_K = \text{block-diag}(\delta_1 I_1, \delta_2 I_2, \cdots, \delta_N I_N) . \tag{2.38}$$

The identity matrices, I_i, of varying dimension reflect the fact that different elements will be decomposed into matrices of different sizes. With this decomposition, and defining the stiffness parameter variation output as $q = C_K\psi$, the updated system can be written as,

$$\begin{gathered} \ddot{I\psi} + \Omega^2\psi = -B_K p_K + \Phi^{\mathsf{T}} B_u u \\ q_K = C_K\psi \\ p_K = \Delta_K q_K \\ y_d = C_d\Phi\psi, y_r = C_r\Phi\dot{\psi}, y_a = C_a\Phi\ddot{\psi} \end{gathered} . \tag{2.39}$$

This is readily converted to the LF form of Equation 2.22 with $B_p = -B_K$, $C_q = C_K$ and $D_{qp} = 0$.

Global Mass Matrix Update Realization. An equivalent update procedure is applicable to the global mass matrix of the nominal FEM, with the update given by,

$$\hat{M} = M + \Delta M . \tag{2.40}$$

The update applied to the second order system in modal coordinates is,

$$\begin{gathered} \ddot{I\psi} + \Omega^2\psi = -\Phi^{\mathsf{T}}\Delta M\Phi\ddot{\psi} + \Phi^{\mathsf{T}} B_u u \\ y_d = C_d\Phi\psi, y_r = C_r\Phi\dot{\psi}, y_a = C_a\Phi\ddot{\psi} \end{gathered} . \tag{2.41}$$

The differential contribution to the global mass matrix can be decomposed in the same manner as with the stiffness contribution,

$$\Delta M = B_M \Delta_M C_M , \tag{2.42}$$

and Equation 2.41 can be rewritten as,

$$\begin{aligned} I\ddot{\psi} + \Omega^2\psi &= -B_M\Delta_M C_M\ddot{\psi} + \Phi^T B_u u \\ y_d = C_d\Phi\psi,\, y_r &= C_r\Phi\dot{\psi},\, y_a = C_a\Phi\ddot{\psi} \end{aligned} . \tag{2.43}$$

By defining the mass parameter variation output as $q_M = C_M\ddot{\psi}$ this can be rewritten as,

$$\begin{aligned} I\ddot{\psi} + \Omega^2\psi &= -B_M p_M + \Phi^T B_u u \\ q_M &= -C_M\Omega^2\psi - C_M B_M p_M + C_M B_u u \\ p_M &= \Delta_M q_M \\ y_d = C_d\Phi\psi,\, y_r &= C_r\Phi\dot{\psi},\, y_a = C_a\Phi\ddot{\psi} \end{aligned} . \tag{2.44}$$

This is readily converted to the LF form of Equation 2.22 with $B_p = -B_M$, $C_q = -C_M\Omega^2$, $D_{qp} = -C_M B_M$ and $D_{qu} = C_M B_u$.

Stiffness and Mass Element Decomposition

The stiffness and mass update realizations provide a powerful updating framework that can be applied to many kinds of elements. The final step for realizing the parameter update input and output is to determine the decomposition for individual elements, $\Delta K_i = b_i\delta_i c_i$, and $\Delta M_i = b_i\delta_i c_i$. This decomposition is performed here for three NASTRAN different elements.

The CELAS element is a spring that acts to add stiffness between two degrees of freedom,

$$K_i = \begin{bmatrix} k_i & -k_i \\ -k_i & k_i \end{bmatrix} . \tag{2.45}$$

The parameter to update is the spring stiffness, k_i. The updated spring stiffness is defined as,

$$\tilde{k}_i = k_i(1 + \delta_i) , \tag{2.46}$$

so that the updated element stiffness matrix becomes,

$$\tilde{K}_i = \begin{bmatrix} k_i & -k_i \\ -k_i & k_i \end{bmatrix} + \begin{bmatrix} k_i & -k_i \\ -k_i & k_i \end{bmatrix} \delta_i = K_i + b_i \delta_i c_i \,. \tag{2.47}$$

where,

$$b_i = \sqrt{k_i} \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \; c_i = \sqrt{k_i} \begin{bmatrix} 1 & -1 \end{bmatrix}. \tag{2.48}$$

The CBAR element is a Timoshenko beam that incorporates several geometric and material properties. This example illustrates a decomposition for the Young's Modulus, E_i. Because the Young's Modulus does not appear linearly in the CBAR stiffness matrix, an SVD is performed on the stiffness matrix,

$$K_i = U \Sigma V^{\mathsf{T}}, \tag{2.49}$$

where S is a diagonal matrix of singular values.

$$\tilde{K}_i = K_i + \Delta K_i = U \Sigma^{1/2} (1 + \delta_i) I^{(6)} \Sigma^{1/2} V^{\mathsf{T}} = K_i + b_i \delta_i c_i \,, \tag{2.50}$$

where, $I^{(6)}$ is the 6x6 identity matrix, and,

$$b_i = U \Sigma^{1/2}, \; C_i = \Sigma^{1/2} V^{\mathsf{T}}. \tag{2.51}$$

The updated Young's modulus is given by,

$$\tilde{E}_i = E_i (1 + \delta_i) \,, \tag{2.52}$$

Note that this is exact if the shear factor is equal to zero (the Young's Modulus is linear in the stiffness matrix of a Bernoulli-Euler beam).

The CONM2 mass element add mass and inertia to a single node,

$$M = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{11} & -I_{21} & -I_{31} \\ 0 & 0 & 0 & -I_{21} & I_{22} & -I_{32} \\ 0 & 0 & 0 & -I_{31} & -I_{32} & I_{33} \end{bmatrix}. \tag{2.53}$$

The decomposition acts to pick the separate terms, m, I_{11}, I_{21}, I_{22}, I_{31}, I_{32} and I_{33},

$$b_i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 1 \end{bmatrix}, \; c_i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}^{\mathsf{T}}. \tag{2.54}$$

The updated terms are given by,

$$\begin{aligned} \tilde{m} &= m(1+\delta_{i1}) \\ \tilde{I}_{11} &= I_{11} + \delta_{i2} \qquad \tilde{I}_{31} = I_{31} + \delta_{i5} \\ \tilde{I}_{21} &= I_{21} + \delta_{i3} \qquad \tilde{I}_{32} = I_{32} + \delta_{i6} \\ \tilde{I}_{22} &= I_{22} + \delta_{i4} \qquad \tilde{I}_{33} = I_{33} + \delta_{i7} \end{aligned}. \tag{2.55}$$

Note that whereas the mass term uses a multiplicative update, the inertia terms use an additive update. This is to allow for initially zero entries in the inertia terms. The update parameter matrix is assembled as,

$$\Delta_{\text{CONM}} = \text{block-diag}(\delta_{i1} I^{(3)}, \delta_{i2}, \delta_{i3} I^{(2)}, \delta_{i4}, \delta_{i5} I^{(2)}, \delta_{i6} I^{(2)}, \delta_{i7}). \tag{2.56}$$

The identity matrices reflect the fact that some of the parameters must be repeated in the update parameter matrix.

2.3.2 Cost

The updating cost is based on the error between the estimated frequency response, $\hat{G}(j\omega)$, and the experimentally measured frequency response, $G(j\omega)$,

$$\begin{aligned} J(j\omega) &= \frac{1}{2} f^{H}(j\omega) f(j\omega) \\ f(j\omega) &= W(j\omega)[\log(G(j\omega)+\mu) - \log(\hat{G}(j\omega)+\mu)] \end{aligned}, \tag{2.57}$$

and the estimated frequency response is given by,

$$\hat{G}(j\omega) = G_{yu}(j\omega) + G_{yp}(j\omega)(I - \Delta G_{qp}(j\omega))^{-1}\Delta G_{qu}(j\omega), \tag{2.58}$$

where $G_{yu}(j\omega) = C_y(j\omega I - A)^{-1}B_u + D_{yu}$, and so forth for $G_{yp}(j\omega)$, $G_{qu}(j\omega)$ and $G_{qp}(j\omega)$. The matrix W is a weighting matrix, and μ is a "sensor noise" term that can de-emphasize low signal-to-noise regions of the experimental frequency response. The logarithmic cost is used to place emphasis on zeros as well as poles.

The assumption in choosing a cost based on frequency responses is that the modal frequencies and residues of the measured and modeled responses must be reasonably "close", or within approximately 10% of each other. This is to ensure that the search algorithm will be able to determine the appropriate "direction".

2.3.3 Search Method

The MATLAB function lsqnonlin.m is selected as the search method. The function incorporates state-of-the-art nonlinear least squares minimization algorithms. Furthermore it allows for constraints to be placed on parameter changes; parameters were typically allowed to deviate by no more than 50%. To facilitate the solution process, gradients to the cost are also computed [Blaurock, 2001].

2.3.4 Model Updating Procedure

The model updating procedure begins by generating an initial FEM and associated state-space model. Uncertain parameters are selected for updating, and the parameter update inputs and outputs are realized. The optimization iteratively computes the frequency response of the updated system, computes the updating cost and gradient, and determines a set of parameter updates.

The parameter update realization is a linear approximation to true parameter variations. As the parameter updates increase in magnitude, the LF updated model will differ from a reanalyzed FE model. There are several factors that lead to this discrepancy:

- The state-space model uses a reduced set of eigenvectors, thus not capturing all the strain energy of the system exactly. This reduction potentially eliminates "directions" along which the true system lies.
- The derivatives of the eigenvectors with respect to parameter changes are not considered.
- Parameters selected for updates may not contribute linearly to the mass or stiffness matrix, for example in the case of the Young's modulus in the Timoshenko beam element.
- There may be a nonlinear step in the FEM analysis, such as a gravity pre-stiffening.

To minimize these discrepancies, a complete FE analysis should be performed to re-compute the frequencies and mode shapes from the updated stiffness and mass matrices after the nonlinear optimization. The update procedure must proceed as an iteration of nonlinear optimizations and FE analyses, and is continued until the cost converges across the FE analyses. The updating procedure is summarized in Figure 2.6.

One problem that can arise from parameter updating is that the update may result in non-physical parameters. Extreme cases include negative Young's moduli or spring stiffnesses. This indicates that the parameter space does not encompass the true solution. One reason is that the set of updating parameters has been chosen poorly; either too few param-

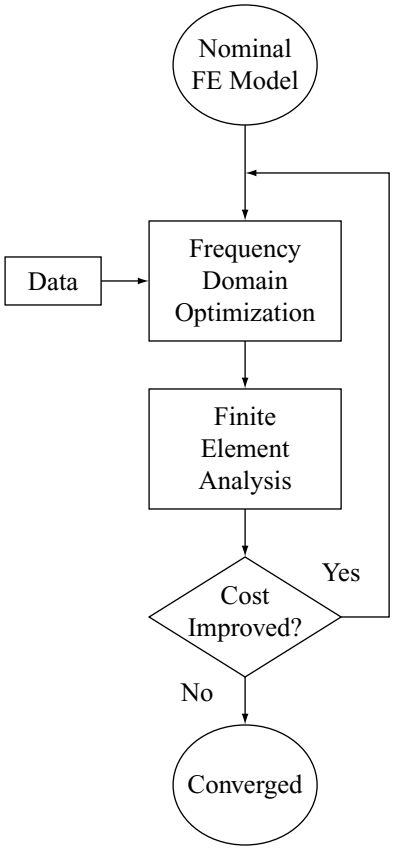

Figure 2.6 Iterative model updating algorithm.

eters, or parameters that do not have a strong influence. Another reason is that the finite element mesh is too coarse to accurately capture the true dynamic response.

This problem can be addressed by verifying the parameter update with frequency responses corresponding to alternative inputs and outputs. The nonlinear optimization is performed with the sensors and actuators used for control design, but an additional step of verification is performed by checking that the cost corresponding to the frequency responses of the unused sensors and actuators has also decreased. If the cost has increased, this is generally indicative that the solution has produced a non-physical update.

Another problem that can arise from parameter updating occurs when the parameter space is non-uniquely determined. In this case, multiple solutions may minimize the updating

cost. This is typically manifested during the iteration as parameters jumping around and not converging to a single parameter vector.

Care must thus be taken to choose the update parameter set. If the set is underdetermined, the solution may converge to a non-physical update. If the set is overdetermined, no unique solution will be found. Different parameters musts be individually examined to determine their effect on the system response.

2.3.5 Parameter Error Models

There are numerous sources of error that can cause significant differences between modeled and measured responses. One categorization for these sources of error considers whether the error is fixed or variable. Manufacturing tolerances allow some variation in the geometry of the structure; the error associated with this is fixed, since the components of the structure are not modified. Material properties such as stiffness and density, as quoted by "book values", may differ from the actual material used in the construction of the system, and the resulting errors may also be considered *fixed* (for the same operating environment).

On the other hand errors may be *variable*. Dis-assembly/re-assembly cycles may cause joints to lock at slightly different configurations, and the same preload may not be consistently achieved. These errors will produce a different dynamic response through each assembly cycle and are thus deemed variable. Other variable sources of error include thermal and environment effects.

The objective of parameter error modeling is to capture both of these kinds of errors in the form of an uncertainty model that is suitable for robust control. A particular challenge for space-based systems is the lack of *in situ* testing before flight. Uncertainty models can be developed from pre-flight 1-g testing, but a process is required to propagate the 1-g uncertainty model to 0-g for on-orbit operation.

The approach for uncertainty modeling is motivated by the success of the Middeck Active Control Experiment (MACE-I) program [Miller et al., 1996], a Space Shuttle flight experiment that flew on STS-67 in March 1995. Modal parameter (frequencies and damping ratios) identification was performed in 1-g for several datasets. These sets of modal parameters were combined into a parameter uncertainty model containing means and variances, capturing both fixed and variable errors. This modal uncertainty model was then propagated to 0-g by correlating the modes of the 1-g FE model to the modes of the 0-g FE model.[1]

In MACE-I, the modal parameter description was used to describe linear systems in 1-g and 0-g. However, the geometrically nonlinear nature of the problem addressed in this thesis precludes the use of modal parameters. A modal uncertainty model valid at one setpoint will not be valid at another setpoint since the modal structure of the system will be different.

One possible solution to generating an uncertainty model for geometrically nonlinear systems is to create modal uncertainty models at different linearized setpoints. However, for MIT/MACE-II, the geometrically nonlinear nature of the 0-g control task cannot be simulated prior to flight. The gimbal motors do not have the authority to command the large angles necessary to realize significant nonlinear behavior in 1-g. This characteristic of not being able to achieve full 0-g nonlinearity is shared between many space-based systems. One reason is the lack of controller authority described above. Another reason is the fact that stringent mass requirements result in structures that cannot support their own weight in 1-g. This can preclude testing of the fully assembled (and thus geometrically nonlinear) system.

For an uncertainty modeling process to be suitable for the problems addressed in this thesis, it must satisfy the following particularly challenging requirements. First, the uncer-

1. Note that the 1-g testing environment (including a pneumatic suspension system) had been designed to simulate the 0-g testing environment as closely as possible. A clear correspondence between 1-g modes and 0-g modes could then be established for propagating the modal parameters.

tainty model must be able to represent uncertainties throughout the nonlinear operational space. Secondly, this model must be generated through testing of a linear system since nonlinear operational is not achievable before flight.

Approach

The uncertainty modeling approach used in this thesis extends the MACE-I approach. The basic methodology of describing both fixed and variable errors by generating a parameter database with multiple datasets is employed. The issues of representing geometric nonlinearity, and identifying uncertainty models with only linear testing, is addressed by the use of the *physical* parameter description developed for model updating. The statistics of physical parameters such as stiffness, mass and inertia are assumed invariant through the 0-g nonlinear range of operation, and also invariant between 1-g and 0-g.

Since the uncertainty model consists of *model* parameters, this approach is limited in the sense that it cannot capture *unmodeled* dynamics. For example, in MACE-I, the limitation to control turned out to be a nonlinearity due to the reaction wheel stiffening as the speed of the wheels increases. However, design experience with the 1-g MIT/MACE-II test article indicates that the dominant source of error, for this experiment, is in fact a variable stiffness error that can be represented by this uncertainty structure. The source of this error is further discussed in Section 5.2.3.

An additional limitation of the error modeling process arises from the use of a frequency response cost function. The calculation of the frequency response (via Discrete Fourier Transform) produces an implicit linearization, which has the effect of smearing nonlinearities in the time response into an average frequency domain response. The Fourier Transform also introduces biasing errors [Ljung, 1999]. In contrast, the MACE-I error model was developed using a time domain identification technique (Extended Kalman Filtering [Campbell, 1996]) which avoids errors introduced by the DFT. As an additional benefit, the EKF produces estimates of parameter variances that can be used to evaluate confi-

dence in the identified parameters. In general, the relative importance of nonlinear effects must be quantified on a system-by-system basis, for example by comparing the results of time and frequency domain analyses (as suggested by the MACE-I approach). For the MACE MBP the MACE-I results indicated that the nonlinear effects were weak, and thus the frequency response based model was acceptable. However, a time domain (EKF) identification process could be developed from the uncertainty realization of Equations 2.22 and 2.23.

Application to Robust Control Design

A simplification can be made by re-scaling the parameter update inputs and outputs to reflect experimentally determined parameter variances, similar to the pointing matrices used to diagonalize the feedback rotation matrix in Section 2.2.1. The resulting uncertainty model can be incorporated into robust control design in several ways:

The uncertainty model can be applied to *synthesis* techniques for control. The parameter update inputs and outputs can be used as sensitivity weights for Sensitivity Weighted Linear Quadratic Gaussian methods to de-sensitize the controller to parametric errors. Alternatively a sampling of the parameter uncertainty space can be made to generate several design models for the Multiple Model method.

The uncertainty model is also useful for control *analysis*. The form of the uncertainty model makes it suitable for structured singular value or μ-analysis methods. An alternative method, used later in this thesis, would be to sample the uncertainty space and perform linear analyses at each sample.

2.4 Chapter Summary

This chapter has presented a methodology for modeling a geometrically nonlinear system for robust control design. The methodology incorporates both geometric nonlinearity and a parametric error model. The features of the modeling process include:

- The LF form allows uncertainty and nonlinearity to be modeled *independently.*
- The LF form allows linearized models to be rapidly generated without repeated FE analysis. This is particularly useful for controller validation, where closed loop analysis must be performed at many linearized setpoints.
- The linear state space form for both model updating and nonlinear modeling allows additional dynamics (such as servos and sensor dynamics) to be trivially incorporated, which is crucial to creating a description of the dynamics as seen by the control computer.
- Both model updating and nonlinear modeling are compatible with commercial FEM packages. The use of a commercial FEM package such as NASTRAN allows complex physical structures to be captured accurately, with tractable modeling effort. Such models are typically available for flight systems.
- Errors in the model are represented by physical parameter uncertainties, allowing for direct extrapolation from experimental to *in situ* environments across the nonlinear configuration space.
- The primary disadvantage is that the higher order rate-dependent terms in the equations of motion are not described. This is not critical to the slow time variations of geometrically nonlinear systems.

Chapter 3

LINEAR DESIGN TOOLS FOR SETPOINT CONTROL

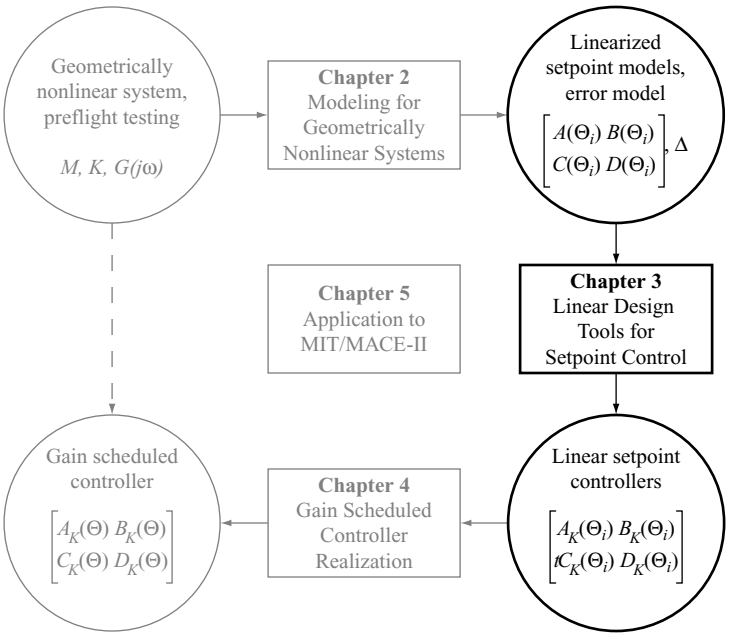

Figure 3.1 Linear design tools for gain scheduling. Given linearized setpoint models and an error model, linear design tools are used to create linear setpoint controllers.

As an integral part of the proposed gain scheduling methodology, linear design tools are used to create linear controllers from linearized setpoint models of the full nonlinear system. This is represented in Figure 3.1. The linearized models represent both the geometric nonlinearity and the uncertainty of the geometrically nonlinear structure. The linear design tools use these setpoint models to design linear controllers, which will be incorporated into a gain scheduled controller in Chapter 4.

The tools presented in this chapter consist of synthesis and analysis tools. Synthesis tools are used to create model-based controllers from the linearized state-space control design models developed in Chapter 2. Analysis tools are used to predict and evaluate the performance and robustness of the linear controllers on the linearized models. Error models and experimental data are incorporated in the design process to improve the robustness of the controllers.

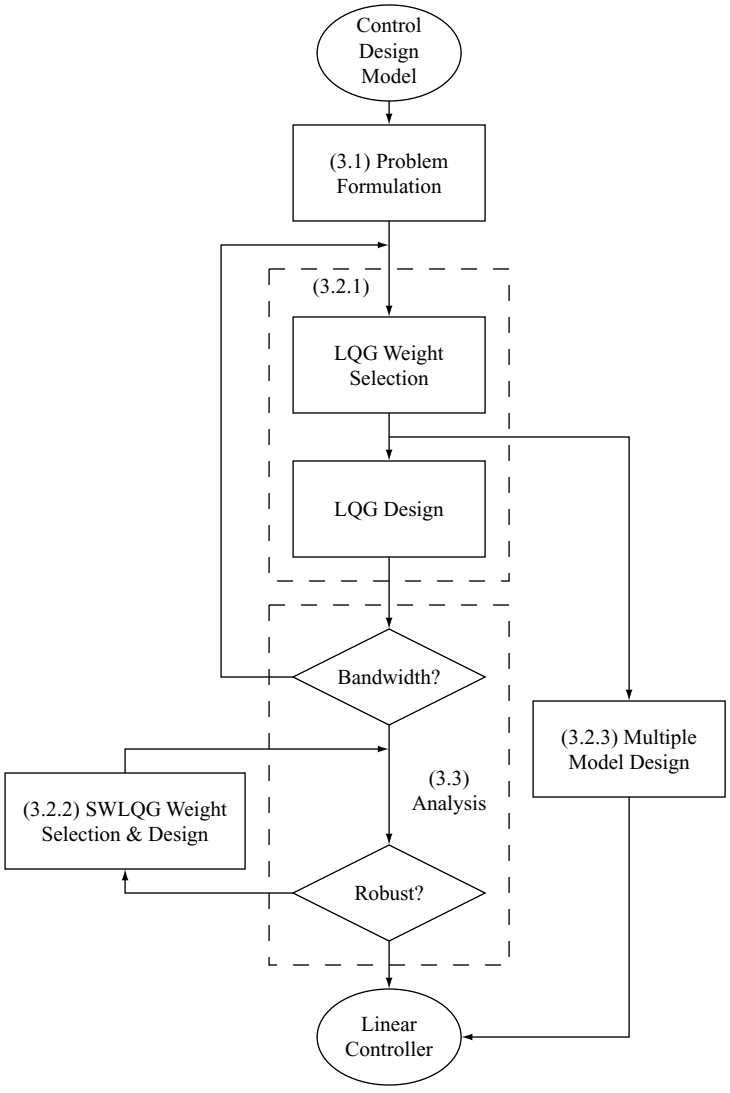

Figure 3.2 Flowchart for linear control design

The outline of this chapter can be represented as a flowchart of the linear control design process, as shown in Figure 3.2. Starting with a linearized control design model (developed in Chapter 2), the control problem must be formulated in terms of performance variables, disturbance inputs, control actuators and sensors, and a performance specification. The next step in the design process is to select weights for Linear Quadratic Gaussian (LQG) design [Kwakernaak and Sivan, 1972]. This must be performed iteratively as performance cannot be explicitly specified for LQG design. Once the LQG weights have been selected to achieve the desired level of performance, a robustness analysis is performed. If the closed loop system is robust to the uncertainties present in the error model, the design is complete. If the closed loop system fails to meet robustness requirements, Sensitivity Weighted Linear Quadratic Gaussian (SWLQG) methods can be used [Sesak, 1974]. As is the case with LQG weights, the selection of sensitivity weights is an iterative process requiring analysis and re-design at each iteration. This iterative design process may not converge if the performance requirements are too stringent or the uncertainty in the model is too great. In this case the performance requirements must be re-evaluated.

There are two iterative loops in the control design process; the selection of control design weights for LQG and SWLQG. Graphic heuristics that assist the control designer in selecting these weights are introduced in order to reduce the iterations in these potentially time-consuming tasks. For LQG, the frequency responses of the weighted channels of the open loop system can be compared to predict the closed loop response without explicitly computing the closed loop response. For SWLQG, the sensitivity weights are reformulated into frequency dependent weightings and a similar graphical analysis can be performed.

An alternative technique of designing linear controllers also reviewed in the synthesis tools section. The Multiple Model method [Ashkenazi and Bryson, 1982] incorporates LQG weights and uses a nonlinear minimization to optimize a linear controller over multiple plants that sample the uncertainty space. This proven method of designing robust lin-

ear controllers is documented here as a Linear Time Invariant (LTI) benchmark that will be used to evaluate the gain scheduled methodology of this thesis.

Because the synthesis tools do not the control designer to explicitly specify the robustness and performance requirements, analysis procedures are required to validate controller designs. These linear analysis tools are reviewed in the second part of this chapter, and consist of procedures for verifying both performance and robustness.

3.1 Control Problem Formulation

The linear control design problem can generally be formulated in a state-space realization. The uncontrolled system is defined with performance metrics (z), actuators (u), sensors (y), and process and sensor noises (w),

$$\begin{aligned} \dot{x} &= Ax + B_w w + B_u u \\ z &= C_z x + D_{zw} w + D_{zu} u \\ y &= C_y x + D_{yw} w + D_{yu} u \end{aligned} . \tag{3.1}$$

The control problem is to minimize a norm of the performance metric, $J = \|z\|_n$, subject to a disturbance w, with the control law $u = K(j\omega)y$. Typically the norm is specified as either H_2 or H_∞. The structure of this general problem allows frequency-dependent weightings to be included by augmenting the open loop system with dynamic weights. Frequency dependent weightings can be used to reject error within certain frequency range, to reflect frequency contents of disturbances, or also to de-emphasize control where the model uncertainty is greater [Zhou et al., 1996].

The frequency response of the open loop system is represented by the matrix,

$$G(j\omega) = \begin{bmatrix} G_{zw}(j\omega) & G_{zu}(j\omega) \\ G_{yw}(j\omega) & G_{yu}(j\omega) \end{bmatrix}, \tag{3.2}$$

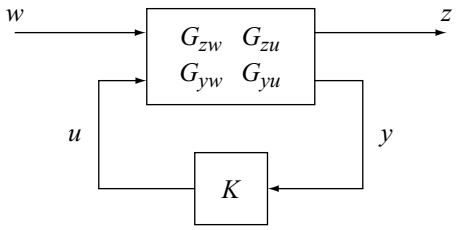

Figure 3.3 Control problem.

where $G_{yu}(j\omega) = C_y(j\omega I - A)^{-1}B_u + D_{yu}$, and so forth for $G_{yw}(j\omega)$, $G_{zu}(j\omega)$ and $G_{zw}(j\omega)$. The control design results in a state-space controller, $K(j\omega)$. The controller is used in a feedback loop, shown i n Figure 3.3, and is of the form,

$$\begin{aligned} \dot{x_K} &= A_K x_K + B_K y \\ u &= C_K x \\ K(j\omega) &= C_K(j\omega I - A_K)^{-1}B_K \end{aligned} \quad . \tag{3.3}$$

The closed loop disturbance to performance response of the system is then,

$$G_{zw}\big|_{\mathrm{CL}}(j\omega) = G_{zw}(j\omega) + G_{zu}(j\omega)K(j\omega)(I - G_{yu}(j\omega)K(j\omega))^{-1}G_{yw}(j\omega)\,. \tag{3.4}$$

G_{zw} and $G_{zw}\big|_{\mathrm{CL}}$ are the open and closed loop disturbance to performance frequency responses respectively. For the sake of brevity these will now be referred to as the open and closed loop performance frequency responses.

To illustrate the use of the linear design tools, a 1-g MACE-II system will be used as a sample problem. The performance output and control sensor are collocated at the primary rate gyro to create a output-analogous control problem, such that $G_{zw} = G_{yw}$ and $G_{zu} = G_{yu}$. Furthermore the actuator is rigidly connected to the control sensor at the primary gimbal. The disturbance is located on the opposite side of the MBP at the secondary gimbal. The frequency responses for this system are shown in Figure 3.4. Both frequency responses are dominated by lightly damped poles. The G_{zw}/G_{yw} frequency response loses phase quickly in comparison to the G_{zu}/G_{yu} frequency response, because of the flexibility between the disturbance source and the sensors. The phase loss in the G_{zu}/G_{yu} frequency

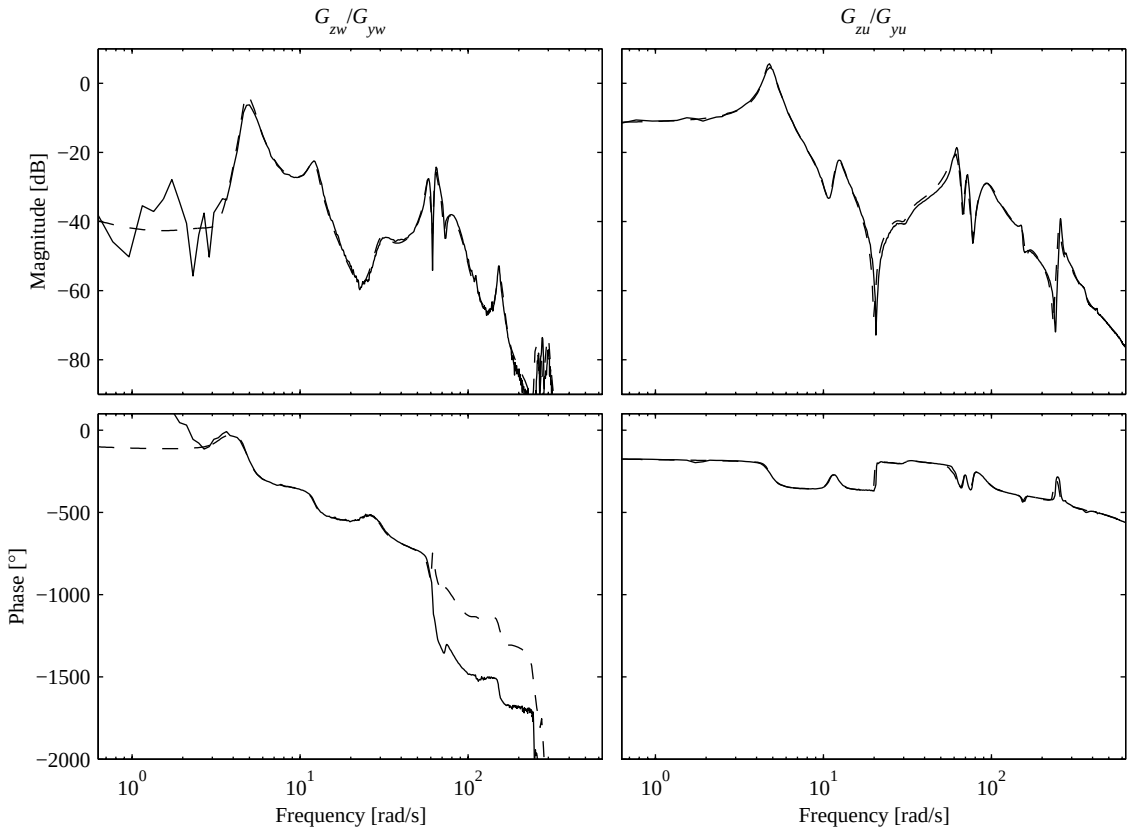

Figure 3.4 Sample plant (1-g MACE-II control problem). Experimental data used as an evaluation model is shown as a solid line, and the design model is shown as a dashed line.

response is dominated by the 2 ms time delay of the real-time control computer. This phase loss is small compared to the phase loss in the G_{zw}/G_{yw} frequency responses.

3.2 Synthesis Tools

This section describes synthesis tools used to generate a linear controller from linearized models. LQG and SWLQG are reviewed, and graphical heuristics are developed to assist the control designer in selecting control design weights. Both LQG and SWLQG design techniques will be used in Chapter 4 as a part of the gain scheduled design framework. The Multiple Model design technique is also reviewed as a benchmark LTI control design technique; the performance of Multiple Model controllers will be compared to the performance of gain scheduled controllers in Chapter 5.

3.2.1 Linear Quadratic Gaussian

The LQG problem [Kwakernaak and Sivan, 1972] solves for a controller for the system of Equation 3.1. A brief summary of the LQG problem and solution is now presented following [Skogestad and Postlethwaite, 1996]. The LQG problem is to find the control $u(t)$, that minimizes the following cost,

$$J = E\left\{ \lim_{T \to \infty} \frac{1}{T} \int_0^T [x^\mathsf{T} R_{xx} x + 2x^\mathsf{T} R_{xu} u + u^\mathsf{T} R_{uu} u] dt \right\}, \tag{3.5}$$

for the system of Equation 3.1 subject to disturbance signals that are white noise processes with covariances given by,

$$E\{w(t)w(\tau)^\mathsf{T}\} = W\delta(t-\tau). \tag{3.6}$$

where $E\{\cdot\}$ is the expectation operator, and $\delta(t-\tau)$ is the delta function. The following constraints must be satisfied,

$$D_{zw} = 0, \tag{3.7}$$

$$R = \begin{bmatrix} R_{xx} & R_{xu} \\ R_{xu}^\mathsf{T} & R_{uu} \end{bmatrix} \geq 0, \; R_{uu} > 0, \tag{3.8}$$

$$V = \begin{bmatrix} V_{xx} & V_{xy} \\ V_{xy}^\mathsf{T} & V_{yy} \end{bmatrix} \geq 0, \; V_{yy} > 0, \tag{3.9}$$

The solution to the LQG problem decouples via the Separation Theorem into two dual problems. The first problem determines state feedback gains to be implemented in the control law $u(t) = -K_r x(t)$. The second problem determines filter gains which are used in a model-based filter to provide an estimate, $\hat{x}$, of the state vector, x, that minimizes $E\{[x-\hat{x}]^\mathsf{T}[x-\hat{x}]\}$.

Optimal State Feedback—Linear Quadratic Regulator

The Linear Quadratic Regulator (LQR) can be thought of as the LQG problem with zero disturbances and sensor noises ($w = 0$). This is a deterministic initial value problem with the cost,

$$J_r = \int_0^\infty (x(t)^\mathsf{T} R_{xx} x(t) + 2x(t)^\mathsf{T} R_{xu} u(t) + u(t)^\mathsf{T} R_{uu} u(t)) dt . \tag{3.10}$$

The solution to the LQR problem is the regulator gain, or the optimal state feedback gain matrix,

$$K_r = R_{uu}^{-1} [R_{xu}^\mathsf{T} + B_u^\mathsf{T} X] . \tag{3.11}$$

The positive semi-definite matrix X is to solution to the following Continuous Algebraic Riccati Equation (CARE),

$$A^\mathsf{T} X + XA - [XB_u + R_{xu}^\mathsf{T}] R_{uu}^{-1} [R_{xu} + B_u^\mathsf{T} X] + R_{xx} = 0 . \tag{3.12}$$

Optimal State Estimation—Kalman Filter

The Kalman Filter (KF) has the structure of a model-based observer,

$$\hat{x} = A\hat{x} + Bu + K_f (y - C_y \hat{x}) . \tag{3.13}$$

The model-based observer simulates the model states, and feeds back the error, $(y - C_y \hat{x})$, via the filter gain, or the optimal state estimation feedback matrix,

$$K_f = [YC_y^\mathsf{T} + V_{xy}] V_{yy}^{-1} . \tag{3.14}$$

The positive semi-definite matrix Y is to solution to the following CARE,

$$YA^\mathsf{T} + AY - [YC_y^\mathsf{T} + V_{xy}] V^{-1} [V_{xy}^\mathsf{T} + C_y Y] + W = 0 . \tag{3.15}$$

Combined Optimal State Feedback and State Estimation—LQG

Linear Quadratic Gaussian (LQG) combines the LQR and KF problems to produce a model-based controller of the form,

$$\begin{aligned} \dot{x_K} &= [A - B_u K_r - K_f C_y] x_K + K_f y \\ u_K &= -K_r x_K \end{aligned}, \tag{3.16}$$

$$K(j\omega) = -K_r[j\omega I - (A - B_u K_r - K_f C_y)]^{-1} K_f. \tag{3.17}$$

The synthesis of an LQG controller involves the selection of multiple design parameters to meet designer-specified performance and robustness requirements. These design parameters are the elements of the R and V matrices. Stabilizing controllers can be determined for an infinite combination of these design parameters (within the LQG constraints), leaving the control designer with many degrees of freedom.

Some simplification can generally be achieved by assigning some of the parameters according to the original control problem. For example, an RMS performance can be chosen as $R_{xx} = C_z^{\mathsf{T}} C_z$, and the disturbance inputs can determine $V_{xx} = B_w B_w^{\mathsf{T}}$ provided that the inputs and outputs of the control problem are scaled properly. A sensor noise floor can be represented by the choice of $V_{yy} = D_{yw} D_{yw}^{\mathsf{T}}$. If the designer is interested in minimizing the control energy used, the choice of $R_{uu} = D_{zu}^{\mathsf{T}} D_{zu}$ can be made.

The LQG design process is a powerful method for specifying broad performance and robustness goals for complex systems (i.e. many states, inputs, and outputs), using a small number of design parameters (knobs). However, there are two factors that complicate the design process. First, the LQG controller is a minimization of a particular cost function. There is no means to specify an absolute performance level. In practice, the control designer must choose the design weights, create the controller, analyze the closed loop, and then if necessary adjust the weights. Second, LQG provides no guarantees of robustness [Skogestad and Postlethwaite, 1996]. Once again, the designer must assess the closed loop system for robustness, and adjust the weights if needed. This is problematic

since the LQG design knobs are generally influential over a broad frequency band (absent particular frequency weighting cost functions such as notch and bandpass filters). This iterative nature, required for both nominal performance and for robustness, is particularly challenging for setpoint controller development, since the choices of the human designer should be consistent across the setpoint design space.

Graphical heuristic for LQG design

A heuristic that streamlines the LQG design process is now presented. This heuristic is derived from the observation that the closed loop LQR response can be approximated without having to solve for the filter gain. This allows the control designer to immediately predict the effects of modifying the control design parameters on LQR. An analogous argument is made for the Kalman Filter design. This is coupled to predict the final LQG closed-loop response.

While LQR minimizes a combination of state penalty and control cost, the control designer is frequently concerned only with the state penalty. This is the case when factors such as robustness and control bandwidth are more significant than control cost. As such, the LQR problem can be constructed so that G_{zu} is partitioned into the contributions from the state penalty, $G_{zu}^{(z)}$, and the contribution from control cost, $G_{zu}^{(u)}$. $G_{zu}^{(z)}$ is typically chosen as the true performance measure of the system, such as a pointing metric. $G_{zu}^{(u)}$ represents the free design parameters that are selected by the control designer. Frequency dependent performance can be implemented by augmenting $G_{zu}^{(u)}$ with additional dynamics.

If the magnitude of $G_{zu}^{(z)}$ is much larger than $G_{zu}^{(u)}$ at some point in frequency, any additional control effort used to reduce the magnitude of $G_{zu}^{(z)}$ will not add significantly to the total LQR cost at that point in frequency. Conversely, if the magnitude of $G_{zu}^{(z)}$ is much smaller than $G_{zu}^{(u)}$, any additional control effort used to reduce the magnitude of $G_{zu}^{(z)}$ will add directly to the total LQR cost. In other words, when the state penalty is much larger than the control cost, it will improve the LQR cost to attenuate the response until the state

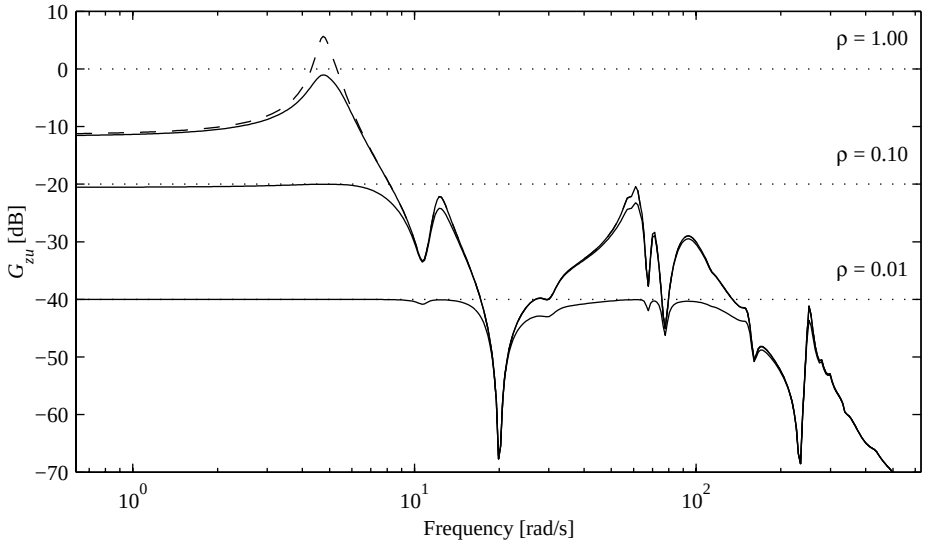

Figure 3.5 Variations in LQR system response with decreasing control penalty $\rho^2 = R_{uu}$. The three frequency responses show the closed loop response of the LQR problem overlaid onto the open loop response. The closed loop responses (solid lines) closely match the corresponding control weightings (dotted lines) wherever the open loop response (dashed line) exceeds the control weighting.

penalty matches the control cost; when the control cost is much larger than the state penalty, it will cost too much to try to attenuate the response.

This interpretation of LQR is presented graphically in Figure 3.5. There is a very strong correlation between $G_{zu}^{(u)}$, $G_{zu}^{(z)}$, and $G_{zu}^{(z)}\Big|_{CL}$. This correlation is close enough that a good approximation of $G_{zu}^{(z)}\Big|_{CL}$ is just the minimum of $G_{zu}^{(u)}$ and $G_{zu}^{(z)}$ at any particular frequency. A mathematical approximation can be used to describe this behavior,

$$\left|G_{zu}^{(z)}\Big|_{CL}\right| \approx \min(\left|G_{zu}^{(u)}\right|, \left|G_{zu}^{(z)}\right|). \tag{3.18}$$

A similar analysis can be applied to the design of the Kalman Filter. The Kalman Filter problem is completely described by the G_{yw} component of the general system of Equation 3.2. In a manner analogous to that of the LQR problem, this component can be partitioned into two parts; the response of the sensor to process noise, uncorrupted by sensor noise, $G_{yw}^{(w)}$; and the sensor noise itself, $G_{yw}^{(y)}$.

The estimation performance of the Kalman Filter can be represented by treating it as a regulator problem, with w being the control input and y being the performance output, and comparing $G_{yw}^{(y)}$ in open loop and in closed loop, $G_{yw}^{(y)}\Big|_{CL}$. Whereas in the LQR problem the difference between $G_{zu}^{(z)}$ and $G_{zu}^{(z)}\Big|_{CL}$ represents a reduction in the performance variable, in the KF problem the difference between $G_{yw}^{(y)}$ and $G_{yw}^{(y)}\Big|_{CL}$ represents the ability of the KF to reject sensor noise and sense the excitation of the plant states due to disturbances.

The $G_{yw}^{(y)}$, $G_{yw}^{(w)}$, and $G_{yw}^{(y)}\Big|_{CL}$ responses for three sample KF designs are shown in Figure 3.6. Each KF design has been designed with a different sensor noise level. Where $G_{yw}^{(y)}\Big|_{CL}$ approximately overlays $G_{yw}^{(y)}$, the filter is unable to estimate any excitation of the states from process noise; sensor noise dominates and there is no useful information detectable in the sensors. Where $G_{yw}^{(y)}\Big|_{CL}$ is smaller than $G_{yw}^{(y)}$, the process noise sufficiently excites the system so that disturbance information is present in the sensor signal. As with the LQR problem, the combination of $G_{yw}^{(y)}$ and $G_{yw}^{(w)}$ in the open loop is sufficient to provide a good approximation of $G_{yw}^{(y)}$ in the closed loop. Again this behavior can be summarized by an approximation,

$$\left|G_{yw}^{(y)}\Big|_{CL}\right| \approx \min(\left|G_{yw}^{(w)}\right|, \left|G_{yw}^{(y)}\right|). \tag{3.19}$$

This understanding of how the control design parameters affect LQR and Kalman Filter design can be used to interpret how the same parameters will affect coupled LQG controller design. Although LQG design is a non-trivial coupling of LQR and the Kalman Filter, the following three examples illustrate how similar reasoning can be used to interpret the LQG closed loop response. The first example is an LQG design where the filter dynamics are effectively eliminated by decreasing V_{yy}. As V_{yy} is decreased to zero, the filter gains become asymptotically large as seen in Equation 3.14. The compensator frequency response of Equation 3.16 approaches,

$$K|_{K_f \to \infty} = -K_r[K_f C_y]^{-1} K_f. \tag{3.20}$$

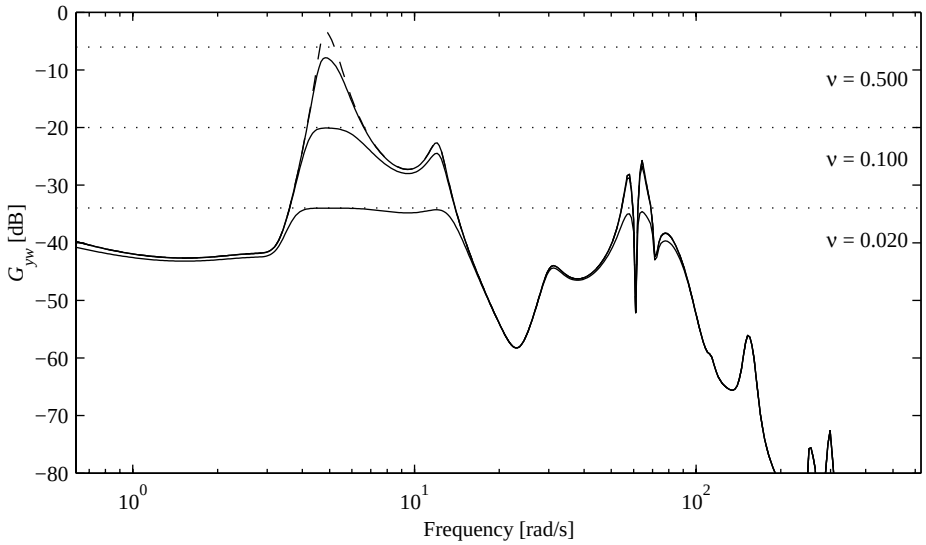

Figure 3.6 Variations in Kalman Filter performance with decreasing sensor noise $\nu = V_{yy}$. The three frequency responses show the closed loop performance of the Kalman Filter overlaid onto the open loop frequency response. The closed loop response (solid lines) closely match the control weighting (dotted lines) wherever the open loop response (dashed line) exceeds the sensor noise.

Applying the feedback law $u = -K|_{K_f \to \infty} y$ to the plant of Equation 3.1, the closed loop system dynamics equation becomes,

$$\dot{x} = (A - B_u K_r)x + B_w w . \tag{3.21}$$

The closed loop response is no longer a function of the filter gain K_f, and the dynamics are modified only by the regulator gain K_r. This allows the coupling between the regulator gain K_r and the LQG closed loop response to be examined. As was the case for the LQR problem, control can only be applied where the physical performance exceeds the control weighting. However, in general, the disturbance and control input may enter the system differently. The assessment of closed loop performance is done separately with G_{zw} in open and closed loop. This is shown in Figure 3.7. Performance is achieved where the control cost is smaller than the state penalty; the difference between the state penalty and the control cost is approximately the amount that the controller can attenuate the disturbance to performance frequency response. In regions where control effort is available, there is a strong correlation between dips in the G_{zu} frequency response and spikes in the

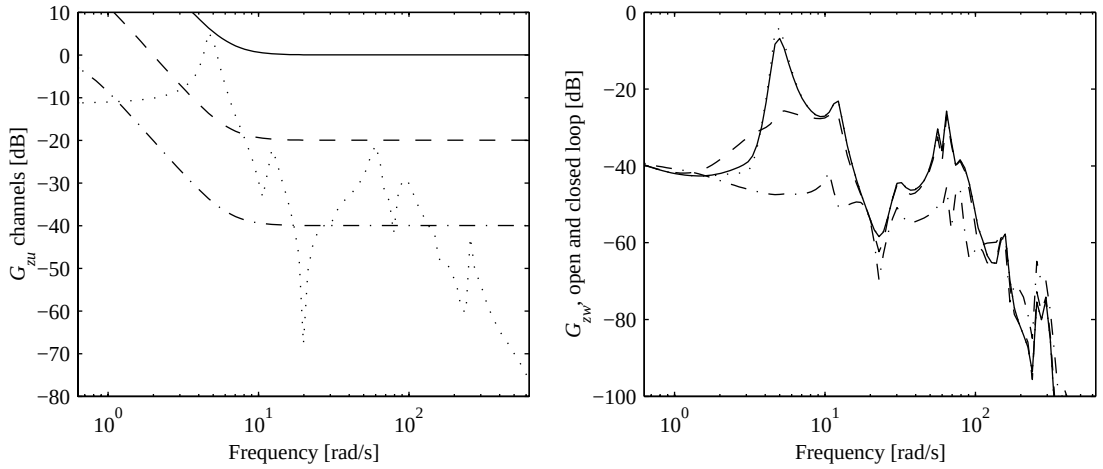

Figure 3.7 LQG design for high filter gains. The left plot shows the open loop G_{zu} channels of control cost and state penalty. The dotted line shows the state penalty, and there are three different control cost weightings corresponding to different LQG designs. The highest control cost is a solid line, corresponding to the lowest regulator gain. The next highest control cost is a dashed line, and the lowest control cost is a dot-dash line. The right plot shows the closed loop performance. The solid, dashed and dot-dash lines are different control designs corresponding to the control costs on the left plot. The open loop response is also shown as a dotted line.

G_{zw} frequency response. Whereas for the LQR problem an approximation was made for the closed loop response, an approximation can now be made to the *improvement* in performance:

$$\frac{\left|G_{zw}\right|_{CL}}{\left|G_{zw}\right|} \approx \frac{\min(\left|G_{zu}^{(u)}\right|, \left|G_{zu}^{(z)}\right|)}{\left|G_{zu}^{(z)}\right|}, \text{ for } \left\|V_{yy}\right\| \ll \left\|V_{ww}\right\|. \tag{3.22}$$

In the second example, the regulator gain is set arbitrarily high by decreasing R_{uu} to zero. The regulator gains become asymptotically large (Equation 3.11), and the compensator frequency response of Equation 3.16 approaches,

$$K\big|_{K_r \to \infty} = -K_r[B_u K_r]^{-1} K_f. \tag{3.23}$$

Applying the feedback law $u = -K\big|_{K_r \to \infty} y$ to the plant of Equation 3.1, the closed loop system dynamics becomes,

$$\dot{x} = (A - K_f C_y)x + B_w w. \tag{3.24}$$

The closed loop response is no longer a function of the regulator gain K_r, and the dynamics are modified only by the filter gain K_f. The effect of varying the sensor noise level on the LQG design is illustrated in Figure 3.8. Control effort is possible only where the estimator is able to detect disturbances above the sensor noise level. Note that in this example, G_{yw} and G_{zw} are the same since the performance and sensor variables are collocated. As with the previous example where the filter gain was large, an approximation can be made for the performance improvement,

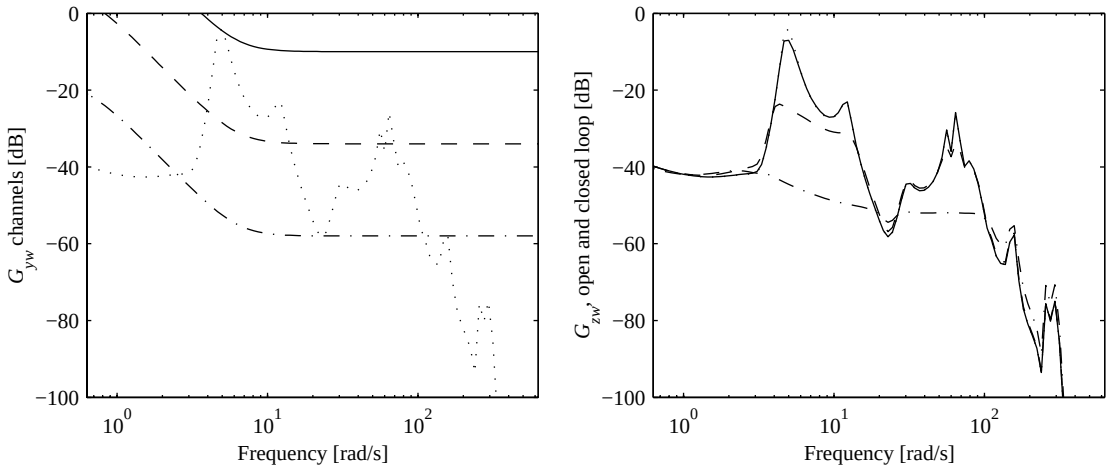

Figure 3.8 LQG design for high regulator gains. The left plot shows the open loop G_{yw} channels of sensor noise and process noise. The dotted line shows the process noise, and there are three different sensor noise weightings corresponding to different LQG designs. The highest sensor noise is a solid line, corresponding to the lowest filter gain. The next highest sensor noise is a dashed line, and the lowest sensor noise is a dot-dash line. The right plot shows the closed loop performance. The solid, dashed and dot-dash lines are different control designs corresponding to the control costs on the left plot. The open loop response is also shown as a dotted line.

$$\frac{\left|G_{zw}\right|_{CL}\left|}{\left|G_{zw}\right|} \approx \frac{\min(\left|G_{yw}^{(w)}\right|, \left|G_{yw}^{(y)}\right|)}{\left|G_{yw}^{(y)}\right|}, \text{ for } \left\|R_{uu}\right\| \ll \left\|R_{zz}\right\|. \tag{3.25}$$

To summarize the previous two examples, two factors determine how the design parameters will influence the closed loop response of LQG control. First, control is possible only in a particular frequency band only if the state penalty is greater than the control cost, as represented by Equation 3.22. Second, control is possible in a particular frequency band

only if the process noise is capable of exciting the system enough to be detected over the sensor noise. This is represented in Equation 3.25.

The high filter and regulator gains used in these examples tend to introduce undesirable high frequency dynamics into the controller, as well as potentially non-robust pole-zero cancellations. However, the insight gained from these limiting cases is still applicable when both the filter and regulator gains are important. Figure 3.9 shows a design example where both the filter and regulator gains are chosen so that they both influence the closed loop performance. Performance is improved only when both 1) the state penalty exceeds

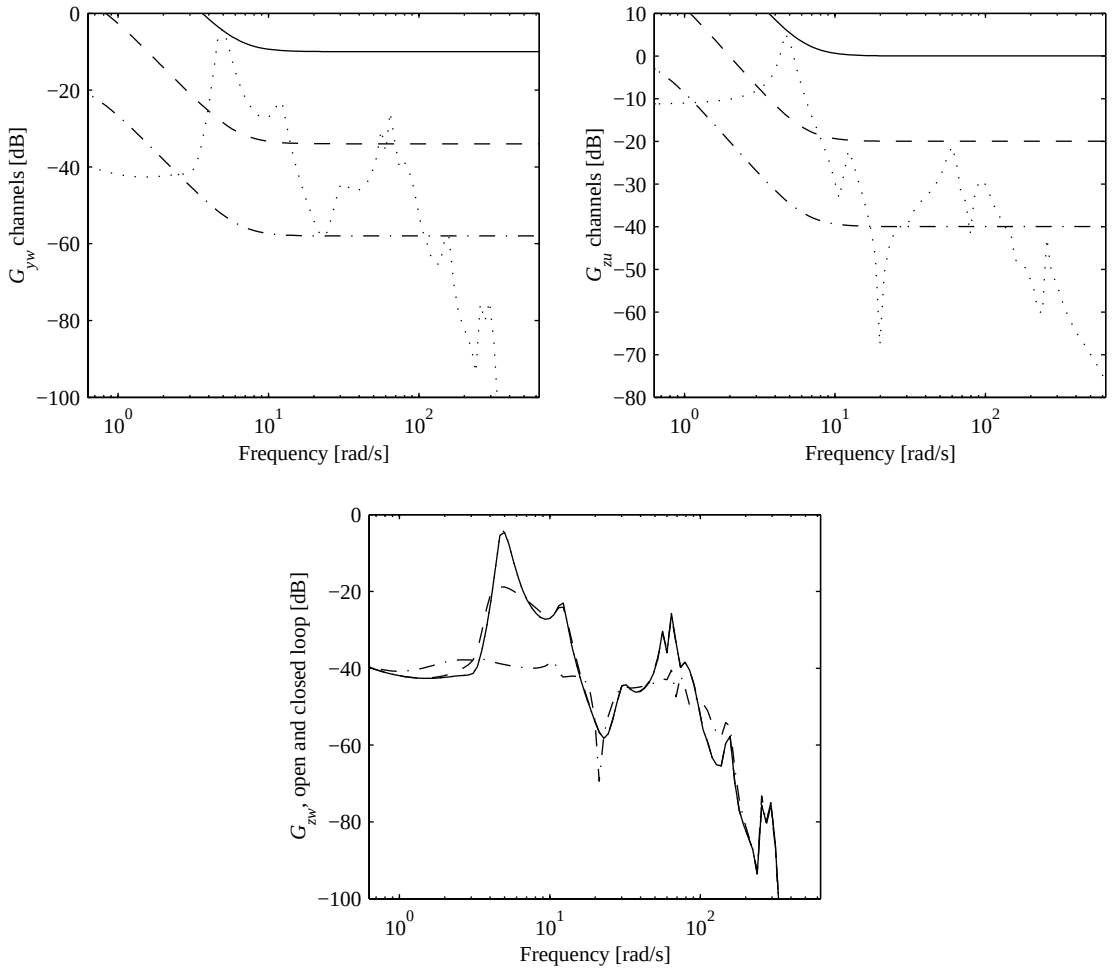

Figure 3.9 LQG problem for moderate regulator and filter gains. The resulting closed loop response is only improved if 1) the state penalty exceeds the control cost, and 2) the process noise is more significant than the sensor noise.

the control cost, and 2) the process noise is more significant than the sensor noise. The following approximation summarizes this statement of performance improvement,

$$\frac{\left|G_{zw}\right|_{CL}}{\left|G_{zw}\right|} \approx \max\left(\frac{\min(\left|G_{zu}^{(u)}\right|, \left|G_{zu}^{(z)}\right|)}{\left|G_{zu}^{(z)}\right|}, \frac{\min(\left|G_{yw}^{(w)}\right|, \left|G_{yw}^{(y)}\right|)}{\left|G_{yw}^{(y)}\right|}\right). \tag{3.26}$$

The insights gained from this discussion streamline the LQG design process in two different ways. First, it allows the control designer to select an initial set of design parameters and be able to predict, to a first order, the closed loop response without performing the complete controller synthesis and analysis. In other words, the control designer can approximately specify the desired performance using this heuristic. Without this heuristic, significant iterations of analysis and re-design would be required.

Second, these insights assist in the tuning of an initial LQG design. The effects of perturbing the design parameters can again be predicted to a first order, and this can give the control designer a "direction" to search in the parameter space.

A SISO control topology has been considered in the examples. Similar heuristics do however apply to MIMO control. The control designer must in this case carefully select the appropriate input and output channels to compare; for example, when choosing the filter gains, it is most appropriate to consider the sensor that is most strongly coupled to the disturbance. Alternatively, the maximum singular values of the different frequency responses can also be considered.

Another feature of this graphical heuristic is that it can accommodate a variety of frequency-dependent weightings. Such weightings are necessary in practical control design; they may be used to de-emphasize regions of control where more sensor noise is anticipated, or to focus control on a narrow band if the disturbance is tonal. As long as the control designer selects the appropriate channels to compare, the insights from the graphical heuristics remain applicable.

One aspect that is not addressed by this graphical heuristic is the phenomenon of non-minimum phase zeros. In general performance can only be achieved either above or below the frequency of NMP zeros.

The heuristics developed for LQG design in this section illustrate the fact that LQG design parameters, or "knobs", tend to have broadband influences on the controller design. However, the lightly damped modally dense characteristics of the structural control problem requires "knobs" that are more narrowband. This fundamentally limits the ability of LQG to handle modal uncertainty, and as a result performance must be traded off for robustness.

3.2.2 Sensitivity Weighted Linear Quadratic Gaussian

SWLQG attempts to address the robustness issues of LQG by desensitizing the controller to uncertainties in the model [Sesak, 1974]. This is achieved by augmenting the LQG cost with a penalty to sensitivity to changes in an uncertain parameter, α. The sensitivities are incorporated as follows,

$$J = E\left\{ \lim_{T\to\infty} \frac{1}{T} \int_0^T \left[x^\mathsf{T} R_{xx} x + \frac{\partial x}{\partial \alpha}^\mathsf{T} R_{\alpha\alpha} \frac{\partial x}{\partial \alpha} + 2x^\mathsf{T} R_{xu} u + u^\mathsf{T} R_{uu} u \right] dt \right\}, \tag{3.27}$$

These uncertainties can be expressed as the derivatives of the state-space matrices with respect to the uncertain parameters. The effect of the sensitivity weighting is to add additional terms to the LQG weighting matrices,

$$\begin{aligned}
R_{xx}^{\mathrm{SW}} &= R_{xx} + \sum_{i=1}^{n_\alpha} \frac{\partial A}{\partial \alpha_i}^\mathsf{T} A^{-\mathsf{T}} R_{\alpha\alpha_i} A^{-1} \frac{\partial A}{\partial \alpha_i} \quad V_{xx}^{\mathrm{SW}} = V_{xx} + \sum_{i=1}^{n_\alpha} \frac{\partial A}{\partial \alpha_i} A^{-1} V_{\alpha\alpha_i} A^{-\mathsf{T}} \frac{\partial A}{\partial \alpha_i}^\mathsf{T} \\
R_{xu}^{\mathrm{SW}} &= R_{xu} + \sum_{i=1}^{n_\alpha} \frac{\partial A}{\partial \alpha_i}^\mathsf{T} A^{-\mathsf{T}} R_{\alpha\alpha_i} A^{-1} \frac{\partial B}{\partial \alpha_i}, \; V_{xy}^{\mathrm{SW}} = V_{xy} + \sum_{i=1}^{n_\alpha} \frac{\partial A}{\partial \alpha_i} A^{-1} V_{\alpha\alpha_i} A^{-\mathsf{T}} \frac{\partial C}{\partial \alpha_i}^\mathsf{T} . \\
R_{uu}^{\mathrm{SW}} &= R_{xx} + \sum_{i=1}^{n_\alpha} \frac{\partial B}{\partial \alpha_i}^\mathsf{T} A^{-\mathsf{T}} R_{\alpha\alpha_i} A^{-1} \frac{\partial B}{\partial \alpha_i} \quad V_{yy}^{\mathrm{SW}} = V_{yy} + \sum_{i=1}^{n_\alpha} \frac{\partial C}{\partial \alpha_i} A^{-1} V_{\alpha\alpha_i} A^{-\mathsf{T}} \frac{\partial C}{\partial \alpha_i}^\mathsf{T}
\end{aligned} \tag{3.28}$$

The SWLQG controller is then computed as the LQG solution (via the CAREs of Equations 3.12 and 3.15), but with this different set of weighting matrices. A useful simplification can be made in the selection of the sensitivity weights in the case of frequency uncertainty. This requires the assumption that the frequency uncertainties do not affect the B and C matrices to the first order,

$$\frac{\partial B}{\partial \alpha_i} = 0 , \frac{\partial C}{\partial \alpha_i} = 0 . \tag{3.29}$$

With this assumption, it is apparent from Equation 3.28 that only R_{xx} and V_{xx} are modified by the sensitivity weights.

Another simplification occurs if the system is transformed to the real modal realization, so that each mode is represented by two states, or a 2x2 block in the A matrix. Each frequency uncertainty is now localized to a 2x2 block, and the modifications to the penalty matrices R_{xx} and V_{xx} simplify to,

$$R_{xx}^{\mathrm{SW}} = R_{xx} + \sum_{i=1}^{n_\alpha} \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & & \\ & R^i_{\alpha\alpha} & \\ & & \ddots \\ 0 & \cdots & 0 \end{bmatrix} , V_{xx}^{\mathrm{SW}} = V_{xx} + \sum_{i=1}^{n_\alpha} \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & & \\ & V^i_{\alpha\alpha} & \\ & & \ddots \\ 0 & \cdots & 0 \end{bmatrix} \tag{3.30}$$

The sensitivity weights become additions to the penalty matrices R_{xx} and V_{xx} at the corresponding locations in the A matrix. Increasing the magnitude of 2x2 blocks $R^i_{\alpha\alpha}$ and $V^i_{\alpha\alpha}$ increases the penalty on uncertainty, but the structure of each block is a choice left to the control designer.

SWLQG is an improvement over LQG for accommodating parameter error. Since the sensitizing weights are added to the LQG cost, the SWLQG controller is suboptimal (with respect to the LQG cost). However, the robustifying effects are targeted to particular modes, and as a consequence are narrowband. In addition, the method is advantageous for

gain scheduled control due to the closed-form nature of the solution, which enables rapid solution for all setpoint designs. However, SWLQG does not guarantee any level of robustness. In practice, the designer must choose a set of sensitizing weights, synthesize the controller, analyze the achieved robustness, and if necessary change the sensitivity weights. Further, since the SWLQG controller is suboptimal, the nominal performance may not be met, necessitating a further iteration on the LQG control and sensor noise weights.

Graphical heuristic for SWLQG design

A heuristic that streamlines the SWLQG design process is now presented. The SWLQG heuristic is similar to the LQG heuristic, in that the magnitudes of different frequency responses of the weighted open-loop system are compared in the frequency domain. The first step is to realize the sensitivity weights as sensitivity inputs and outputs that are appended to the control design problem,

$$C_q^{\mathsf{T}} C_q = \sum_{i=1}^{n_{\mathrm{SW}}} \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \\ & R^i_{\alpha\alpha} & \\ & & \ddots \\ 0 & \cdots & 0 \end{bmatrix}, \; B_p B_p^{\mathsf{T}} = \sum_{i=1}^{n_{\mathrm{SW}}} \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \\ & V^i_{\alpha\alpha} & \\ & & \ddots \\ 0 & \cdots & 0 \end{bmatrix}. \tag{3.31}$$

This produces the sensitivity augmented system,

$$\begin{bmatrix} \dot{x} \\ q \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_p & B_w & B_u \\ C_q & D_{qp} & D_{qw} & D_{qu} \\ C_z & D_{zp} & D_{zw} & D_{zu} \\ C_y & D_{yp} & D_{yw} & D_{yu} \end{bmatrix} \begin{bmatrix} x \\ p \\ w \\ u \end{bmatrix}. \tag{3.32}$$

The sensitivity feedthrough terms (D_{qp}, D_{qw}, D_{qu}, D_{zp}, and D_{yp}) are all zero matrices of commensurate dimension. The sensitivity weights are chosen as,

$$R^i_{\alpha\alpha} = (\sqrt{\beta_i}\begin{bmatrix}1 & 1\end{bmatrix})^{\mathsf{T}}(\sqrt{\beta_i}\begin{bmatrix}1 & 1\end{bmatrix}),\ V^i_{\alpha\alpha} = \left(\sqrt{\gamma_i}\begin{bmatrix}1\\1\end{bmatrix}\right)\left(\sqrt{\gamma_i}\begin{bmatrix}1\\1\end{bmatrix}\right)^{\mathsf{T}}, \tag{3.33}$$

so that the variables β_i and γ_i now become the design parameters for SWLQG. The choice of this structure for $R^i_{\alpha\alpha}$ and $V^i_{\alpha\alpha}$ allows each sensitivity weight to be appended to the control problem as a single sensitivity input-output pair (an arbitrary 2x2 weight structure could be implemented using multiple sensitivity input-output pairs).

For multiple uncertainties, the sensitivity input and output matrices are as follows,

$$B_p = \begin{bmatrix} 0 & \cdots & 0 \\ \sqrt{\gamma_1}\begin{bmatrix}1\\1\end{bmatrix} & \ddots & \vdots \\ 0 & \cdots & 0 \\ \vdots & \ddots & \sqrt{\gamma_{n_\alpha}}\begin{bmatrix}1\\1\end{bmatrix} \\ 0 & \cdots & 0 \end{bmatrix},\ C_q = \begin{bmatrix} 0 & \sqrt{\beta_1}\begin{bmatrix}1 & 1\end{bmatrix} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \sqrt{\beta_{n_\alpha}}\begin{bmatrix}1 & 1\end{bmatrix} & 0 \end{bmatrix}, \tag{3.34}$$

Once the sensitivity inputs and outputs have been realized, the second step is to determine which frequency responses (magnitudes) to compare. Useful insight can be gained by examining how the sensitivity inputs and outputs are related to the SWLQG weighting matrices used in the CAREs of Equations 3.12 and 3.15. These SWLQG weighting matrices can be conveniently recovered as,

$$\begin{aligned} R^{\mathrm{SW}} &= \begin{bmatrix}C_z^{\mathsf{T}} \\ D_{zu}^{\mathsf{T}}\end{bmatrix}\begin{bmatrix}C_z & D_{zu}\end{bmatrix} + \begin{bmatrix}C_q^{\mathsf{T}} \\ D_{qu}^{\mathsf{T}}\end{bmatrix}\begin{bmatrix}C_q & D_{qu}\end{bmatrix} \\ V^{\mathrm{SW}} &= \begin{bmatrix}B_w \\ D_{yw}\end{bmatrix}\begin{bmatrix}B_w^{\mathsf{T}} & D_{yw}^{\mathsf{T}}\end{bmatrix} + \begin{bmatrix}B_p \\ D_{yp}\end{bmatrix}\begin{bmatrix}B_p^{\mathsf{T}} & D_{yp}^{\mathsf{T}}\end{bmatrix} \end{aligned} . \tag{3.35}$$

The structure of the SWLQG matrices suggests that the appropriate frequency responses to compare are G_{zu} against G_{qu}, and G_{yw} against G_{yp}. Comparing G_{zu} against G_{qu} will

determine how the sensitivity weights affect the LQR regulator problem, and comparing G_{yw} against G_{yp} will determine how the sensitivity weights affect the KF estimation problem.

Because of the possibility of a MIMO control topology, as well as multiple sensitivity channels, the Singular Values of each frequency response are compared. Examples of SWLQG design are shown in Figure 3.10 with three plots. The first plot (a) shows the analysis of the regulation problem, comparing the maximum singular values of G_{zu} against G_{qu}. The second plot (b) shows the analysis of the estimation problem, comparing the maximum singular values of G_{yw} against G_{yp}. Finally the resulting SWLQG control designs are overlaid with the LQG control design in (c).

Where G_{qu} is much smaller than G_{zu}, and G_{yp} is much smaller than G_{yw}, the SWLQG controller is very similar to the LQG controller. In this case, the penalty on uncertainty does not contribute to the total cost. When the G_{qu} is on the order of G_{zu}, and G_{yp} is on the order of G_{yw}, the uncertainty penalty is now significant, and thus the SWLQG controller is noticeably different from the LQG controller.

The controllers of Figure 3.10 also illustrate another aspect of SWLQG. The average gain of the SWLQG controllers decreases with increasing sensitivity weights for the same set of LQG gains. This is the trade-off between robustness and H_2 performance. In practical control design, this is typically countered by simultaneously decreasing the control penalty and sensor noise.

By choosing the appropriate state-space realization, and the appropriate uncertainty input-output structure, the effect of sensitivity weighting can now be viewed graphically. The SWLQG graphical heuristic does not provide an approximation to closed loop behavior as does the LQG heuristic. Rather, it answers the question of whether a sensitivity weight is strong enough to modify the underlying LQG design. This can save on many control design iterations by providing an initial sensitivity weighting.

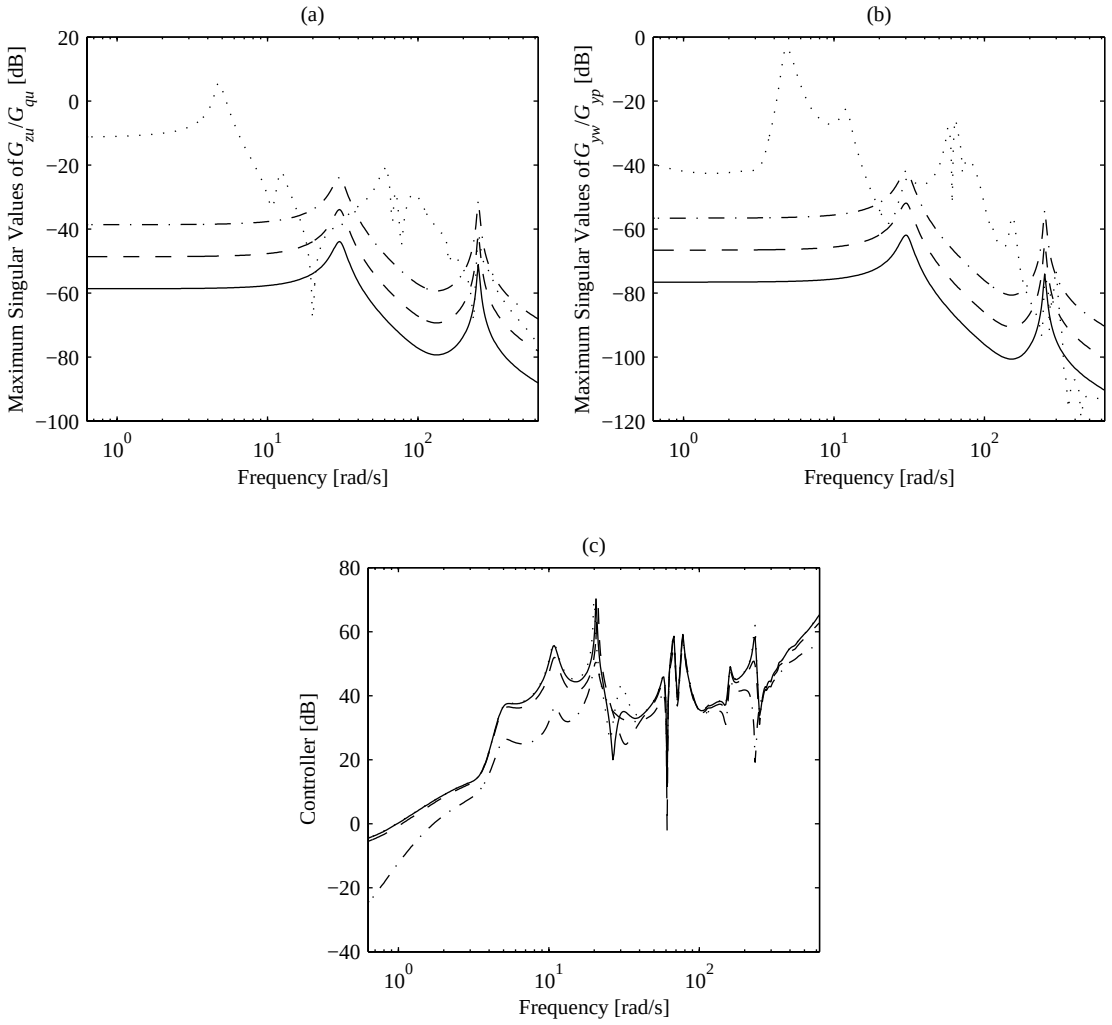

Figure 3.10 Sample SWLQG controller designed with varying sensitivity weights. Different sensitivity weight, (a) and (b), are used to design SWLQG controllers (c). Each design is shown with a different line style: solid for the lowest sensitivity weight; dashed and dot-dash for increasing weights. In (a), the maximum singular values of the sensitivity weight G_{qu} are compared to G_{zu} (dotted line), and in (b) the maximum singular values of the sensitivity weight G_{yp} are compared to G_{yw} (dotted line). Finally, the reference LQG compensator is also shown in (c) for comparison (dotted line).

3.2.3 Multiple Model

Whereas SWLQG desensitizes the controller to possibly destabilizing parameter uncertainty, Multiple Model (MM) control design can provide stability *guarantees* to uncertainty. This comes at the cost of much greater computational effort in the synthesis stage, as the solution involves a cost-minimizing gradient-search. This subsection reviews MM

control and provides the foundation for optimizing gain-scheduled controllers, discussed in the next chapter.

The objective of MM design is to find a controller that minimizes the H_2 cost for a discrete set of plants, $[A^i, B^i, C^i, D^i]$, $i = 1,...,n_\theta$. Although stability is only strictly guaranteed for each plant in the discrete set, if the variations between within the set of plants are small it is likely that other plants within the bounds of the set are also stable. The solution is a minimization of the following cost functional,

$$J = \sum_{i=1}^{n_\theta} \mathrm{tr}\left\{ Q^i C_z^i\Big|_{\mathrm{CL}}{}^{\mathsf{T}} C_z^i\Big|_{\mathrm{CL}} \right\}, \tag{3.36}$$

over the matrices $[A_K, B_K, C_K]$ of a state-space controller. There is no known closed form solution to this control problem, so recourse must be made to nonlinear optimization techniques. The gradients of the cost functional with respect to the controller matrices should be derived to facilitate the solution process. These gradients have been derived in the literature [Grocott, 1994]; the solution is summarized here. For each closed-loop system, two Lyapunov equations are solved,

$$\begin{aligned} A^i\Big|_{\mathrm{CL}} Q^i + Q^i A^i\Big|_{\mathrm{CL}}{}^{\mathsf{T}} + B_w^i\Big|_{\mathrm{CL}} B_w^i\Big|_{\mathrm{CL}}{}^{\mathsf{T}} &= 0 \\ P^i A^i\Big|_{\mathrm{CL}} + A^i\Big|_{\mathrm{CL}}{}^{\mathsf{T}} P^i + C_z^i\Big|_{\mathrm{CL}}{}^{\mathsf{T}} C_z^i\Big|_{\mathrm{CL}} &= 0 \end{aligned}, \tag{3.37}$$

The Lyapunov equation solutions P and Q are partitioned as,

$$P^i = \begin{bmatrix} P_{11}^i & P_{12}^i \\ P_{21}^i & P_{22}^i \end{bmatrix}, \; Q^i = \begin{bmatrix} Q_{11}^i & Q_{12}^i \\ Q_{21}^i & Q_{22}^i \end{bmatrix}, \tag{3.38}$$

$$(PQ)^i = \begin{bmatrix} (PQ)^i{}_{11} & (PQ)_{12}^i \\ (PQ)_{21}^i & (PQ)_{22}^i \end{bmatrix}, \; (QP)^i = \begin{bmatrix} (QP)_{11}^i & (QP)_{12}^i \\ (QP)_{21}^i & (QP)_{22}^i \end{bmatrix}, \tag{3.39}$$

where the $(\cdot)_{11}$ partition has the dimensions of the matrix A^i, and the $(\cdot)_{22}$ partition has the dimensions of the matrix A_K. Then the derivatives are given by,

$$\frac{\partial J}{\partial A_K} = \sum_{i=1}^{n_\theta} (QP)_{22}^{i\,\mathsf{T}} + (PQ)_{22}^{i}, \tag{3.40}$$

$$\frac{\partial J}{\partial B_K} = \sum_{i=1}^{n_\theta} ((QP)_{12}^{i\,\mathsf{T}} + (PQ)_{21}^{i}) C_y^{i\,\mathsf{T}} + (P_{12}^{i\,\mathsf{T}} + P_{21}^{i}) B_w^i D_{yw}^{i\,\mathsf{T}} + (P_{22}^{i\,\mathsf{T}} + P_{22}^{i}) B_c D_{yw}^i D_{yw}^{i\,\mathsf{T}}, \tag{3.41}$$

$$\frac{\partial J}{\partial C_K} = \sum_{i=1}^{n_\theta} D_{zu}^{i\,\mathsf{T}} D_{zu}^i C_c (Q_{22}^{i\,\mathsf{T}} Q_{22}^i) + D_{zu}^{i\,\mathsf{T}} C_z^i (Q_{21}^{i\,\mathsf{T}} + Q_{12}^i) + B_u^{i\,\mathsf{T}} ((QP)_{21}^{i\,\mathsf{T}} + (PQ)_{12}^i). \tag{3.42}$$

Since the MM design process is an optimization, an initial guess is required. This is typically an SWLQG design. Another requirement is that the initial guess be stabilizing over the entire set of plants. If this is not the case, two approaches can be taken. The overall gain of the initial controller can be decreased by modifying the SWLQG design until the controller is stabilizing over the entire set of plants. The optimization can then proceed with the original design weights. Another approach is to design successive MM controllers with progressively increasing variations. An initial set of plants is selected with a variation that is sufficiently small so that the initial controller is stabilizing. The MM controller for this initial set is used as the starting point for the next MM design with a set of plants that has larger variations. This chain of MM designs is continued until the desired variations are achieved with a stabilizing controller.

The MM control design process has been shown to provide robust high performance linear controllers [Grocott, 1994], but the disadvantage lies in the computational effort required. A further application of MM control design that deserves special attention is in the design

of reduced order controllers. Although (SW)LQG controllers are the same size as the plant, the gradient equations (3.36 through 3.42) apply equally to controllers of other sizes. The initial guess can thus be a reduced order (SW)LQG controller.

3.3 Analysis Tools

Once a linear controller is designed several analyses must be performed to ensure that the closed loop behavior meets the design requirements. This section covers four important analyses tools that will be used for linear setpoint control design: performance, complementary sensitivity/bandwidth, Multivariable Nyquist Criterion on a Nichols chart, and finally the singular values of the sensitivity transfer function.

3.3.1 Performance Analysis

Nominal controller performance is generally assessed by examining the performance frequency response of the closed loop system. For multiple disturbance inputs or multiple performance outputs the cost as a function of frequency is given by,

$$J(\omega) = \mathrm{tr}\left\{ G_{zw}(j\omega)\big|_{\mathrm{CL}} G_{zw}(j\omega)\big|_{\mathrm{CL}}^{\mathsf{H}} \right\}, \tag{3.43}$$

where the closed loop performance frequency response, $G_{zw}(j\omega)\big|_{\mathrm{CL}}$, is given by Equation 3.4. This allows the control designer to examine the frequency dependence of performance. A sample plot of performance is provided in Figure 3.11. To compare different control designs, this cost is typically integrated between a low and high frequency specification. Performance is measured in decibels relative to the open loop performance,

$$J = 20\log \sqrt{\frac{\int_{\omega_1}^{\omega_2} \mathrm{tr}\{(G_{zw}(j\omega))G_{zw}(j\omega)^{\mathsf{H}}\}d\omega}{\int_{\omega_1}^{\omega_2} \mathrm{tr}\left\{ G_{zw}(j\omega)\big|_{\mathrm{CL}} G_{zw}(j\omega)\big|_{\mathrm{CL}}^{\mathsf{H}} \right\} d\omega}}. \tag{3.44}$$

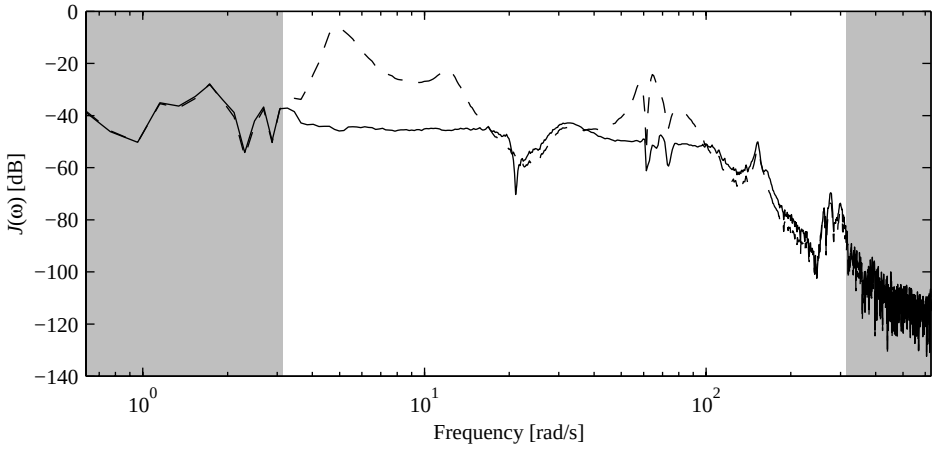

Figure 3.11 Typical performance plot. The closed loop response is shown as a solid line; this is overlaid onto the open loop response (dashed line) for comparison. The performance is typically integrated between a low and high frequency. The frequencies that do not contribute to the performance integral are indicated by the shaded areas.

3.3.2 Complementary Sensitivity and Bandwidth

The complementary sensitivity transfer function, defined as,

$$C(j\omega) = (I + G_{yu}(j\omega)K(j\omega))^{-1} G_{yu}(j\omega)K(j\omega), \tag{3.45}$$

provides a measure of the frequency range over which a system tracks signals. Examining the complementary sensitivity can reveal where the controller is affecting the closed loop system. This is particularly important when it comes to verifying that the controller has rolled off at higher frequencies, since the validity of control design models is generally limited in frequency. For example, digital control computers are limited by the sampling rate; MIT/MACE-II uses a sampling rate of 500 Hz and the Nyquist Rate fundamentally limits control to 250 Hz.

Complementary sensitivity and the bandwidth are closely related since bandwidth is defined by the highest frequency where the loop frequency response is close to unity (in SISO terms, the magnitude of the loop frequency response $G_{yu}(j\omega)K(j\omega)$ must be close to or greater than unity for control to influence the closed loop system).

3.3.3 Multivariable Nyquist Stability Criterion on the Nichols Chart

Nominal stability can be determined using the Multivariable Nyquist Stability criterion (hereafter referred to as the Nyquist Criterion). The Nyquist Criterion considers the net number of encirclements in the complex plane of the critical point $(-1+j0)$ of the Nyquist function [Skogestad and Postlethwaite, 1996],

$$H_n(j\omega) = -1 + \left| I + G_{yu}(j\omega)K(j\omega) \right| . \tag{3.46}$$

For stability, the net number of counterclockwise encirclements of the critical point must be equal to the number of open loop unstable poles. Then for a plant and controller that are both stable in the open loop, any encirclement of the critical point implies that the closed loop system will be unstable.

For a lightly damped system, complex poles appears as large circles in the complex plane. For a system where $\left|H_n(j\omega)\right|$ is much larger than one, the evaluation of encirclements can be difficult as the critical point is relatively close to the origin. A more accurate assessment can be made using the Nichols Chart, where the Nyquist criterion is shown on a phase/log-magnitude axis. This is shown in Figure 3.12. The phase/log-magnitude scale has the effect of unwrapping the Nyquist criterion, so that the critical point appears as multiple points on the phase scale. The net number of counterclockwise encirclements of the critical point in the complex plane is equal to the net number of left-to-right passes over any critical point in the phase/log-magnitude axis.

In a SISO system, robustness can be evaluated in terms of gain and phase margins. In the Nichols chart, these conveniently correspond to the vertical and horizontal distances from the critical points. In MIMO systems this can also be used as a guideline [Mallory, 2000]; for example if all features of the Nyquist criterion are far away from the critical point then the system can be assumed robustly stable.

However, this guideline can be misleading when the plant or controller frequency responses are poorly scaled. The determinant operator in the Nyquist criterion can mask

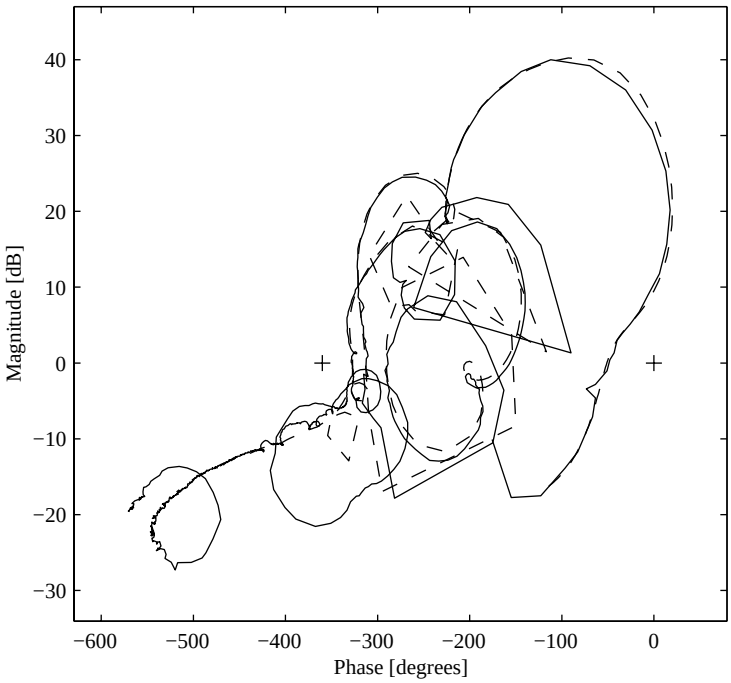

Figure 3.12 Nichols chart for sample problem. The controller closed on the experimental data is shown in as a solid line, and the controller closed on the design model is shown as a dashed line. The Nyquist critical points are shown as "+" symbols. Since there are no passes over any critical points, and both controller and plant are open loop stable, the Multivariable Nyquist criterion predicts stability in the closed loop for the both the experimental data and the design model.

sensitivities of the closed loop to small perturbations. This motivates the use of the singular values of the sensitivity transfer function.

3.3.4 Singular Values of the Sensitivity Transfer Function

One the Nyquist criterion is used to determine absolute stability, the relative stability or stability robustness can be determined by examining the singular values of the sensitivity transfer function.

The singular values of the sensitivity transfer function are determined as:

$$S(\omega) = \sigma((I + G_{yu}(j\omega)K(j\omega))^{-1}). \tag{3.47}$$

These singular values are a measure of the sensitivity of the closed loop response to perturbations. Although a large singular value does not necessarily imply instability, large differences between the singular values computed for experimental frequency response and the singular values computed for the design model indicate that the closed loop system will be very sensitive to modeling errors. Figure 3.13 shows the sensitivity maximum singular values for two different controller designs.

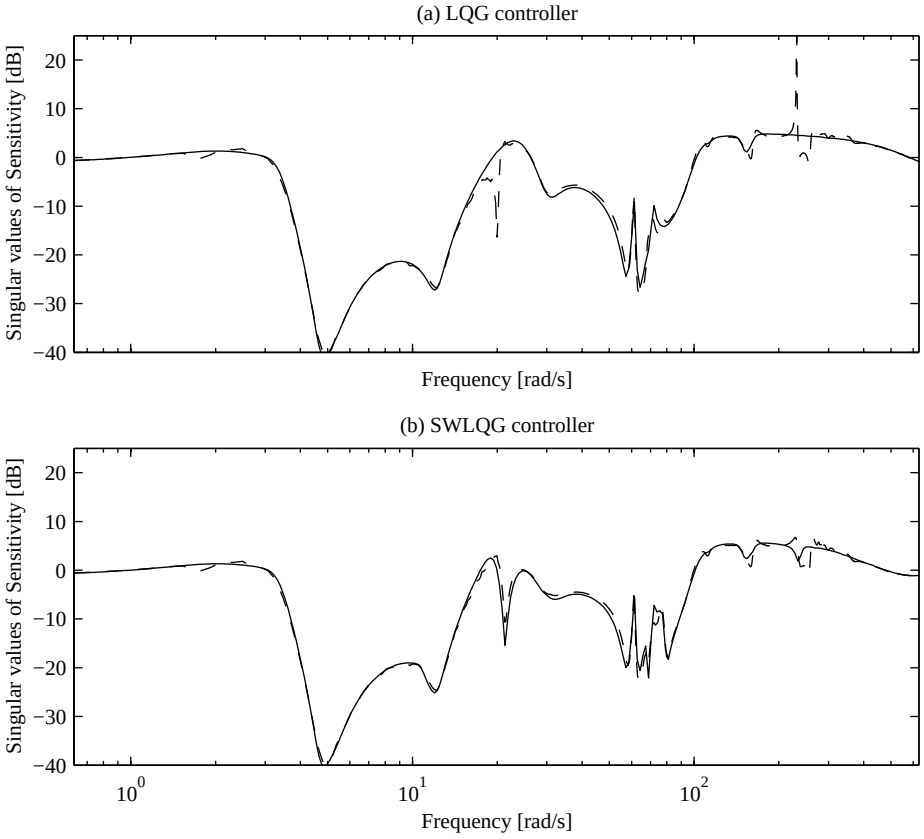

Figure 3.13 Maximum singular values of the sensitivity transfer function two different control designs. An LQG design is shown in (a); the evaluation on the design model (solid line) differs greatly from the evaluation on the experimental data (dotted line) at 230 rad/s. This indicates sensitivity of the closed loop at that frequency to small perturbations and potential stability problems. An SWLQG design is shown in (b); there are no large differences between the evaluation on the design model (solid line) and the evaluation on experimental data (dotted line).

Through extensive design experience [Grocott, 1994, Miller et al., 1996 and Mallory, 2000], a good design rule is that if the maximum sensitivity singular value of the controller evaluated on experimental data exceeds 10 dB and there are large differences between the evaluation on the design model and the evaluation on the experimental data, stability problems are likely to occur.

3.3.5 Analysis Without Experimental Data

Control designs are also performed where experimental data is not available. The analysis tools presented here can be applied using a nominal model and an error model. Performance and stability are evaluated simultaneously for multiple models within the bounds of the error model. Figure 3.14 shows a sample analysis for a model with frequency uncertainty. Three models are evaluated: a nominal model, a perturbed model with all uncertain frequencies shifted down, and a perturbed model with all uncertain frequencies shifted up.

An alternative method of stability analysis is the structured singular value μ [Balas et al., 2001], which was used to predict closed loop behavior for MACE-I 0-g controllers designed without on-orbit data [Miller et al., 1996] with a high degree of success. μ-analysis was not chosen in this work for two reasons:

- The MACE-I program used modal uncertainty description for μ-analysis. Because of the lack of 0-g *in situ* test data, modal statistics were obtained by testing in 1-g and then projected to 0-g. There is no method for propagating these modal statistics to a geometrically nonlinear system, so the modal uncertainty description cannot be used.
- The physically uncertainty description developed in Chapter 3, results in over 600 uncertain inputs and outputs. The size of this uncertainty structure renders μ-analysis too computationally intensive.

It is important to note that although the 0-g setpoint controllers were designed without experimental data, extensive 1-g design was performed using 1-g experimental data (and measurement models derived from experimental data). Data-based design was essential for hardware debugging, as well as during the early stages of model development. The results of the data-based designs provided insights into the limitations of performance and

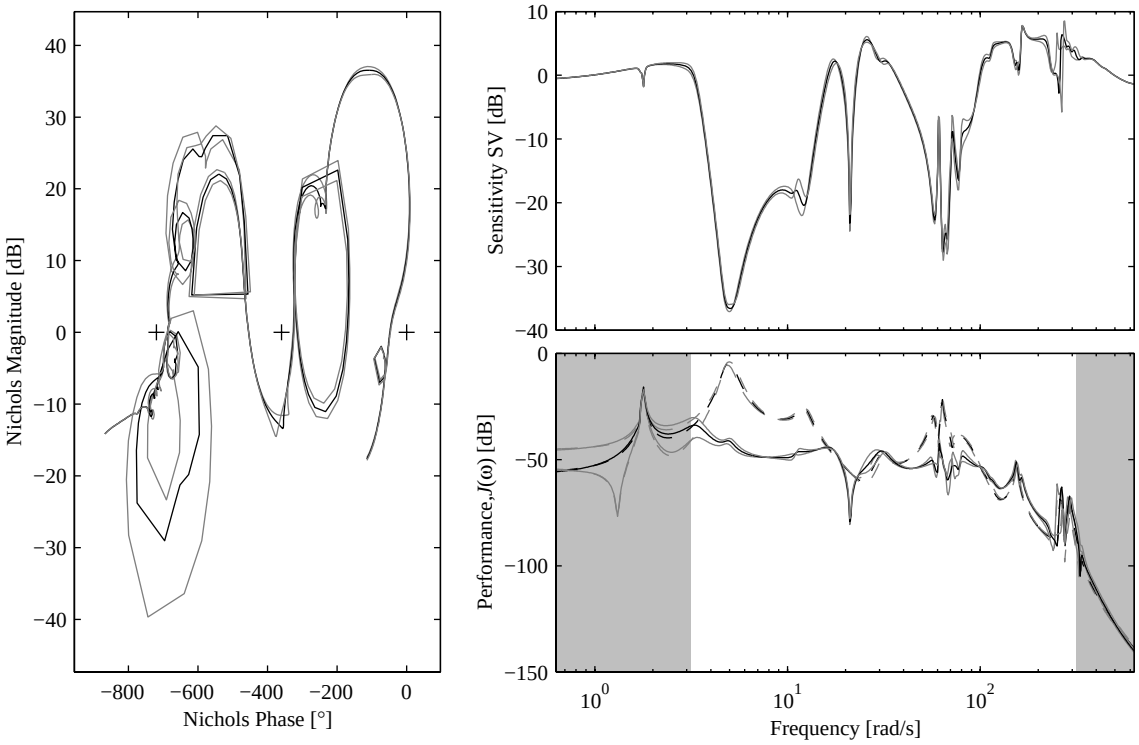

Figure 3.14 Analysis plots for nominal model (black) and perturbed models (grey). This simultaneous analysis indicates that the controller will perform robustly in the closed loop for all three models.

stability, as well as insight into the development of the uncertainty model. For example, in MACE-I, data-based designs were used to determine which modes had to be selected as part of the modal parameter uncertainty database. For MIT/MACE-II, data-based designs provided insight for the selection of the appropriate physical parameters to be updated and incorporated into the physical parameter uncertainty database. Specifically, the design process revealed inconsistencies in the frequencies of the modes dominated by the motion of the flexible appendages. These were eventually found to be uncertainties in the appendage attachment mechanisms.

Data-based control design also provides a useful benchmark as to the effectiveness of the overall modeling and control approach. A good indicator of over-conservativeness in the approach is if the performance of controllers designed using the model does not approach the performance of the data-based designs. Care must be taken to ensure that this "test" is

unbiased: the data-based designs must be valid over all the datasets used to generate the uncertainty model.

3.4 Chapter Summary

This chapter has presented several design tools for the synthesis and analysis of linear controllers. Graphical heuristics have been developed to aid in the design of LQG and SWLQG controllers.

Although more sophisticated linear control design techniques can be found in the literature, the closed-form nature of the solution to both LQG and SWLQG makes them attractive choices for gain-scheduled control. A requirement for gain scheduling methodology developed in the next chapter is that for a small change in the plant, the corresponding change in controller designs should also be small. Iterative methods used in other design techniques generally do not possess this “smoothness” property. An additional feature of the LQG/SWLQG solution is that the solution time to Riccati equation is fixed and relatively short, which is important because gain scheduling requires many linear designs to be performed.

Chapter 4

GAIN SCHEDULED CONTROLLER REALIZATION

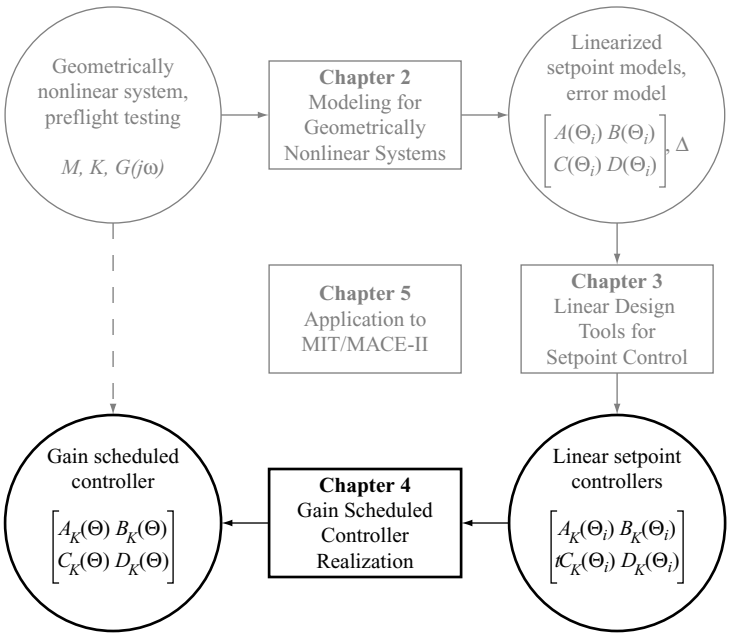

Figure 4.1 Gain scheduled realization in overall gain scheduling framework.

"Gain-scheduling" is a frequently used expression in the control literature that implies that controller properties are changed according to scheduling parameters. For geometrically nonlinear structures these scheduling parameters are the angles that describe the geometric configuration of the structure. Each geometric configuration is called a *setpoint*, and the modeling framework of Chapter 2 has been developed to provide linearized control design models at arbitrary setpoints. The tools of Chapter 3 can be used to design linear control-

lers for these linearized models. The focus of this chapter is the realization of a gain-scheduled controller from a family of linear controllers designed at different setpoints. This is summarized in a flowchart of the overall gain scheduling framework as shown in Figure 4.1.

The outline of this chapter is as follows. First the form of the gain scheduled controller is motivated and presented. The tools required to convert a family of linear controllers to this form, incorporating state matrix smoothing, reduction and interpolation as shown in Figure 4.2 are then developed. The analysis of gain-scheduled controllers is described next, and finally an integrated controller design procedure incorporating an iterative process of synthesis and analysis is presented.

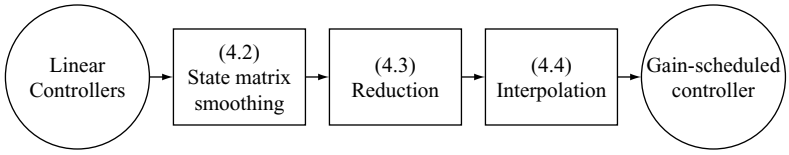

Figure 4.2 Procedure for converting a family of linear controllers into a gain-scheduled controller.

4.1 Controller Representation

4.1.1 Control Problem Formulation

The slow time variations of geometrically nonlinear systems suggest that a linearized approach to control synthesis and analysis is appropriate. The modeling effort of Chapter 2 provides a control design model,

$$\begin{bmatrix} z \\ y \end{bmatrix} = \begin{bmatrix} G_{zw}(\Theta, \Delta) & G_{zu}(\Theta, \Delta) \\ G_{yw}(\Theta, \Delta) & G_{yu}(\Theta, \Delta) \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix}, \tag{4.1}$$

so that an uncertain linearized plant can be determined at any setpoint, Θ, for design and analysis. The performance objective is to minimize the cost,

$$J = z^{\mathsf{T}}(\Theta, \Delta) z(\Theta, \Delta), \tag{4.2}$$

with a controller of the form,

$$u = K(\Theta) y, \tag{4.3}$$

within a prescribed operating space, Θ, and at a prescribed level of uncertainty, Δ. To make the analysis procedure tractable the operating space is sampled as a set of linearized plants,

$$\begin{bmatrix} z \\ y \end{bmatrix} = \begin{bmatrix} G_{zw}(\Theta_i, \Delta) & G_{zu}(\Theta_i, \Delta) \\ G_{yw}(\Theta_i, \Delta) & G_{yu}(\Theta_i, \Delta) \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix}, \; i = 1 \ldots n_P, \tag{4.4}$$

with the cost,

$$J(\Theta_i) = z^{\mathsf{T}}(\Theta_i, \Delta) z(\Theta_i, \Delta). \tag{4.5}$$

This assumes that a small change in Θ, will affect only a small change in the system dynamics and hence the performance. With this assumption the analysis at a single set-point is considered valid in a local region around that setpoint. The number of plants for analysis must therefore be large enough to capture the variations in system dynamics with respect to Θ.

4.1.2 Gain-Scheduled Controller Implementation

A Linear Parameter Varying (LPV) form will be used to describe the gain-scheduled controller,

$$\begin{bmatrix} \dot{x}_K \\ u \end{bmatrix} = \begin{bmatrix} A_K(\Theta) & B_K(\Theta) \\ C_K(\Theta) & D_K(\Theta) \end{bmatrix} \begin{bmatrix} x_K \\ y \end{bmatrix}. \tag{4.6}$$

This controller form allows linearized controllers to be computed for analysis and for real-time execution. If real-time computational resources were infinite, an optimal solution to the control problem such as LQG and SWLQG could be computed continuously, or at

every setpoint. However, one characteristic of control for structural systems is large order of the control design. The state order of the controller generally scales with the number of flexible modes in the bandwidth of the control loop. As such, many controller states are required to achieve high levels of performance. This large state order can place stringent requirements on the computational capabilities of the real-time control hardware, and precludes certain forms of gain scheduling (for example re-solving the controller Riccati equations) from being implemented.

The alternative is then to store pre-calculated controllers for multiple setpoints. To reduce the memory requirements and to ensure that the controllers vary smoothly between setpoints, an interpolation scheme must be implemented.[1] The first stage in designing a gain-scheduled controller is then to obtain linear controllers at several setpoints in the operating space,

$$\begin{bmatrix} \dot{x}_K \\ u \end{bmatrix} = \begin{bmatrix} A_K(\Theta_i) & B_K(\Theta_i) \\ C_K(\Theta_i) & D_K(\Theta_i) \end{bmatrix} \begin{bmatrix} x_K \\ y \end{bmatrix}, \; i = 1 \ldots n_K. \tag{4.7}$$

Linear control design techniques do not generally take into account the internal state vector of the state-space representation. Many combinations of state matrices can produce the same output behavior. Although this is not an issue for linear controllers, care must be taken to preserve the representation of the state vector as the system moves between setpoints in order to minimize undesirable switching transients.

This amounts to "smoothing" the state matrices $A_K(\Theta_i)$, $B_K(\Theta_i)$, and $C_K(\Theta_i)$. If the differences between the state matrices of neighboring setpoints are minimized, then the differences in state vector representation will also be minimized. Note that the D matrix is excluded from further discussion, because it does not influence the state vector.

1. A detailed evaluation of several different implementation methods in the context of MIT/MACE-II control requirements and computational capability is presented in Appendix A.

Although not explicitly used this thesis, the smoothness property can be defined as follows. Given a function $m(\theta)$ (such as a varying matrix element), the samples $m(\theta_i)$, $i = 1...n_\theta$, are considered *smooth* if for all neighboring triples $m(\theta_{i-1})$, $m(\theta_i)$, and $m(\theta_{i+1})$, the middle element $m(\theta_i)$ can be approximated with linear interpolation,

$$\hat{m}(\theta_i) \approx m(\theta_{i-1})\left(\frac{\theta_{i+1} - \theta_i}{\theta_{i+1} - \theta_{i-1}}\right) + m(\theta_{i+1})\left(\frac{\theta_i - \theta_{i-1}}{\theta_{i+1} - \theta_{i-1}}\right), \tag{4.8}$$

where,

$$\left|\hat{m}(\theta_i) - m(\theta_i)\right| < S, \tag{4.9}$$

and S is a smoothness specification. A function can be smoothed by increasing the density of the samples so long as it is Lipschitz continuous within its bounds. A function $m(\theta)$ is said to be Lipschitz continuous on the interval $[a,b]$ if and only if there is a constant $K \geq 0$ such that $\left|m(\theta_1) - m(\theta_2)\right| \leq K\left|\theta_1 - \theta_2\right|$ whenever θ_1 and θ_2 are both in the interval $[a,b]$. Note that this is a less restrictive criterion for continuity than differentiability (two examples are piece-wise linear functions and the absolute value function $|\theta|$ which are both Lipschitz continuous but not differentiable).

An additional note should be made at this point concerning the stability of the setpoint controllers used to generate the gain scheduled controllers. Both LQG and SWLQG synthesis methods can occasionally return controllers that are unstable in the open loop. While this is a valid solution to the control problem, two reasons preclude the use of controllers that have unstable poles. The first reason is motivated by practical concerns, namely that unstable controllers will fail ungracefully in the event of a sensor failure.

The second reason is more subtle, and is important when the controller is stable in some regions of setpoint space and unstable in others. This implies that the unstable controller is valid only for specific regions of the setpoint space. If this is the case, the switch of the controller between stable and unstable must be performed between exactly the appropriate setpoints, otherwise there will be some regions of setpoint space where the closed loop

system will be unstable. It is unlikely that this switching can be consistently performed at the exact setpoint, and thus the use of controllers that are stable in some regions of setpoint space and unstable in others is precluded.

4.2 State Matrix Smoothing

As is motivated in the previous section, the focus of this section is to develop a set of tools which preserve the internal state representation of a time-varying state-space system through setpoint space, *without changing the overall input-output behavior.* This objective is achieved in several steps, summarized in Figure 4.3. The family of linear systems

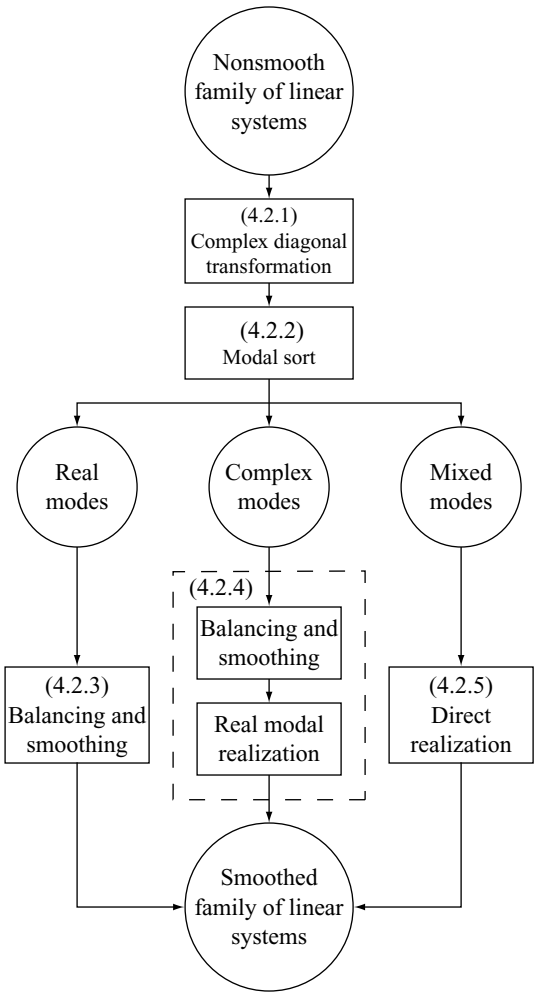

Figure 4.3 Flowchart for smoothing linear systems.

is first transformed into the complex diagonal representation, which decouples each system into modal dynamics. Using a mode sort, pole loci are assembled that correspond to the "true" loci of the nonlinear system from which the linearized systems are derived. These loci or varying modes can be divided into three categories; real modes, complex modes, or mixed modes. Mixed modes are defined as those which transition between real and complex modes. Each of these three types are realized into the smooth family of linear controllers using different techniques. Real modes are the most straightforward to realize with a balancing and smoothing technique. The realization of complex modes follows a similar process, but an additional step is required to realize the complex elements. This is the real modal realization. Finally mixed modes must be realized using alternative methods. The three sets of modes are finally combined to produced a smoothly varying set of linear systems.

A notational simplification is first made,

$$A^i \equiv A_K(\Theta_i)\,,\; B^i \equiv B_K(\Theta_i) \text{ and } C^i \equiv C_K(\Theta_i)\,, \tag{4.10}$$

so that the superscript $(\,\cdot\,)^i$ is an index to a list of setpoints.

4.2.1 Complex Diagonal Transformation

The first step in the smoothing process is to transform each state-space system $[A^i, B^i, C^i]$ to the complex diagonal realization. The A matrix of the complex diagonal realization has as entries the solution to eigenvalue problem $(A^i - \omega)\phi = 0$,

$$A^i_{[d]} = \text{diag}\left[\omega^{i,(1)}_{[c]}\;\; \omega^{i,(1)*}_{[c]} \;\cdots\; \omega^{i,(n_c)}_{[c]}\;\; \omega^{i,(n_c)*}_{[c]}\;\; \omega^{i,(1)}_{[r]} \;\cdots\; \omega^{i,(n_r)}_{[r]}\right]. \tag{4.11}$$

There are n_c complex conjugate pole pairs and n_r real poles for a total of $n_{st} = 2n_c + n_r$ states. The $(\cdot)^*$ notation denotes the conjugate operation. The $(\cdot)^{i,(j)}_{[r]}$ notation refers to the j^{th} real mode of the i^{th} setpoint. For example, $\omega^{1,(2)}_{[r]}$ refers to the second real eigenvalue of the first setpoint, and $\phi^{1,(2)}_{[r]}$ refers to the corresponding eigenvector. The $[c]$ sub-

script denotes a complex eigenvalue with a *positive* imaginary component. Since the original state-space matrices are real, every [*c*] term must have a matching complex conjugate term.

The B and C matrices are determined by assembling the eigenvectors of $(A^i - \omega)\phi = 0$ into an eigenvector matrix $\Phi^i_{[d]}$,

$$\Phi^i_{[d]} = \left[\phi^{i,(1)}_{[c]} \;\; \phi^{i,(1)*}_{[c]} \;\cdots\; \phi^{i,(n_c)}_{[c]} \;\; \phi^{i,(n_c)*}_{[c]} \;\; \phi^{i,(1)}_{[r]} \;\cdots\; \phi^{i,(n_r)}_{[r]}\right], \tag{4.12}$$

and using this matrix to transform the original state-space matrices,

$$B^i_{[d]} = \Phi^{i\;-1}_{[d]} B^i,\; C^i_{[d]} = C^i \Phi^i_{[d]}. \tag{4.13}$$

Note that, within the real and complex partitions, the ordering of the eigenvalues is unspecified. At this point, any ordering will suffice; this ambiguity will be resolved by the mode sort process (Section 4.2.2).

4.2.2 Mode Sort

In the complex diagonal realization, each diagonal entry of the *A* matrix is a pole of the system. In a complex diagonal nonlinear system, these entries are varying, and can be considered as pole loci in the complex plane. The mode sort procedure rearranges the state order of each setpoint to ensure that the pole loci correspond to the "true" loci of the nonlinear system from which the linearized systems are derived.

Sort order and reference list

Each unsorted setpoint must be assigned a reference system which is already sorted. For a one-dimensional setpoint space this can proceed linearly. The setpoints are ordered so that setpoints that are neighbors in setpoint space have consecutive indices. Setpoint 1 is selected as an initial reference setpoint, and setpoint 2 is sorted with respect to the setpoint 1. Setpoint 3 can now be sorted with respect to setpoint 2. This can be propagated from

one extreme of the setpoint space to the other, so that n_θ-1 sort functions are executed. The *sort order* is then [2,3,...,n_θ], and the *reference list* is [1,2,...,$n_{\theta-1}$].

For TV systems that have a setpoint space with multiple dimensions, the sort order and reference list should be carefully determined. This must done to ensure that only sorted systems are used as reference systems. A simple ordering that follows a "nearest neighbor" method may fail as the nearest neighbor may not be sorted. The initial reference setpoint is chosen as the setpoint closest to the geometric center of the setpoint space. The sort proceeds radially from the initial reference setpoint, so that setpoints closest to the center of the setpoint space are sorted first. The corresponding reference setpoint for each unsorted setpoint is the closed *sorted* setpoint.

Figure 4.4 shows two examples of a sort order and reference list for a two-dimensional setpoint space. Note that while any multiple-dimension TV system can be reordered so that the sort order is increasing ([2,3,...,n_θ]), the reference list will depend on the geometric arrangement in setpoint space.

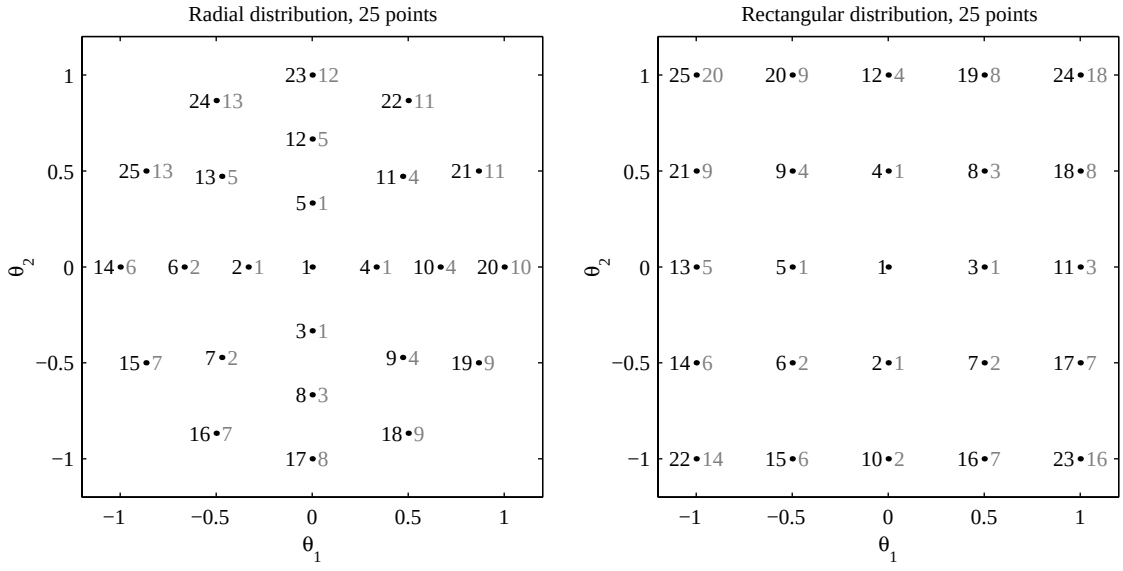

Figure 4.4 Sort orders for a 2D radial distribution and a 2D rectangular distribution. The numbers in black correspond to the order in which each setpoint will be sorted; the numbers in grey are the reference setpoint for that sort. In both cases, setpoint 1 is the initial reference system, and as such has no reference setpoint of its own.

Sorting function

First consider a reference setpoint, $[A^i, B^i, C^i, D^i]$, and an unsorted setpoint $[A^{i+1}, B^{i+1}, C^{i+1}, D^{i+1}]$. Both the reference and unsorted setpoints are the complex diagonal realization; indices are assigned to the poles of the reference setpoint. The distance in the complex plane of every pole in the unsorted setpoint to every pole in the reference setpoint is computed. Each pole of the unsorted setpoint is assigned the index of the closest pole of the reference setpoint. If the assigned indices are all different, the sort is complete, and the state vector of the unsorted setpoint is reordered to match the assigned indices.

If the same index is repeated twice, there is a conflict (Figure 4.5). Two of the poles in the unsorted setpoint have been matched up with the same pole in the reference setpoint. In this case, the next closest pole of the reference system is considered and the total complex distance of all combinations of indices is minimized.

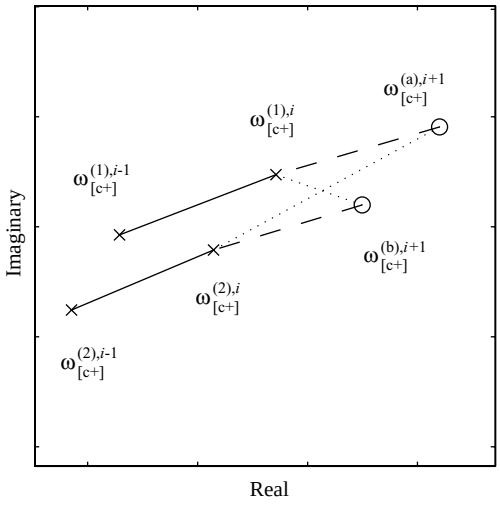

Figure 4.5 Sorting conflict resolution. The x's represent sorted poles of the (i-1)th and (i)th setpoints, and the o's are the unsorted poles of the (i+1)th setpoint. Both pole (a) and pole (b) of the (i+1)th setpoint are closest to pole (1) of setpoint (i); the next closest pole of setpoint (i) is pole(2). The possible sort results are shown by the dashed and dotted lines. The final solution is given by the dashed lines, since the total length of the dashed lines is smaller. Pole (a) will be assigned index (1), and pole (b) assigned index (2).

An example of the application of this sorting procedure is shown in Figure 4.7. The unsorted pole loci of a family of linear controllers has several large discontinuities. After the sorting procedure is applied, the pole loci appear to be “smoother”, without the large discontinuities of the unsorted system.

If the sorting algorithm fails to produce smooth pole loci, this is generally an indicator that the sampling density of setpoints is too low. In this case additional setpoints will be required to clear up sorting ambiguities. This is a verification step that must be performed by careful manual inspection of each pole locus.

The sort procedure produces a complex diagonal system with the state space matrices in three distinct partitions,

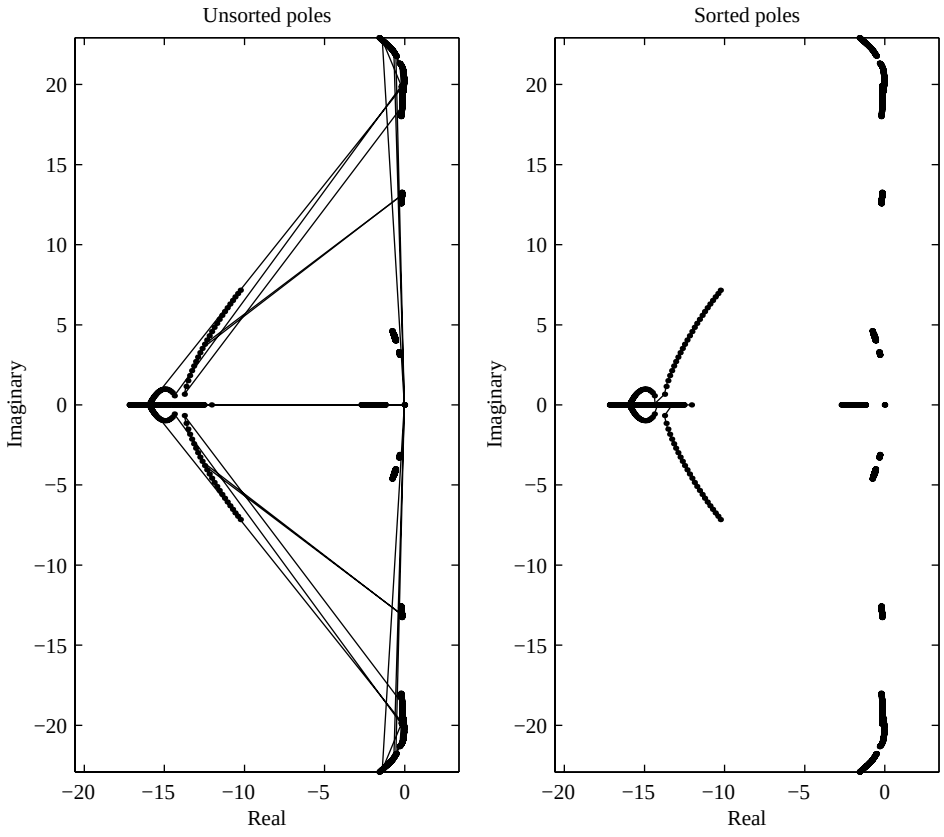

Figure 4.6 Sample TV system showing unsorted and sorted pole loci.

$$A^i = \begin{bmatrix} \Omega^i_{[c]} & & 0 \\ & \Omega^i_{[m]} & \\ 0 & & \Omega^i_{[r]} \end{bmatrix}, B^i = \begin{bmatrix} B^i_{[c]} \\ B^i_{[m]} \\ B^i_{[r]} \end{bmatrix}, C^i = \begin{bmatrix} C^i_{[c]} & C^i_{[m]} & C^i_{[r]} \end{bmatrix}. \quad (4.14)$$

The subscript [*c*] and [*r*] refer to complex and real modes as before, but the [*m*] subscript refers to *mixed* modes, where the modes change from real modes to complex modes and vice versa. Although mixed modes are rare in lightly damped structural systems, they can occur in servo loops where the damping of a resonant mode is close to critical. A small increase in damping will result in two real poles; or a small decrease in damping will result in a complex pole pair.

These three kinds of poles, real, complex and mixed must now be realized as smoothly varying *A*, *B* and C matrices.

4.2.3 Smoothing: Realization of Real Modes

A single real mode is represented by a scalar *A*, a vector *B* and a row vector *C*. The scalar A will vary smoothly since it is exactly the eigenvalue of that mode; there is no ambiguity. The *B* and *C* matrices do however have ambiguities. The elements of the *B* and *C* matrices of a typical varying real mode are shown in Figure 4.7. Even though this varying real mode has smoothly varying input-output properties, there are multiple discontinuities.

These discontinuities result from the fact that the eigenvector transformation used to transform the system into the real modal realization does not remove input-output scaling ambiguities; there remains an extra degree of freedom in the *B* and *C* matrices.

One alternative to removing the extra degree of freedom in the *B* and *C* matrices is to normalize with respect to a single element in either matrix. For example, the *B* matrix can be divided by its first element, and the *C* matrix multiplied by that same value. This would constrain the first element of the *B* matrix to be one, removing the extra degree of freedom while preserving the input-output properties.

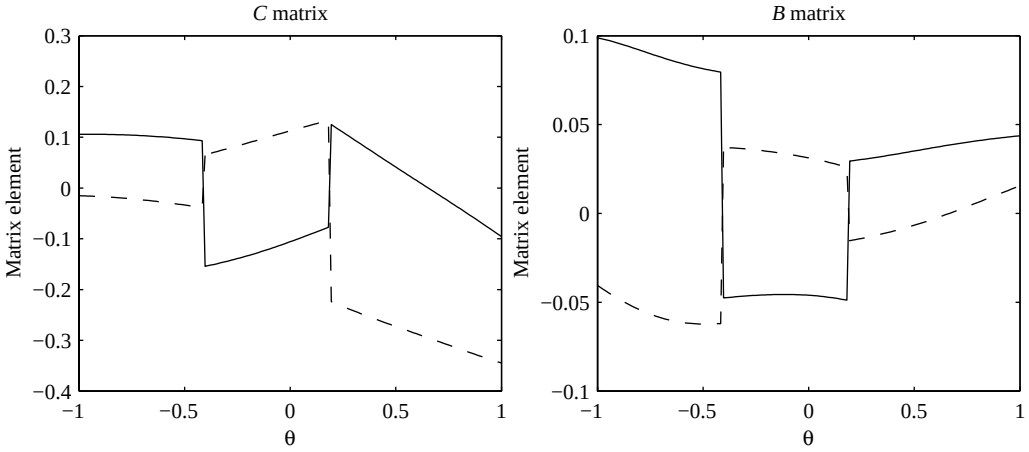

Figure 4.7 B and C matrices of a typical varying real mode before smoothing and balancing.

However, this method fails when states become unobservable or uncontrollable at some point in the setpoint space. Continuing the previous example, the normalization will result in a singular *C* matrix if the state is uncontrollable with respect to the first input channel at any point in the setpoint space.

The method that is used in this work to find smooth *B* and *C* matrices utilizes *modal residues*. If the eigenvalue and the input-output properties vary smoothly, clearly the modal residues must also vary smoothly. For a MIMO system, the modal residue is a n_oxn_i matrix, $\Xi_{[r]}$, that can be computed as,

$$\Xi_{[r]} \equiv C_{[r]}^{i,(j)} B_{[r]}^{i,(j)} . \tag{4.15}$$

Any B and C matrix of commensurate dimensions will produce the same input-output behavior so long as they preserve the modal residues. Figure 4.8 shows the eigenvalue, *B* matrix, *C* matrix and modal residues of the varying real mode of Figure 4.7. The modal residues are smooth even though the *B* and *C* matrices are not smooth.

It is intuitive that a smoothly varying factorization, $C_{[r]}^{i,(j)} B_{[r]}^{i,(j)}$, should exist for the smoothly varying modal residue matrix, $\Xi_{[r]}$. One choice of factorization utilizes the Singular Value Decomposition (SVD),

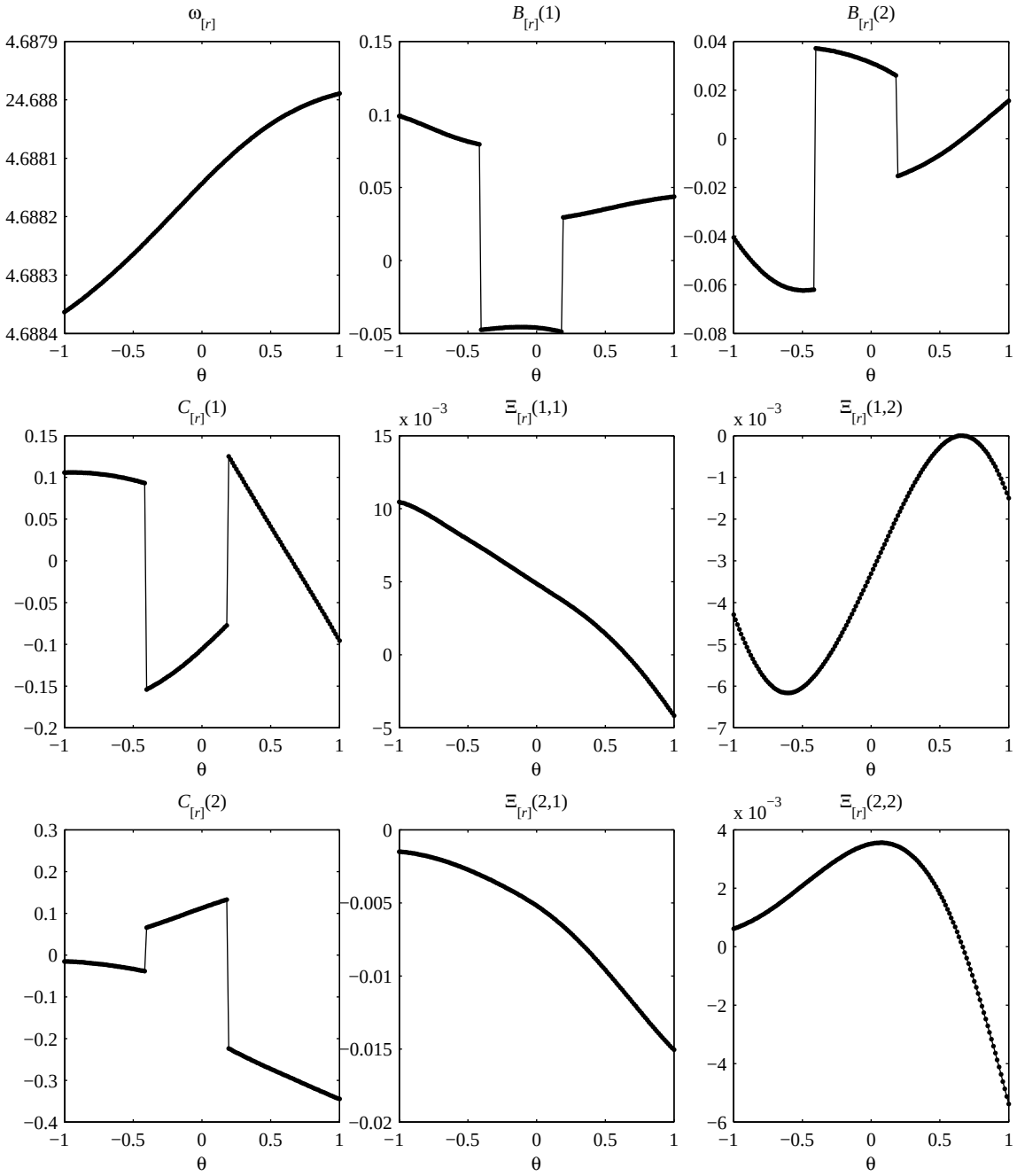

Figure 4.8 Modal residues for a typical varying real mode. Also shown are the eigenvalue and elements of the B and C matrices. Note that while the B and C matrices are discontinuous, the modal residues are smooth.

$$U_r \Sigma_r V_r^{\mathsf{T}} = \Xi_{[r]}, \tag{4.16}$$

where U_r and V_r are both unitary matrices, and Σ_r is a diagonal matrix of singular values. Note that because $C_{[r]}^{i,(j)}$ has one column and $B_{[r]}^{i,(j)}$ has one row, Σ_r reduces to a scalar and U_r and V_r reduce to vectors. $B_{[r]}^{i,(j)}$ and $C_{[r]}^{i,(j)}$ can be recovered using the following relations,

$$C_{[r]}^{i,(j)} \Leftarrow U_r \Sigma_r^{1/2}, \; B_{[r]}^{i,(j)} \Leftarrow \Sigma_r^{1/2} V_r^{\mathsf{T}}. \tag{4.17}$$

The square root of the singular value matrix removes scaling differences between the $B_{[r]}^{i,(j)}$ and $C_{[r]}^{i,(j)}$ matrices, in effect balancing the system and adding numerical stability.

Since the singular value, Σ_r, is unique for a given modal residue matrix, $\Xi_{[r]}$, and U_r and V_r are both unitary, the factorization is unique except for a sign ambiguity. The balanced factorization takes into account the relative magnitudes of the $B_{[r]}^{i,(j)}$ and $C_{[r]}^{i,(j)}$, but not their sign.

$$C_{[r]}^{i,(j)} \Leftarrow \pm U_r \Sigma_r^{1/2}, \; B_{[r]}^{i,(j)} \Leftarrow \pm \Sigma_r^{1/2} V_r^{\mathsf{T}}. \tag{4.18}$$

This sign ambiguity can only be resolved by a direct comparison with the corresponding mode of an adjacent system. This requires that the smoothing must be done in order, and that each system to be smoothed must have a system for comparison, or a reference system. The sort order and reference list of Section 4.2.2 can be used for this purpose. The final results of balancing and smoothing the varying real mode are shown in Figure 4.9. The *B* and *C* matrix terms now vary smoothly, and the modal residues are unchanged from that of Figure 4.8.

4.2.4 Smoothing: Realization of Complex Modes

The realization of a complex mode is performed using the complex *B* and *C* matrix elements of the complex diagonal realization. This allows the treatment of the complex

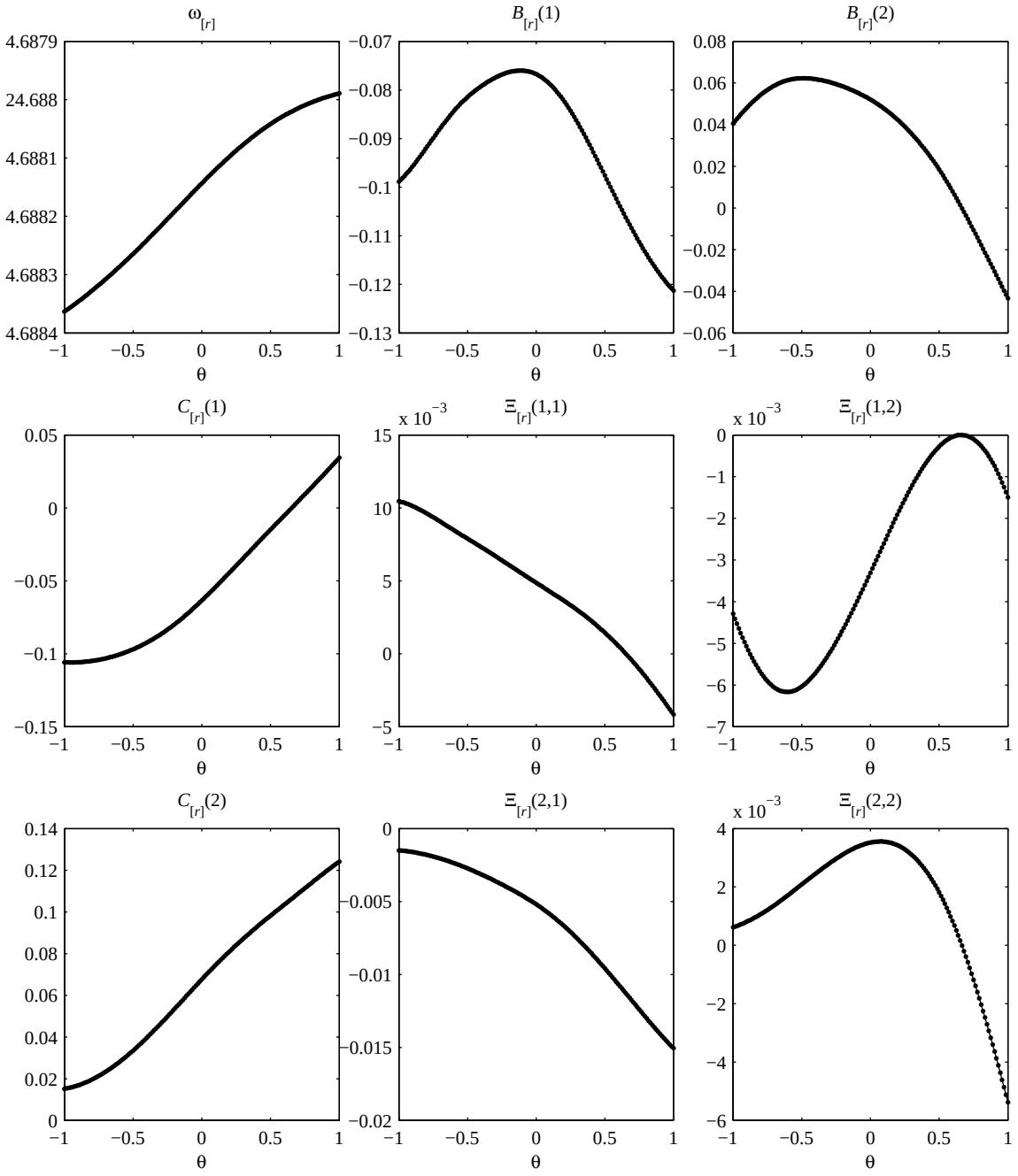

Figure 4.9 Eigenvalue, B and C matrices, and modal residues of the balanced and smoothed varying real mode. Compare to the corresponding plots of Figure 4.8. The eigenvalue and modal residues are the same, while the B and C matrix elements are now smooth and balanced.

mode with a single complex state, as opposed to two real states. The complex modal residue matrix, $\Xi_{[c]}$, can computed as,

$$\Xi_{[c]} \equiv C_{[c]}^{i,(j)} B_{[c]}^{i,(j)}. \tag{4.19}$$

Because all the relevant quantities are now complex, they can be visualized in the complex plane. Figure 4.10 shows the pole loci, complex *B* and *C* matrix terms, and the modal residues for a varying complex mode in the complex plane. As was the case with the real modal residues, the complex modal residues are continuous despite discontinuities in the B and C matrix terms.

The same operations for smoothing and balancing a real mode can then be applied to the complex mode. Balancing is performed via an SVD-based factorization, and the smoothing that addresses the sign ambiguity for real modes must now correct a complex phase ambiguity. The complex modal residue matrix is decomposed via an SVD, and a balanced factorization is computed,

$$U_c \Sigma_c V_c^{\mathsf{H}} = \Xi_{[c]}, \tag{4.20}$$

$$C_{[c]}^{i,(j)} \Leftarrow U_c \Sigma_c^{1/2},\ B_{[c]}^{i,(j)} \Leftarrow \Sigma_c^{1/2} V_c^{\mathsf{H}}. \tag{4.21}$$

Note that U_c and V_c are complex vectors, and Σ_c is a real scalar. Instead of the sign ambiguity of the real mode case, there is now a complex phase ambiguity. The complex matrices $C_{[c]}^{i,(j)}$ and $B_{[c]}^{i,(j)}$ are otherwise completely determined. The ambiguity can be expressed as a factor $e^{i\gamma}$ that multiplies (adds phases) to $C_{[c]}^{i,(j)}$ and divides into (subtracts phase from) $B_{[c]}^{i,(j)}$.

$$\Xi_{[c]} \equiv (C_{[c]}^{i,(j)} e^{j\gamma})(B_{[c]}^{i,(j)} e^{-j\gamma}). \tag{4.22}$$

Note that the transpose operation of Equations 4.17 and 4.18, $(\cdot)^{\mathsf{T}}$, has been replaced by the complex conjugate transpose operation, $(\cdot)^{\mathsf{H}}$.

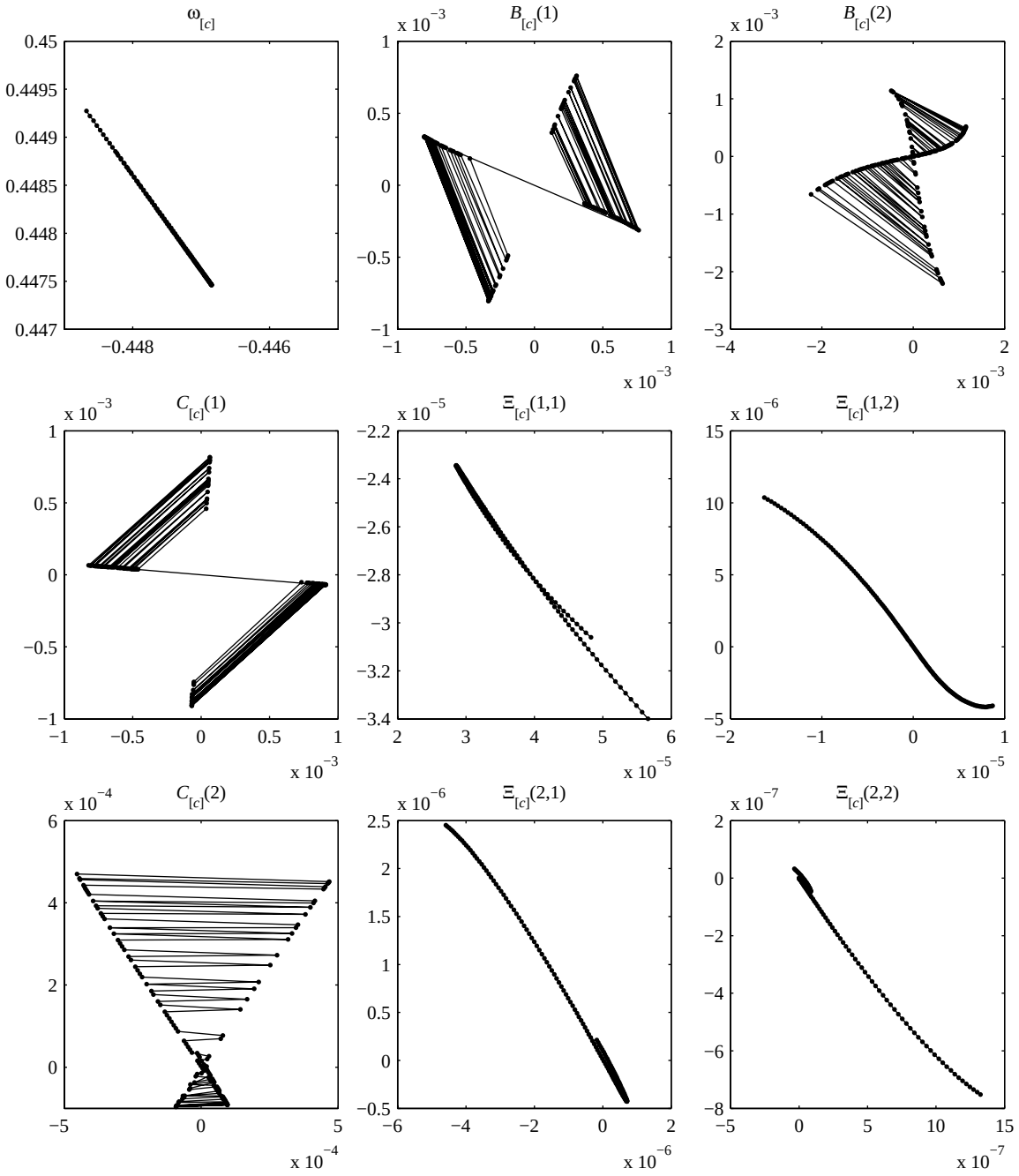

Figure 4.10 Pole locus, complex B and C terms, and modal residues of a sample complex mode before smoothing. All plots are in the complex plane. Note the discontinuities indicated by the large jumps in the input and output channels.

As with the smoothing of the real mode, this ambiguity must be resolved by comparison with an adjacent setpoint. Let the reference setpoint be denoted by the superscript i, and the unsmoothed system be denoted by the superscript i+1. The B matrix terms and the Hermitian of the C matrix terms are stacked as follows,

$$X_{[c]}^{i,(j)} = \begin{bmatrix} C_{[c]}^{i,(j)\mathsf{H}} \\ B_{[c]}^{i,(j)} \end{bmatrix}, X_{[c]}^{i+1,(j)} = \begin{bmatrix} C_{[c]}^{i+1,(j)\mathsf{H}} \\ B_{[c]}^{i+1,(j)} \end{bmatrix}. \tag{4.23}$$

The phase ambiguity is then resolved by minimizing the following cost,

$$J_{\Upsilon} = \sum_{X} \left| X_{[c]}^{i,(j)} - X_{[c]}^{i+1,(j)} e^{j\gamma} \right|. \tag{4.24}$$

The $\sum_{X}(\cdot)$ operator denotes a sum over the n_o+n_i elements of the X vector. The direction in which to minimize is given by the following complex quantity,

$$\begin{aligned} e^{j\gamma} = & \sum_{X} (\mathrm{Re}X_{[c]}^{i,(j)} \otimes \mathrm{Im}X_{[c]}^{i+1,(j)} - \mathrm{Im}X_{[c]}^{i,(j)} \otimes \mathrm{Re}X_{[c]}^{i+1,(j)}) \\ & + \sqrt{-1} \sum_{X} (\mathrm{Re}X_{[c]}^{i,(j)} \otimes \mathrm{Re}X_{[c]}^{i+1,(j)} - \mathrm{Im}X_{[c]}^{i,(j)} \otimes \mathrm{Im}X_{[c]}^{i+1,(j)}) \end{aligned} . \tag{4.25}$$

The $(\cdot \otimes \cdot)$ operation denotes an element-to-element multiplication of the arguments. (The minimizing phase is given by solving for ψ, using a four quadrant inverse tangent such as MATLAB atan2, but only $e^{j\gamma}$ is required). This formula weights the larger terms of $X_{[c]}^{i,(j)}$ so that the phase of very small terms (in the noise), is ignored. The final smoothed complex B and C matrices are calculated as,

$$C_{[c]}^{i,(j)} \Leftarrow C_{[c]}^{i,(j)} e^{j\gamma}, \; B_{[c]}^{i,(j)} \Leftarrow B_{[c]}^{i,(j)} e^{-j\gamma}. \tag{4.26}$$

Figure 4.11 shows the pole loci, complex B and C matrix terms, and the modal residues after balancing and smoothing. After the balancing and smoothing the modal residues have remained the same, and thus the input-output properties are preserved.

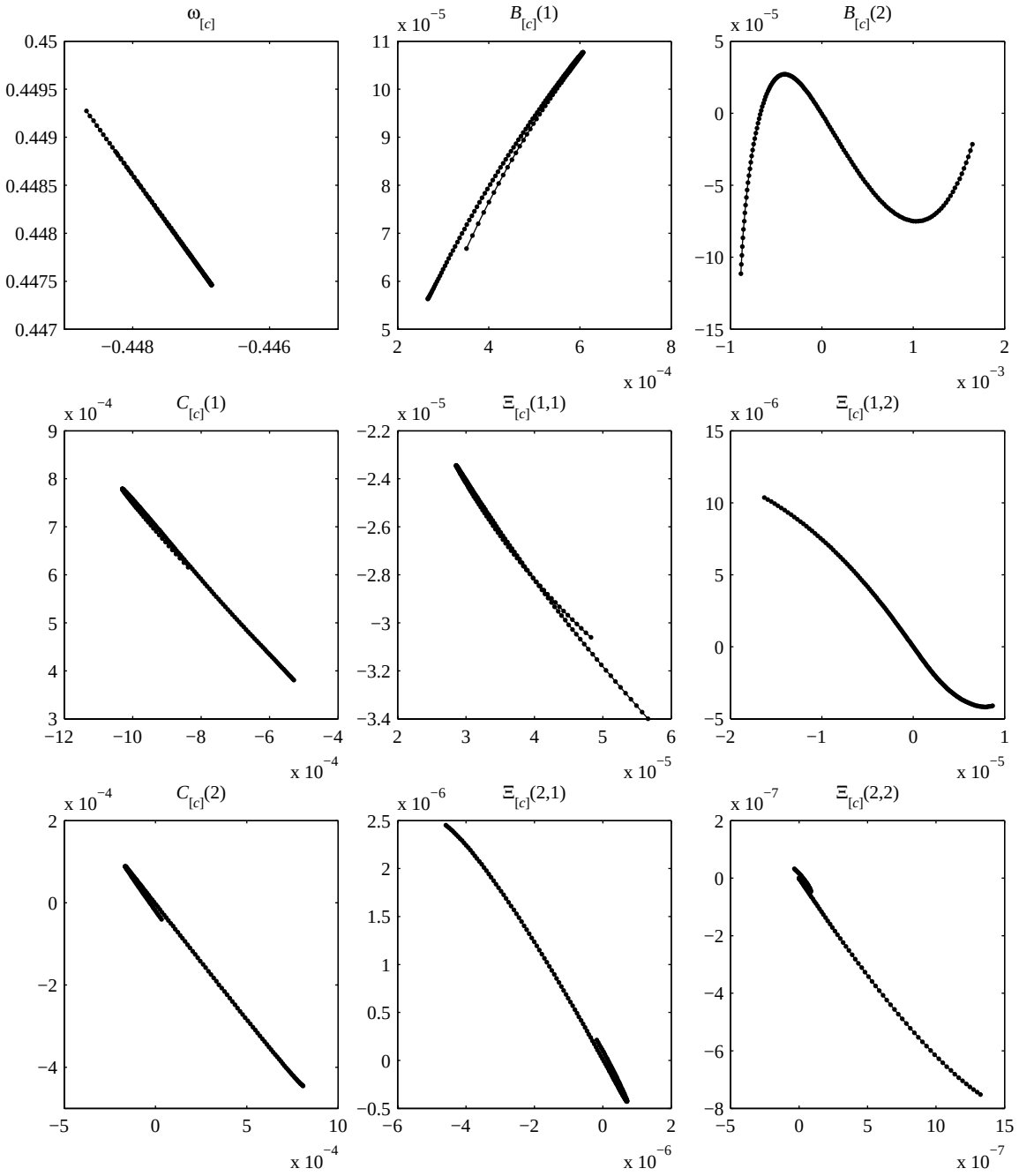

Figure 4.11 Pole locus, complex B and C terms, and modal residues of a sample complex mode after smoothing. All plots are in the complex plane. Compare with the unbalanced complex B and C terms of Figure 4.10. These terms are smooth, while the modal residues are unchanged, ensuring that the input output behavior of the system remains the same.

As was the case with the varying real modes, the smoothing operation is performed one setpoint at a time, with respect to an adjacent setpoint. The sort order and reference list is again used to determine the progression of the balancing and smoothing through setpoint space.

The final step is to realize the complex diagonal system into a state-space system with real matrices. The representation chosen is called the real modal realization, and can be affected with the following state transformation matrix,

$$T_{[rm]} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ j & -j \end{bmatrix}. \tag{4.27}$$

Since complex poles occur in real systems as complex conjugates, the transform simplifies to the following steps: The A matrix is given by,

$$A^{i,(j)}_{[rm]} = \begin{bmatrix} \mathrm{Re}\,\omega^{i,(j)}_{[c]} & -\mathrm{Im}\,\omega^{i,(j)}_{[c]} \\ \mathrm{Im}\,\omega^{i,(j)}_{[c]} & \mathrm{Re}\,\omega^{i,(j)}_{[c]} \end{bmatrix}, \tag{4.28}$$

and the B and C matrices by,

$$B^{i,(j)}_{[rm]} = \sqrt{2}\begin{bmatrix} \mathrm{Re}\,B^{i,(j)}_{[c]} \\ -\mathrm{Im}\,B^{i,(j)}_{[c]} \end{bmatrix},\ C^{i,(j)}_{[c+]} = \sqrt{2}\begin{bmatrix} \mathrm{Re}\,C^{i,(j)}_{[c]} & \mathrm{Im}\,C^{i,(j)}_{[c]} \end{bmatrix}. \tag{4.29}$$

This realization is chosen for multiple reasons. As previously mentioned, the block diagonal structure is efficient to implement. More importantly, the conversion from the complex diagonal realization to the real modal realization is very straightforward. If the A, B and C matrices of the complex diagonal system have smoothly varying entries, the matrices of the real modal system will also be smooth.

The elements of the real modal B and C matrices corresponding to the complex diagonal system of Figures 4.10 and 4.11 are shown in Figure 4.12. Before smoothing, the ele-

ments of the *B* and *C* matrices are highly discontinuous; the elements appear to be continuous after smoothing.

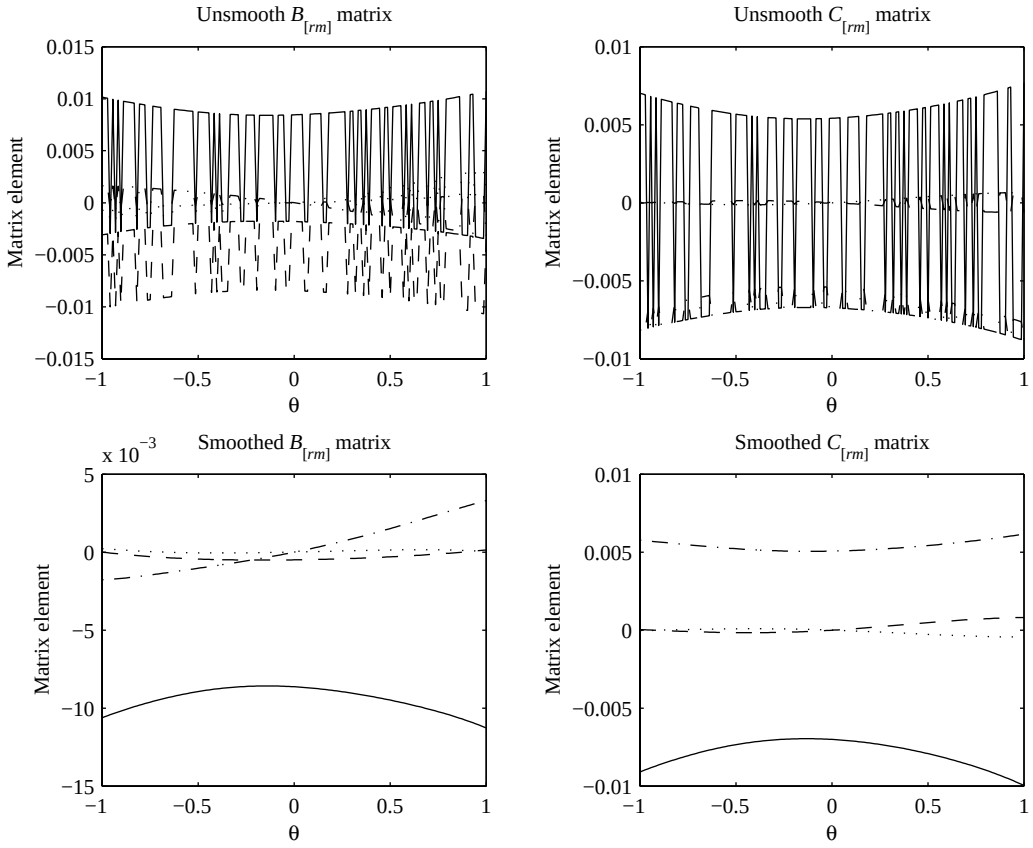

Figure 4.12 B and C matrix elements before and after smoothing.

4.2.5 Smoothing: Realization of Mixed Modes

Although both complex and mixed modes are represented by two real states, the real modal realization cannot be used for mixed modes, because it cannot capture both real and complex eigenvalues. A different realization must be used for mixed modes. The traditional second order form of,

$$A = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix} \tag{4.30}$$

does capture both real and complex modes, but the eigenvector matrices become singular as the poles coalesce, and thus modal residues are not appropriate. A different approach is required to generate smooth state-space matrices A, B and C.

Smoothing the real and complex modes begins with finding a smooth quantity, or the modal residues, and then reconstructing the B and C matrices from that smooth quantity. An equivalent smooth quantity can be determined for mixed modes. A MIMO second order state space system can be represented as,

$$G^{i,(j)}(s) = \frac{1}{s^2 + d_1^{(j),i} s + d_0^{(j),i}} [N_1^{i,(j)} s + N_0^{i,(j)}], \tag{4.31}$$

where the denominator terms $d_0^{i,(j)}$ and $d_1^{i,(j)}$ are derived from the characteristic equation of the system,

$$s^2 + d_1^{i,(j)} s + d_0^{i,(j)} = \left| sI - A_{[m]}^{i,(j)} \right|, \tag{4.32}$$

and the numerator matrices $N_0^{i,(j)}$ and $N_1^{i,(j)}$ are calculated from the following matrix relationships,

$$N_0^{i,(j)} = C_{[m]}^{i,(j)} \left[\frac{-(A_{[m]}^{i,(j)})^{-1}}{\left| A_{[m]}^{i,(j)} \right|} \right] B_{[m]}^{i,(j)}, \; N_1^{i,(j)} = C_{[m]}^{i,(j)} B_{[m]}^{i,(j)}. \tag{4.33}$$

The dimensions of N_0 and N_1 are then n_o x n_i. Together N_0 and N_1 are analogous to the modal residues of the real modes and the complex modes. Equation 4.31 is a special case of a Matrix Fraction Description (MFD), which is well described in the literature [Kailath, 1980].

The elements of a smoothly varying real mode are shown in Figure 4.13; there are clearly several discontinuities. The numerator and denominator matrices are computed for these unsmooth terms and shown in Figure 4.14. The numerator and denominator terms are smooth; a decomposition equivalent to the SVD based factorization used for the real modes and complex modes would allow for a similar smoothing process to be developed.

However no such decomposition was found. Two alternative methods for smoothing are proposed.

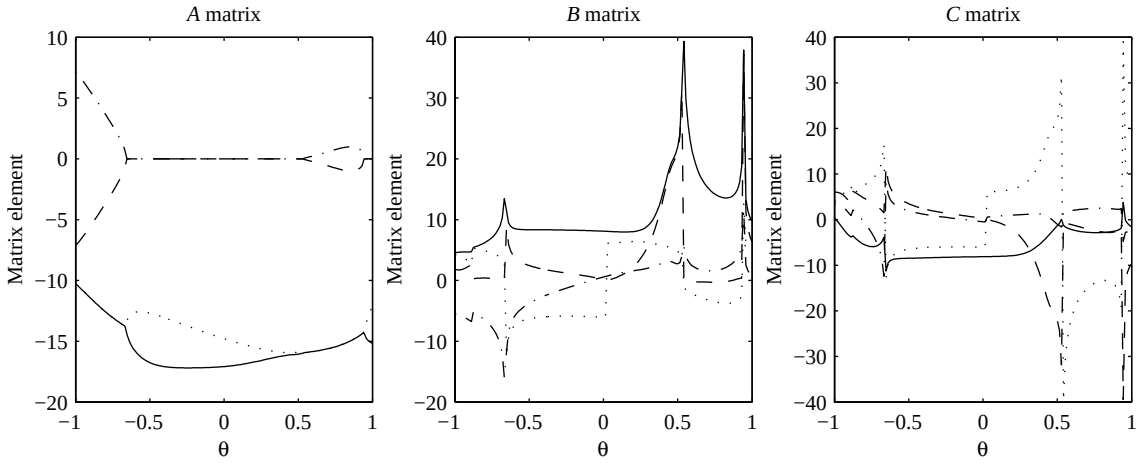

Figure 4.13 Matrix elements of a sample varying mixed mode before smoothing.

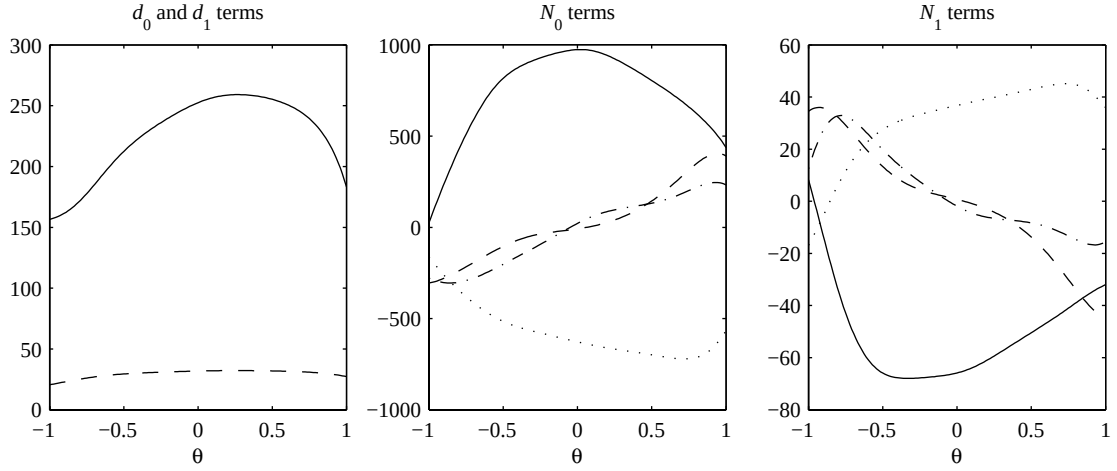

Figure 4.14 Denominator and numerator terms for transfer function representation of mixed mode. Where the matrix terms of Figure 4.13 are nonsmooth and discontinuous, these denominator and numerator terms are all smooth.

Direct conversion by realizing multiple systems

One method that works well when the either number of inputs or the number outputs is small is to directly realize the mode in controllable or observable canonical form [some

textbook reference here]. The disadvantage is that this involves adding states to the system; a mixed mode of 2 states, n_o outputs and n_i inputs results in a system with min(n_o,n_i)x2 states. If there are more outputs than inputs, the numerator matrices are partitioned into multiple columns,

$$N_0^{i,(j)} = \begin{bmatrix} n_0^1 & \cdots & n_0^{n_o} \end{bmatrix}, N_1^{i,(j)} = \begin{bmatrix} n_1^1 & \cdots & n_1^{n_o} \end{bmatrix}. \tag{4.34}$$

Each input corresponds to the column pair (n_0^k, n_1^k). A separate mixed mode is created for each input,

$$A_{[m]}^{i,(j)} = \begin{bmatrix} \begin{bmatrix} 0 & 1 \\ -d_0^{i,(j)} & -d_1^{i,(j)} \end{bmatrix} & & 0 \\ & \ddots & \\ 0 & & \begin{bmatrix} 0 & 1 \\ -d_0^{i,(j)} & -d_1^{i,(j)} \end{bmatrix} \end{bmatrix}, B_{[m]}^{i,(j)} = \begin{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} & & 0 \\ & \ddots & \\ 0 & & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{bmatrix},$$

$$C_{[m]}^{i,(j)} = \begin{bmatrix} \begin{bmatrix} n_0^1 & n_1^1 \end{bmatrix} & \cdots & \begin{bmatrix} n_0^{n_o} & n_1^{n_o} \end{bmatrix} \end{bmatrix}. \tag{4.35}$$

If there are more outputs than inputs, the numerator matrices are partitioned into multiple rows,

$$N_0^{i,(j)} = \begin{bmatrix} n_0^1 \\ \vdots \\ n_0^{n_i} \end{bmatrix}, N_1^{i,(j)} = \begin{bmatrix} n_1^1 \\ \vdots \\ n_1^{n_i} \end{bmatrix}. \tag{4.36}$$

Each output corresponds to the row pair (n_0^k, n_1^k). A separate mixed mode is created for each output,

$$A_{[m]}^{i,(j)} = \begin{bmatrix} \begin{bmatrix} 0 & -d_0^{i,(j)} \\ 1 & -d_1^{i,(j)} \end{bmatrix} & & 0 \\ & \ddots & \\ 0 & & \begin{bmatrix} 0 & -d_0^{i,(j)} \\ 1 & -d_1^{i,(j)} \end{bmatrix} \end{bmatrix}, \; B_{[m]}^{i,(j)} = \begin{bmatrix} \begin{bmatrix} n_0^1 \\ n_1^{n_i} \end{bmatrix} \\ \vdots \\ \begin{bmatrix} n_0^1 \\ n_1^{n_i} \end{bmatrix} \end{bmatrix},$$

$$C_{[m]}^{i,(j)} = \begin{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} & & \\ & \ddots & \\ & & \begin{bmatrix} 0 & 1 \end{bmatrix} \end{bmatrix}. \tag{4.37}$$

Since the numerator and denominator terms are directly substituted into the state-space matrices, this technique guarantees that the resulting system will be smooth if the numerator and denominator terms are smooth.

Pseudo-least squares match

An alternative technique is also proposed that does not increase the number of states. The smoothing problem can be stated in terms of finding the minimizing transform M. With a reference two state system $[A_{[m]}^{i,(j)}, B_{[m]}^{i,(j)}, C_{[m]}^{i,(j)}]$ and an initial unsmoothed adjacent two state system $[A_{[m]}^{i+1,(j)}, B_{[m]}^{i+1,(j)}, C_{[m]}^{i+1,(j)}]$, find a transformation M such that the transformed system $[M^{-1}A_{[m]}^{i+1,(j)}M, M^{-1}B_{[m]}^{i+1,(j)}, C_{[m]}^{i+1,(j)}M]$ is as similar as possible to the reference system.

This is a minimization problem over the four elements of the transformation matrix M, minimizing the three quantities:

$$\begin{aligned} \tilde{J}_A &= \left\| A_{[m]}^{i,(j)} - M^{-1}A_{[m]}^{i+1,(j)}M \right\| \\ \tilde{J}_B &= \left\| B_{[m]}^{i,(j)} - M^{-1}B_{[m]}^{i+1,(j)} \right\| \\ \tilde{J}_C &= \left\| C_{[m]}^{i,(j)} - C_{[m]}^{i+1,(j)}M \right\| \end{aligned} \quad . \tag{4.38}$$

This can be transformed into a linear least squares minimization by modifying the costs. The J_A and J_B costs are pre-multiplied by the matrix M to remove the inverse matrix operation:

$$\begin{aligned} J_A &= \left\| MA_{[m]}^{i,(j)} - A_{[m]}^{i+1,(j)} M \right\| \\ J_B &= \left\| MB_{[m]}^{i,(j)} - B_{[m]}^{i+1,(j)} \right\| \\ J_C &= \left\| C_{[m]}^{i,(j)} - C_{[m]}^{i+1,(j)} M \right\| \end{aligned} \quad . \tag{4.39}$$

When written in this form, the minimization is linear in terms of the elements of transformation matrix M,

$$M = \begin{bmatrix} M^{11} & M^{12} \\ M^{21} & M^{22} \end{bmatrix} . \tag{4.40}$$

The minimization is over the individual elements of the J_A, J_B and J_C costs. Expanding the reference system as,

$$A_{[m]}^{i,(j)} = \begin{bmatrix} A_1^{11} & A_1^{12} \\ A_1^{21} & A_1^{22} \end{bmatrix}, B_{[m]}^{i,(j)} = \begin{bmatrix} B_1^1 \\ B_1^2 \end{bmatrix}, C_{[m]}^{i,(j)} = \begin{bmatrix} C_1^1 & C_1^2 \end{bmatrix}, \tag{4.41}$$

and the unsmooth system as,

$$A_{[m]}^{i+1,(j)} = \begin{bmatrix} A_2^{11} & A_2^{12} \\ A_2^{21} & A_2^{22} \end{bmatrix}, B_{[m]}^{i+1,(j)} = \begin{bmatrix} B_2^1 \\ B_2^2 \end{bmatrix}, C_{[m]}^{i+1,(j)} = \begin{bmatrix} C_2^1 & C_2^2 \end{bmatrix}. \tag{4.42}$$

Remember that B_1, B_2 are 2xn_i matrices, and C_1 and C_2 are n_ox2 matrices.

$$\begin{bmatrix} A_1^{11}-A_2^{11} & A_1^{21} & -A_2^{12} & 0 \\ A_1^{12} & A_1^{22}-A_2^{11} & 0 & -A_2^{12} \\ -A_2^{21} & 0 & A_1^{11}-A_2^{22} & A_1^{21} \\ 0 & -A_2^{21} & A_1^{12} & A_1^{22}-A_2^{22} \\ C_2^{1\mathsf{T}} & 0 & C_2^{2\mathsf{T}} & 0 \\ 0 & C_2^{1\mathsf{T}} & 0 & C_2^{2\mathsf{T}} \\ B_1^1 & B_1^2 & 0 & 0 \\ 0 & 0 & B_1^1 & B_1^2 \end{bmatrix} \begin{bmatrix} M^{11} \\ M^{12} \\ M^{21} \\ M^{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ C_1^{1\mathsf{T}} \\ C_1^{2\mathsf{T}} \\ B_2^1 \\ B_2^2 \end{bmatrix}. \tag{4.43}$$

$$A_{[m]}^{i+1,(j)} \Leftarrow M^{-1} A_{[m]}^{i+1,(j)} M,\ B_{[m]}^{i+1,(j)} \Leftarrow M^{-1} B_{[m]}^{i+1,(j)},\ C_{[m]}^{i+1,(j)} \Leftarrow C_{[m]}^{i+1,(j)} M \tag{4.44}$$

This is a simple least squares fit and can be solved with a pseudo-inverse. As was the case with real modes and complex modes, this tool operates by comparing one setpoint to an adjacent setpoint. The tool is applied to all the setpoints following a sort order and corresponding reference list.

It was found the best results were obtained by first converting the varying mixed mode to the real modal realization, so that complex modes have a skew-symmetric A matrix and real modes have a diagonal A matrix. The results of smoothing are shown in Figure 4.15. Note that all four elements of the A matrix may change independently; whereas in the real modal representation there are only two independent elements.

This section has presented a procedure for smoothing the state matrices for a set of linear controllers that correspond to a nonlinear controller linearized at multiple setpoints. Each state matrix is isomorphically transformed so that the input-output behavior is exactly preserved, but the differences between state matrices, A, B and C, are minimized so that the variations of each matrix element are smooth in setpoint space.

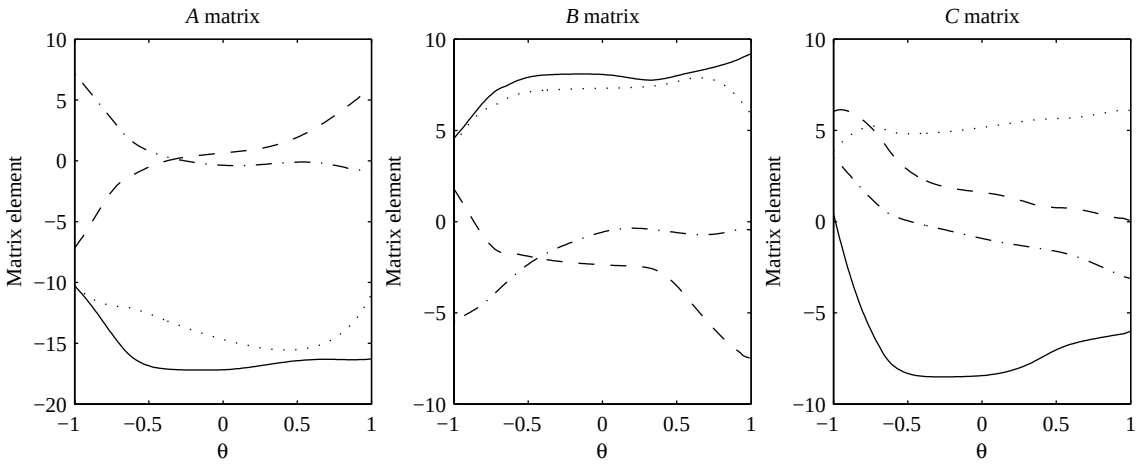

Figure 4.15 Matrix elements of a sample varying mixed mode after pseudo-least squares smoothing.

4.3 Reduction

System reduction is a commonly used procedure in the analysis and control synthesis of linear systems. The objective of reduction is to determine a system of reduced state order that approximates the input-output behavior of the original system. Reduction is important in particular for the large state-space systems required for the representation of flexible structures; both off-line and on-line computational requirements scale with the state order of the system.

The standard technique for reducing state-space systems is a balanced reduction [Moore, 1981]. This involves transforming the system to a balanced form so that the observability and controllability gramians are the same. The states can then be ranked by their Hankel Singular Values, and truncated to either a desired state order, or a desired level of fidelity.

This reduction procedure does not work for time varying systems. The balancing transformation previously described does not preserve the smoothly varying nature of the state matrices; thus any truncation may also result in state matrices that are not smooth. The reduction approach that is taken in this thesis is to determine a cost for each varying mode throughout setpoint space, and to retain the largest cost for that mode. This cost is used to rank the modes for truncation.

Two possible modal costs are considered here. The first is a direct extension of the balanced reduction technique. Each mode is taken as a separate system, and the Hankel Singular Values of each system are computed as modal costs. For complex and mixed modes, the modal costs are computed as,

$$\begin{aligned} A^{i,(j)}_{[c/m]}P + PA^{i,(j)\ \mathrm{T}}_{[c/m]} + B^{i,(j)}_{[c/m]}B^{i,(j)\ \mathrm{T}}_{[c/m]} &= 0 \\ A^{i,(j)\ \mathrm{T}}_{[c/m]}Q + QA^{i,(j)}_{[c/m]} + C^{i,(j)\ \mathrm{T}}_{[c/m]}C^{i,(j)}_{[c/m]} &= 0 \end{aligned}, \tag{4.45}$$

$$R^{i,(j)}_{[c/m]} = \max_{\lambda}[\lambda^{1/2}(PQ)]\,. \tag{4.46}$$

This simplifies for real modes since each real mode corresponds to only one state,

$$\begin{aligned} 2\omega^{i,(j)}_{[r]}P_j + B^{i,(j)}_{[r]}B^{i,(j)\mathrm{T}}_{[r]} &= 0 \\ 2\omega^{i,(j)}_{[r]}Q_j + C^{i,(j)\mathrm{T}}_{[r]}C^{i,(j)}_{[r]} &= 0 \end{aligned} \quad \begin{aligned} \omega^{i,(j)}_{[r]}P_j &= \frac{-B^{i,(j)}_{[r]}B^{i,(j)\mathrm{T}}_{[r]}}{2} \\ \omega^{i,(j)}_{[r]}Q_j &= -\frac{C^{i,(j)\mathrm{T}}_{[r]}C^{i,(j)}_{[r]}}{2} \end{aligned}, \tag{4.47}$$

$$R^{i,(j)}_{[r]} \equiv \sqrt{(B^{i,(j)}_{[r]}B^{i,(j)\mathrm{T}}_{[r]})(C^{i,(j)\mathrm{T}}_{[r]}C^{i,(j)}_{[r]})}\,. \tag{4.48}$$

An alternative cost considers the log based error introduced by the removal of that particular mode,

$$R^{i,(j)} \equiv \sum_{\omega} \log\left|\frac{G(\omega) - (G_{2\mathrm{x}2}(\omega))_i}{G(\omega)}\right|. \tag{4.49}$$

A log-error based cost has been shown to be an effective cost for the optimization of reduced order state-space systems [Jacques, 1995]. Although less accurate in an RMS sense, the log-error cost is more appropriate for control design since it minimizes differences in the log-scaled frequency response.

Modal costs can now be computed for any setpoint that is available in state-space form. The cost of a mode over the entire setpoint space is chosen as the maximum cost of that mode over all setpoints,

$$R^i \equiv \max_j [R^{i,(j)}] . \tag{4.50}$$

In this way, modes that are poorly observable or controllable in certain setpoints, but highly observable or controllable in other setpoints will have a higher cost than modes which have intermediate observability and controllability throughout setpoint space.

The actual number of modes to truncate depends on the application. The control designer may wish to preserve a high level of fidelity, in which case fewer modes will be truncated. Alternatively, a more aggressive truncation may be useful if numerical efficiency is required.

4.4 Interpolation

The procedures described in Sections 4.2 and 4.3 produce a reduced order time-varying system with smooth variations in the state matrices. For the purposes of control design, this system can be specified at a very large number of setpoints. However practical constraints of real-time control hardware often require that a much smaller number of controllers to be stored. To obtain controllers at arbitrary setpoints, an interpolation procedure must be thus be implemented.

Since the state matrices have been transformed so that they vary smoothly, this interpolation procedure can be applied on an element-by-element basis. Two methods of interpolation are investigated for the purposes of gain-scheduled control. The first is a direct scheme that stores multiple controllers corresponding to different setpoints. The second scheme “fits” the variations of the element over the entire setpoint space to a family of shapefunctions. These interpolation schemes were chosen because the interpolation can be performed quickly, and thus real-time update rate can be maximized.

4.4.1 Nearest Neighbor and Linear Interpolation

The simplest interpolation scheme is to use the pre-computed controllers as a lookup table. Each entry of the lookup table corresponds to a pre-computed controller at a *reference* set-point. The interpolated controller is taken to be the closest predetermined controller, as determined by setpoint space coordinates. This nearest neighbor interpolation scheme is very simple in implementation, but if the number of reference setpoints is too small signif-icant discrete jumps will be seen as the controller moves through setpoint space.

A improvement to this scheme can be achieved with linear interpolation. This ensures that the element variation will be continuous and piece-wise linear. Figure 4.16 shows an example of a varying element that is approximated with both interpolation schemes. As expected, both interpolation schemes match the pre-computed (reference) element exactly at the reference setpoints. The nearest neighbor interpolation scheme results in several discontinuities; and the linear interpolation scheme provides a continuous piece-wise lin-ear approximation. The error in of the nearest neighbor scheme is much greater than that of the linear interpolation scheme.

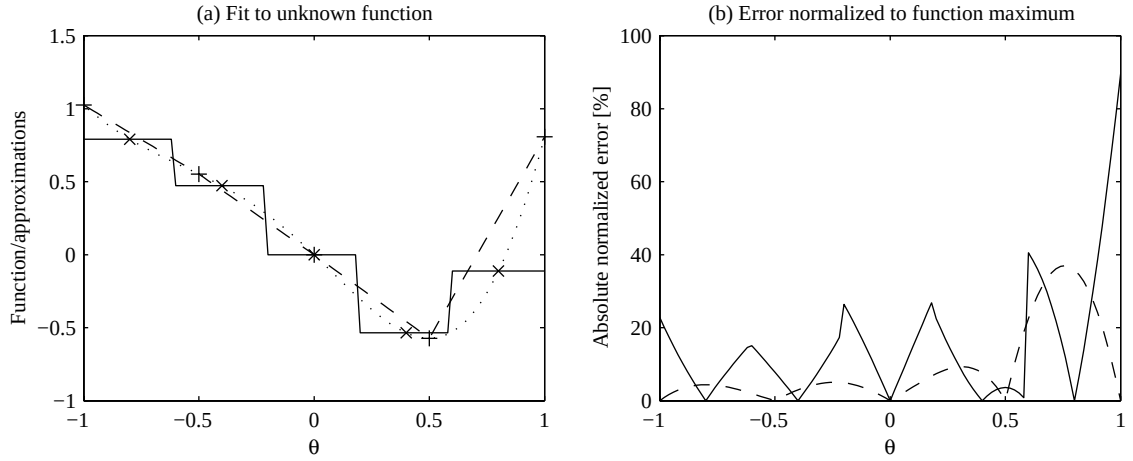

Figure 4.16 Interpolation for an unknown function and corresponding normalized error. The unknown function is shown as a dotted line. The nearest neighbor interpolation scheme is shown as a solid line; the linear interpolation scheme is shown as a dashed line. Both interpolation schemes are implemented using reference points. For the nearest neighbor interpolation scheme, the reference (pre-computed) points are marked as ‘x’ symbols; for the linear interpolation scheme the reference points are marked as ‘+’ symbols.

These direct interpolation schemes are readily extended to setpoint spaces with multiple dimensions. It should be noted that the storage requirements scale rapidly with the dimensionality of the setpoint space.

4.4.2 Global Shapefunction Representation

An alternative approach to interpolation is to fit the element variations with a pre-determined family of shapefunctions using a least square fit. For a varying matrix $M(\Theta)$, define each element as $m_{a,b}(\Theta)$. Each varying element is specified only at a discrete number of setpoints $m_{a,b}(\Theta^i)$. To interpolate between this set of points, the element is parameterized as the sum of multiple shapefunctions weighted by coefficients,

$$m_{a,\,b}(\Theta) \approx \sum_{j\,=\,1}^{n_f} m_{a,\,b}^{(j)} \Psi_j(\Theta)\,. \tag{4.51}$$

The setpoint vector consists of multiple angles each corresponding to a single dimension of the setpoint space, $\Theta = [\theta^1,\theta^2,...,\theta^{nd}]^\mathsf{T}$. The shapefunction $\Psi_j(\Theta)$ is a product of basis functions, $\psi_k(\theta^i)$. Once the set of basis functions is selected, a mapping is created that assembles the shapefunctions from the basis functions. For example, if the setpoint space has three dimensions and two shapefunctions are chosen the mapping is,

$$\begin{aligned}
\Psi_1(\Theta) &= \psi_1(\theta^1)\psi_1(\theta^2)\psi_1(\theta^3) & \qquad \Psi_5(\Theta) &= \psi_2(\theta^1)\psi_1(\theta^2)\psi_1(\theta^3)\\
\Psi_2(\Theta) &= \psi_1(\theta^1)\psi_1(\theta^2)\psi_2(\theta^3) & \qquad \Psi_6(\Theta) &= \psi_2(\theta^1)\psi_1(\theta^2)\psi_2(\theta^3)\\
\Psi_3(\Theta) &= \psi_1(\theta^1)\psi_2(\theta^2)\psi_1(\theta^3) & \qquad \Psi_7(\Theta) &= \psi_2(\theta^1)\psi_2(\theta^2)\psi_1(\theta^3)\\
\Psi_4(\Theta) &= \psi_1(\theta^1)\psi_2(\theta^2)\psi_2(\theta^3) & \qquad \Psi_8(\Theta) &= \psi_2(\theta^1)\psi_2(\theta^2)\psi_2(\theta^3)
\end{aligned}\,. \tag{4.52}$$

For a one-dimension setpoint space the number of shapefunctions is thus the same as the number of basis functions chosen. Chebyshev polynomials and sinusoids were considered in this work as they are simple to compute; this was a consideration for real-time implementation. Note the emphasis on *global* shapefunctions, as opposed to the local shape-

functions of which nearest neighbor and linear interpolation are a effectively a subset of. Examples of these shapefunctions are shown in Figure 4.17.

Once the choice of shapefunctions is established, the matrix elements are fit using a simple least squares fit. The only requirement is that there are many more setpoints, n_θ, than there are shapefunctions, n_f. Each matrix element is approximated at every pre-computed setpoint,

$$\begin{bmatrix} m_{a,b}(\Theta^1) \\ m_{a,b}(\Theta^2) \\ \vdots \\ m_{a,b}(\Theta^{n_\theta}) \end{bmatrix} \approx \begin{bmatrix} \Psi_1(\Theta^1) & \Psi_2(\Theta^1) & \cdots & \Psi_{n_f}(\Theta^1) \\ \Psi_1(\Theta^2) & \Psi_1(\Theta^2) & \cdots & \Psi_{n_f}(\Theta^2) \\ & \vdots & & \\ \Psi_1(\Theta^{n_\theta}) & \Psi_1(\Theta^{n_\theta}) & \cdots & \Psi_{n_f}(\Theta^{n_\theta}) \end{bmatrix} \begin{bmatrix} m_{a,b}^{(1)} \\ m_{a,b}^{(2)} \\ \vdots \\ m_{a,b}^{(n_f)} \end{bmatrix}. \tag{4.53}$$

A pseudo-inverse is then used to determine the coefficients $m_{a,b}^{(i)}$. A example of fitting an unknown function with Chebyshev polynomials and sinusoidal functions is shown in Figure 4.18. The Chebyshev polynomial achieves a better overall fit to the unknown function than the sinusoidal functions.

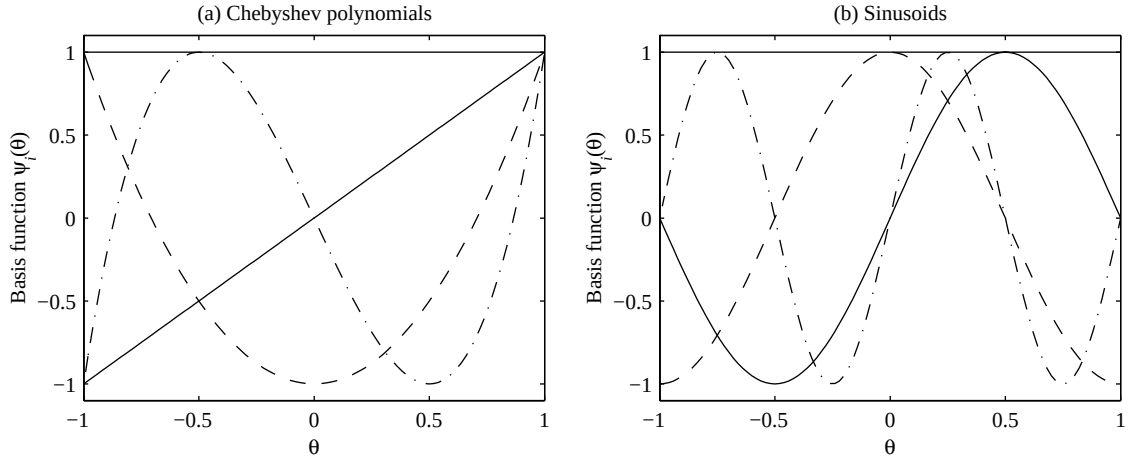

Figure 4.17 Sample shapefunctions: (a) polynomial; (b) sinusoidal.

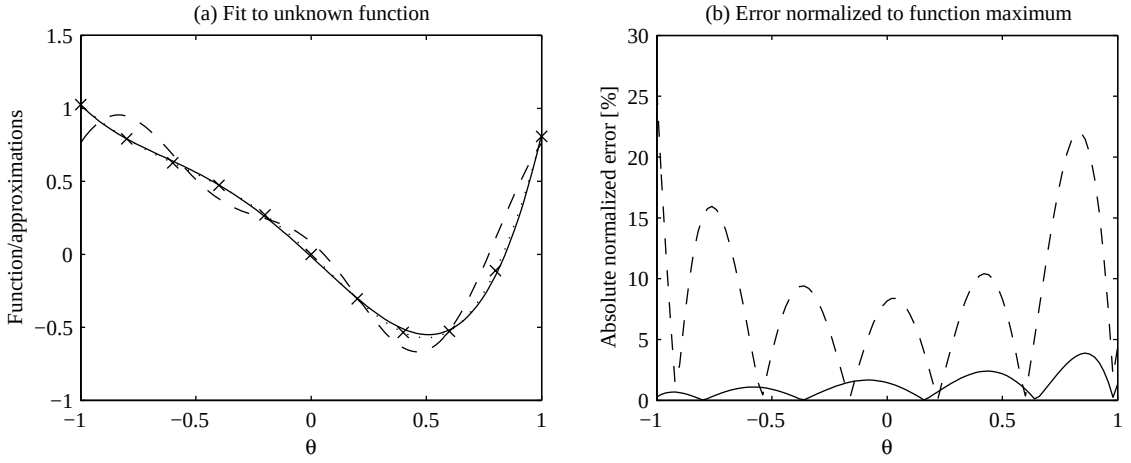

Figure 4.18 Sample fit for an unknown function and corresponding normalized error. The unknown function is shown as a dotted line. 11 reference or pre-computed points are used in the least squares fit; these are indicated as 'x' marks. A fit with 5 Chebyshev shapefunctions is shown as a solid line; a fit with 5 sinusoids is shown as a dashed line.

In practice the number of basis functions required to achieve a good fit varies greatly. This naturally depends on the TV system in question[1]. The element is first fit with a predetermined number of shapefunctions. The coefficients $m_{a,b}^{(i)}$ are then ordered with respect to magnitude, and normalized with respect to the largest coefficient. A threshold is then used to truncate the smaller coefficients to achieve a measure of compression.

Global shapefunction based fits are partially motivated by the fact that there may be underlying structure in the elements that are being fit. For example, since the nonlinearities of the geometrically nonlinear system enter through the sine and cosine functions (of the feedback rotation matrix), it is reasonable to expect that there may be some related pattern in the varying elements of the final control design. However, there are several mathematical steps that take place between the assembly of the feedback matrix and the eventual design of the controller that can obscure any underlying structure. For example, the LFT

1. For example, the main flexible modes of the MACE-II MBP are highly coupled to the gimbal angles; but the modes of the electronics support module are independent. This motivates the need for a interpolation scheme that can accommodate elements of varying complexity.

transformation involves an inverse of the cosine and sine functions. Another step that may obscure underlying structure is controller synthesis.

4.5 Nonlinear Analysis

The stability analysis of gain-scheduled systems remains an active field of research. [Shamma and Athans, 1990] derive a useful result for gain-scheduling on a reference trajectory: Given that the linearized systems are stable, robust performance and robust stability of the nonlinear system are inherited from the linearized systems, provided that the reference trajectory varies slowly. This result verifies the intuitive guideline of "scheduling on a slow variable", and is a central theme to the stability analysis of gain-scheduled systems, [Hoppensteadt, 1966 and Khalil and Kokotovic, 1991].

The slow variations of geometrically nonlinear systems suggest then that linearized analyses of the gain-scheduled controller are appropriate. The nonlinear analysis employed in this thesis utilizes the analysis tools developed in Chapter 3 for linear systems. The linear analyses for performance and robustness must be repeated over several setpoints.

Clearly the remaining question concerns the "density" of the analysis: how many setpoints must be analyzed in order to achieve some measure of confidence in closed-loop behavior? Two factors must be considered. First, sufficient variation of the open-loop system must be captured. The setpoints used should be dense enough to characterize all the nonlinear variations of the system.

If each linearized analysis requires relatively little computational effort, a uniform density of setpoints can be used to capture the variations in the nonlinear plant. A simple check was devised to determine whether the density was sufficient using the *smoothed* linearized models of the plant. At several locations in the setpoint space, linear interpolation was performed between two plant models at given setpoints, θ_{i-1} and θ_{i+1}, to provide an estimate of the plant model θ_i. The frequency response between this estimate and the actual frequency response of the plant model θ_i were computed and compared to ensure that

there were no visual differences. This provided confidence that the density of setpoints was sufficient to capture the nonlinear variations of the plant.

If the linearized analysis proves to be a computational burden, a varying density of setpoints can be used to represent the nonlinearity. The same criterion based on interpolating between plants and verifying the interpolated plant can be used iteratively to determine whether a particular region of the nonlinear space requires a higher density of setpoints for the nonlinear variations to be captured.

Second, analysis must be performed at "off-design" points, or setpoints which have not been used for the synthesis of the controller. If the performance and robustness characteristics of the closed-loop system at these setpoints do not deviate significantly from the "on-design" points, the implication is that the control design is reasonable.

The nonlinear analysis that is used in this gain scheduling framework thus involves a sampling of both the *uncertainty* and *nonlinearity* spaces. The computational effort of analyzing a gain scheduled controller becomes considerable, but careful application of tools such as the reduction method of Section 4.3 and diligent book-keeping can result in considerable time savings in the analysis process.

4.6 Integrated Design Process for Gain-Scheduled Controllers

The procedures required to convert a family of linear controllers into a gain-scheduled representation that can be implemented in hardware have been presented in the preceding sections of this chapter. These procedures are now combined with the linear design tools of Chapter 3 to produce an integrated design process for gain-scheduled controllers.

The gain-scheduled procedure is an iterative process and is summarized in Figure 4.19. The process begins with the uncertain nonlinear control design model of Chapter 2, which is the nonlinear model linearized at multiple points in both the uncertainty and nonlinear spaces. The first step is to reduce the state order of this control design model by modal

truncation. This reduction must be performed on a mode-sorted varying system, to ensure that the same mode is truncated in all the linearized systems (thus preserving the smoothness of the system).

The next step is to select a set of LQG design weights for the nominal system that satisfies the performance requirements. The linear LQG controllers are then synthesized and analyzed with the uncertainty samples. If the linearized systems do not meet both the robustness and performance requirements then SWLQG design must be iteratively performed to meet these requirements. The LQG and SWLQG design steps result in a set of control design parameters (LQG weighting matrices and sensitivity weights) that can be used at any setpoint in the operating space to generate a linear controller that meets the performance and robustness requirements. A family of linear controllers can now be designed at all setpoints in the evaluation space.

The state matrices of this family of controllers must now be smoothed with the mode sort and smoothing procedures developed earlier in this chapter. The reduction procedure is then applied to this smoothed family to produce a smoothed and reduced controller.

A sampling of the setpoints of the smoothed and reduced controller is then used to determine a compact implementation for the gain scheduled controller. This constitutes the interpolation step. The nonlinear analysis is then applied to the implementation of the gain-scheduled controller. If the desired performance and robustness requirements are not met, there are two options. The first is to implement an interpolation scheme that can capture the variations of the ideal controller more accurately. The other option is to identify the setpoints that have the worst performance or robustness and examine the linearized systems corresponding to these setpoints. SWLQG design is then focused at these setpoints and additional sensitivity weights can be added.

If the performance and robustness specifications are too stringent, SWLQG may not produce a feasible controller. The resulting controller design may be unstable, the closed loop bandwidth may exceed specifications, or the closed loop system destabilizes some-

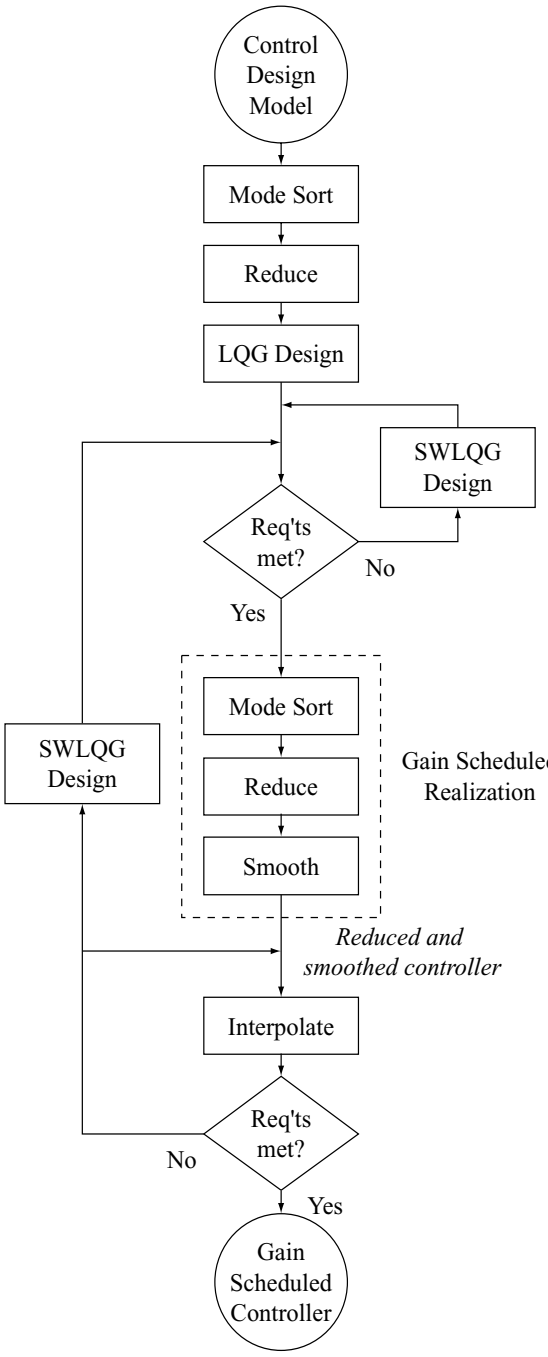

Figure 4.19 Integrated gain scheduled control design process, starting with the control design model from Chapter 2, and incorporating the linear design tools of Chapter 3.

where in the uncertainty space. If this happens, the iterative design process will not converge, and the performance and robustness specifications must be re-examined.

Some final notes on the gain-scheduled design framework presented in this thesis:

- Whereas in the linear problem trade-offs are generally made between performance and robustness, an additional factor must be considered for gain-scheduled design: A more detailed interpolation scheme will be able capture the variations of the ideal gain-scheduled controller more accurately. However, this must be traded off with greater real-time storage and computational requirements.
- The smoothing procedure assumes that the variations of the underlying ideal controller are smooth. Since the optimal controllers are model-based controllers, this in turn depends on the smoothness of the control design plant. The modeling methodology of Chapter 2 is able to produce such a smooth control design model because the models are generated with a smoothly varying feedback matrix. This would not have been the case if the control design models were the result of separate finite element analyses.
- Of related note is that the control design weights must also vary smoothly. For this thesis the design weights are held constant over the setpoint space, ensuring smoothness. However, each design iteration then involves analyzing the controller over the entire design space; the graphical design heuristics of Chapter 3 were used here to reduce the number of design iterations.
- The controller design process is performed entirely in the continuous frequency domain. This follows the MACE-I approach which used the Tustin transform to discretize the controller for real-time implementation. It should be noted that discrete setpoint controllers were designed, but there were problems with the interpolation. In particular, the frequency response of the interpolated discrete controllers was very sensitive to errors in the A matrix.
- The interpolation of continuous-time controllers has the drawback of requiring that discretization, necessary for hardware implementation, be performed in real-time. In order to alleviate this computational burden, simple algebraic equations that are equivalent to the Tustin transform for a two state system are derived. (see Appendix B). Note that this simplification is feasible only because the smoothing process creates a 2x2 block-diagonal A matrix.

4.7 Chapter Summary

A procedure for designing gain scheduled controllers for geometrically nonlinear systems has been developed. The uncertain, nonlinear control design model of Chapter 2 is used as a starting point, and the linear design tools of Chapter 3 are incorporated.

The procedure consists of three major steps. The first is the design of a family of linear controllers for linearized models of the nonlinear system. The designs are performed at several reference setpoints. The second step involves smoothing the state space matrices of this family of linear controllers. The final step is to implement an interpolation scheme that can interpolate between the reference setpoints so that controllers can be determined at an arbitrary setpoint. Although the design process does not directly achieve specified performance and robustness requirements, several options for iteration are incorporated in the design process to allow the control designer to gradually refine the controller to meet these requirements.

Chapter 5

APPLICATION TO MIT/MACE-II

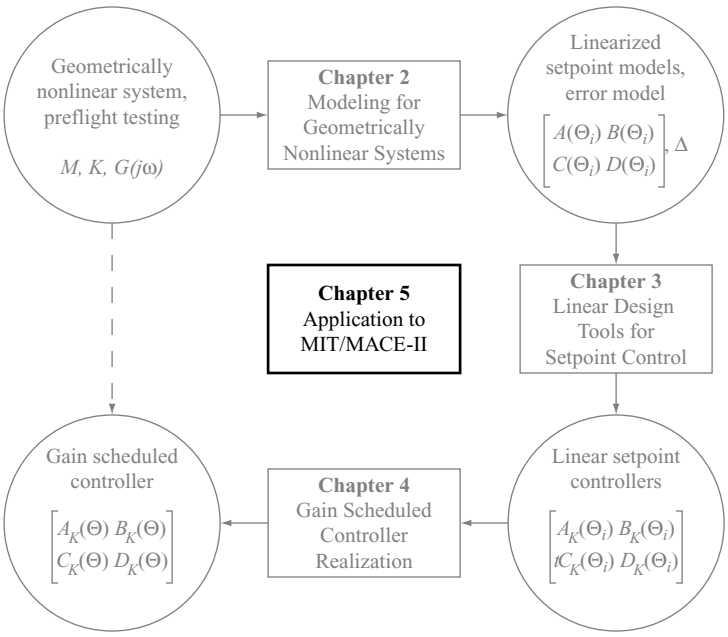

Figure 5.1 Thesis overview. The gain scheduling framework developed in this thesis is applied to the MIT/MACE-II experiment in this chapter.

A framework for the modeling and control of geometrically nonlinear flexible structures using gain scheduling has been presented in the preceding chapters. This framework is now applied to the modeling and control of the MIT Middeck Active Control Experiment reflight (MIT/MACE-II) flight experiment, which was flown on STS-106 for deployment on the International Space Station in September 2000. All modeling and control design tools are implemented using MATLAB[1].

TABLE 5.1 MIT/MACE-II Actuator and Sensor Abbreviations

Abbreviation	Description
PGX	Primary gimbal x-axis servo
PGZ	Primary gimal z-axis servo
SGX	Secondary gimbal x-axis servo
SGZ	Secondary gimbal z-axis servo
PRXI	Integrated primary gimbal x-axis rate gyro
PRZI	Integrated primary gimbal x-axis rate gyro

5.1 Experiment Description

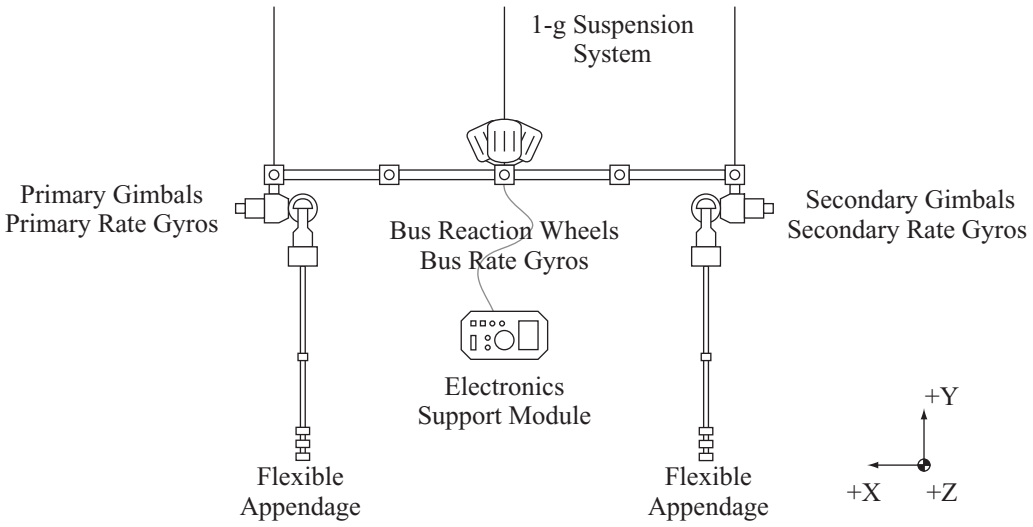

Figure 5.2 The MACE-II Multi-Body Platform (MBP), flexible appendages, Electronics Support Module (ESM) and 1-g suspension system

A diagram of the MIT/MACE-II hardware is shown in Figure 5.2. The experiment consists of four major components. The primary structure is the Multi-Body Platform (MBP), a one-meter truss constructed of Lexan tubes with aluminum nodes. A three axis reaction wheel package is located at the center of gravity. A three-axis rate gyro package is located at the reaction wheels to provide coarse attitude control capability. At each end of the MBP, a two-axis gimbal is mounted. The gimbals drive payload cans containing three-

1. © Mathworks Inc., Natick MA.

axis rate gyro packages. The gimbal angles are sensed using relative angle optical encoders, with a timing pulse to provide an absolute angular position measure.

The second component in the 0g dynamics comprise the flexible appendages. The flexible appendages are 0.7-meter long, ¾" diameter G10 fiberglass rods which are mounted to the outboard side of the payload cans.

The third component is the Electronics Support Module (ESM). This unit provides all of the real-time computing power, sensor conditioning and actuator amplification necessary to perform active control. The software has been designed to be flexible enough to accommodate many types of advanced control design techniques.

The final component of MACE-II is the suspension system, used for software and model development in 1-g. The suspension is a pneumatic system with active stiffness and mass cancellation. The suspension carriages attach to the reaction wheel frame and gimbal mount frames through 3.4 meter fiberglass rods which give pendulum modes between 0.2 and 0.7 Hz. These frequencies are below those of the flexible modes of the MBP.

Geometric Nonlinearity

The geometric nonlinearity of MIT/MACE-II arises when the gimbals move over large angles. A feedback loop between each gimbal motor and the collocated optical encoder creates servo loops through which the gimbal angles can be commanded. Figure 5.3 shows both Z-axis gimbals of the MBP being commanded at different angles. Each combination of four commanded angles comprise the setpoint, Θ = [PGX, PGZ, SGX, SGZ].

Because of the added moment arm of the attached flexible appendage, each gimbal is only capable of slewing a total of 10° in 1-g. The full geometric nonlinearity of the *in situ* 0-g environment cannot be achieved, providing additional challenges to the modeling effort.

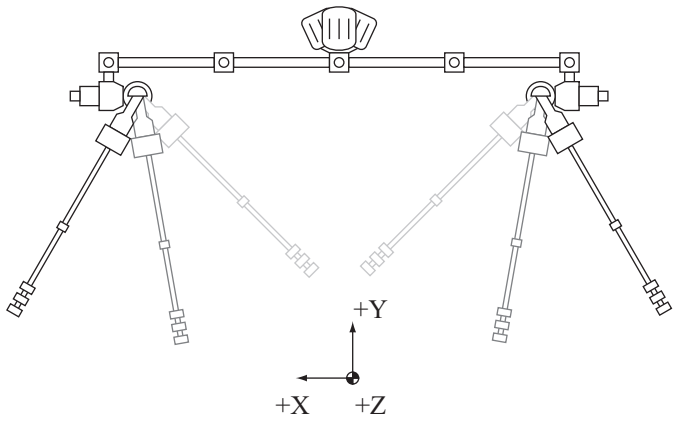

Figure 5.3 MBP with Z-axis gimbals at different commanded angles. The setpoint shown in light gray corresponds to $\Theta = [0,+45^o,0,+45^o]$; dark gray $\Theta = [0,-10^o,0,-10^o]$; and black, $\Theta = [0,-30^o,0,-30^o]$.

Experimental Frequency Responses

Experimental data is obtained by exciting each actuator with a bandlimited random signal, and recording time traces for the corresponding actuator and all sensors. The time data is converted to frequency responses using Welch's averaged periodogram method. Each frequency response has 1107 points from 0 to 120 Hz, and is obtained from 180 seconds of time data sampled at 500 Hz.

The software protocols used to obtain the experimental data are the essentially the same as the software protocols used for the implementation of closed-loop control, with the exception that the elements of the control matrices are set to zero. This ensures that factors such as time delays and servos are automatically included in the identified data. In this way the identified data for the control actuators and sensors accurately represents the control loop as seen by the software, facilitating both model updating and control design.

5.2 Geometrically Nonlinear Modeling

The nonlinear modeling framework of Chapter 2 is now applied to the MIT/MACE-II experiment. One important feature of this modeling methodology is that the extensive modeling effort of MACE-I can be incorporated. The modeling process for MIT/MACE-

II is summarized in Figure 5.4, illustrating how the prior modeling work of [Glaese, 1994] is incorporated into the MIT/MACE-II modeling effort.

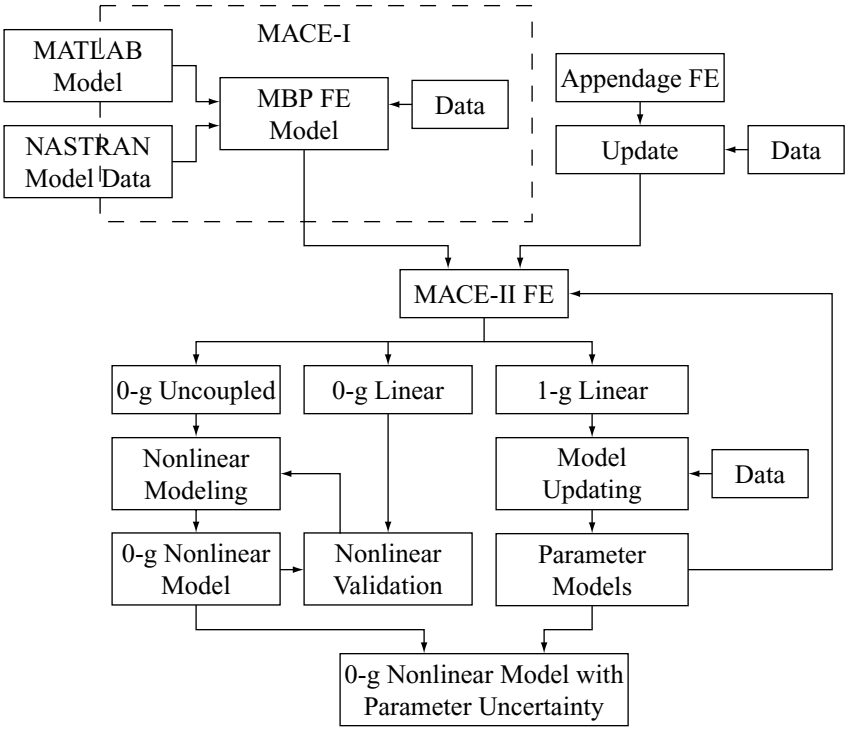

Figure 5.4 MIT/MACE-II modeling methodology, incorporating the prior modeling work of MACE-I. NASTRAN models had to be constructed for the flexible appendages. Updates also had to be applied to the original NASTRAN input decks and MATLAB code to reflect changes in the Multi-Body Platform and the Electronics Support Model.

5.2.1 Finite Element Model

The first step in the modeling process for MIT/MACE-II is to update the existing model that was developed for MACE-I [Glaese, 1994]. Several elements had to be added or modified:

- Descriptions of the flexible appendages had to be appended to the NASTRAN input deck. Each appendage was individually modeled and updated with data from ringdown tests.
- The sensor input and output equations were augmented with joint interface accelerations a and forces f.

- Computational delays had to be re-determined to reflect changes in the real-time software.

The resulting model is a state space description of the uncoupled MBP (bus, gimbal, and appendage) dynamics. The final model is shown in Figure 5.5 and consists of 168 nodes and 217 elements. The use of NASTRAN Multi-Point Constraints (MPC) greatly simplifies the modeling process as the same input deck can be used for 1-g analysis, 0-g uncoupled analysis, and 0-g linear coupled analysis with only minor modifications.

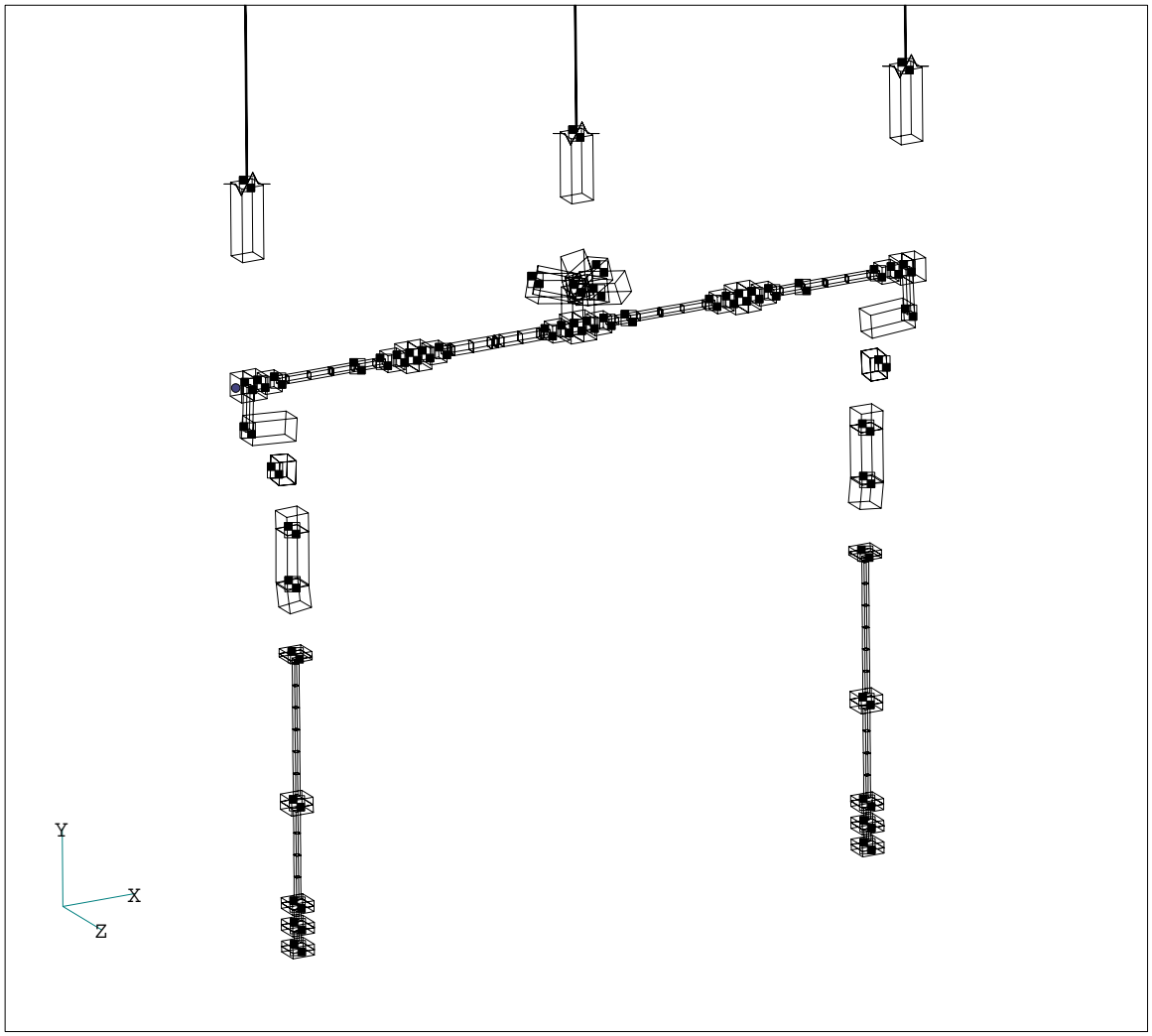

Figure 5.5 MIT/MACE-II finite element model, with uncoupled gimbal elements and suspension system. The FEM is implemented as a NASTRAN deck with 217 elements and 168 nodes. The gimbals can readily be attached using Multi-Point Constraints (MPC) to generate linear models to validate linearized version of the full nonlinear model.

5.2.2 Nonlinear Model

Geometric Nonlinearity

The geometric nonlinearity of MIT/MACE-II can now be examined. Figure 5.6 shows the variation in the magnitude of the eigenvalues as the configuration changes linearly from $\Theta = [-50^\circ,-40^\circ,-50^\circ,-40^\circ]$ to $\Theta = [50^\circ,40^\circ,50^\circ,40^\circ]$. Two different frequency ranges are shown: (a) 2 to 20 rad/s; and (b) 40 to 400 rad/s. Changes are largest for the gimbal modes between 2 and 4 rad/s.

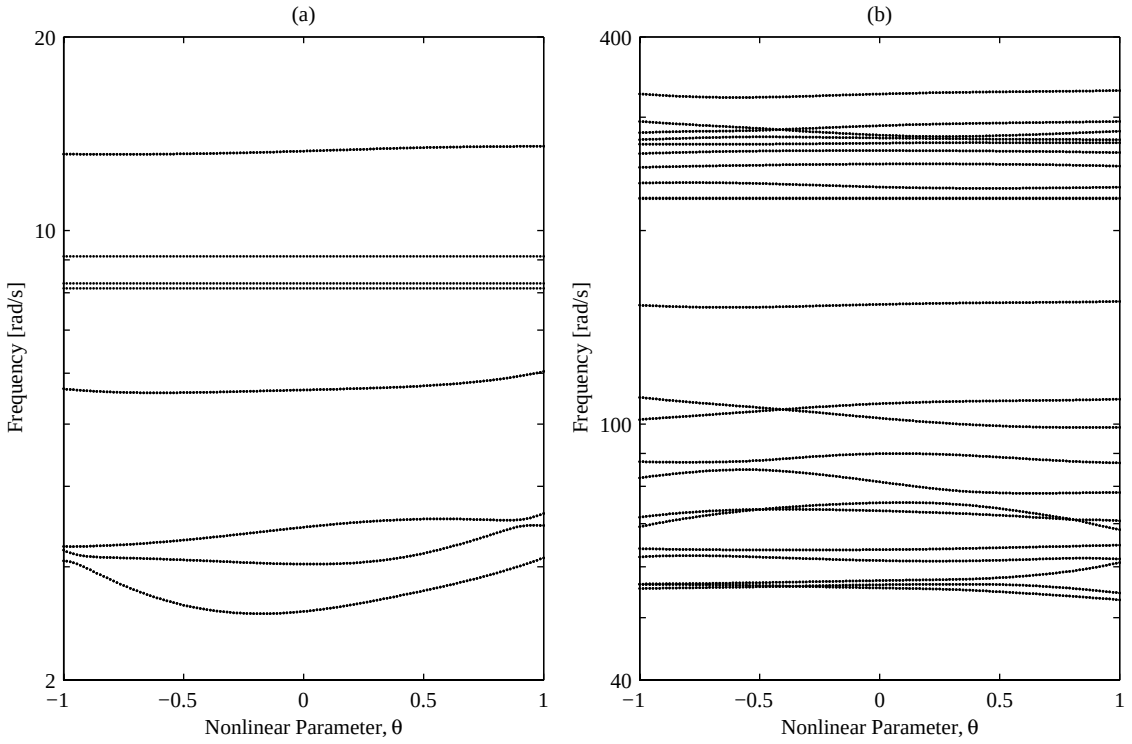

Figure 5.6 MIT/MACE-II geometric nonlinearity. The magnitudes of the eigenvalues are shown as the MBP changes configuration linearly from $\Theta = [-50^\circ,-40^\circ,-50^\circ,-40^\circ]$ to $\Theta = [50^\circ,40^\circ,50^\circ,40^\circ]$. This change is mapped to the nonlinear parameter θ varying from -1 to 1. Two different frequency ranges are shown: (a) 2 to 20 rad/s; and (b) 40 to 400 rad/s.

A more dramatic illustration of the nonlinearity is observed in the variations of the frequency responses. Frequency responses contain information about damping, zeros and residues as well as the magnitude of the poles. Figure 5.7 shows the nonlinearities present

in the geometrically nonlinear MIT/MACE-II model as variations in the frequency responses. As was illustrated in Figure 5.6, there are large changes in the frequencies of the gimbal modes between 2 and 4 rad/s. However at higher frequencies large changes in the residues of the modes are also visible.

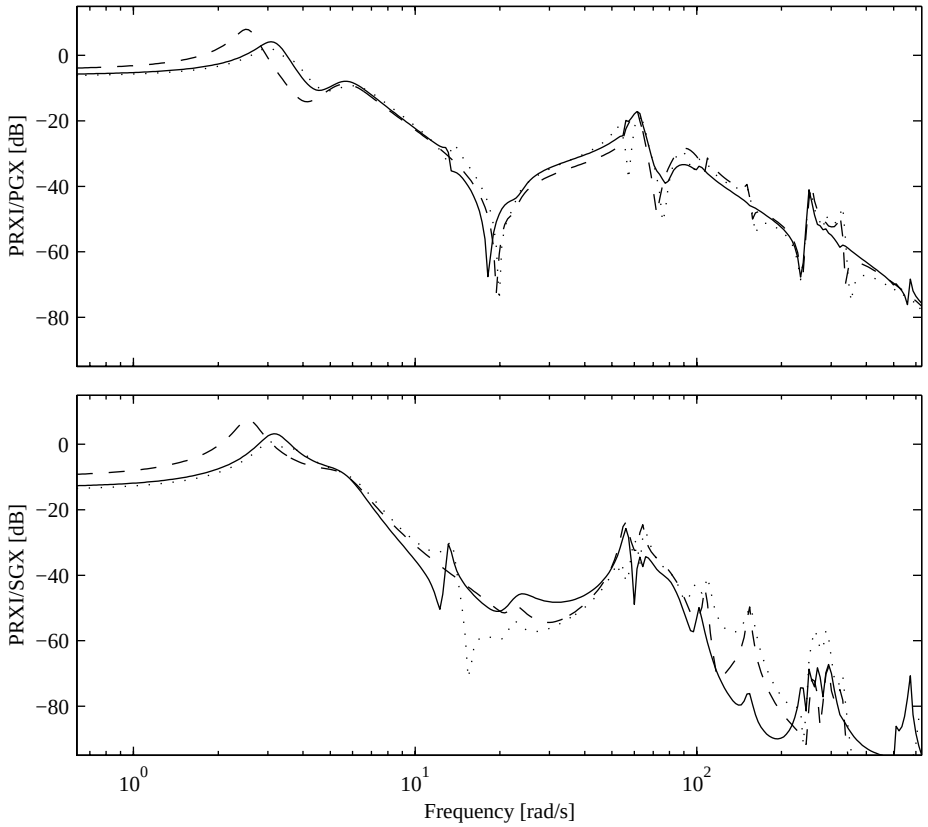

Figure 5.7 MIT/MACE-II geometric nonlinearity. Frequency responses are shown at three different configurations: $\Theta = [-50^\circ,-40^\circ,-50^\circ,-40^\circ]$ (solid); $\Theta = [0^\circ,0^\circ,0^\circ,0^\circ]$ (dotted); and $\Theta = [50^\circ,40^\circ,50^\circ,40^\circ]$ (dashed).

This change in residues can be explained by considering the structural configuration of MACE-II. The frequency responses of Figure 5.7 involve actuators and sensors of the gimbals in the X-axis. At the $\Theta = [0^\circ, 0^\circ, 0^\circ, 0^\circ]$ setpoint the Z-axis modes couple poorly into X-axis dynamics, since the gimbal axes are parallel to the bus axes and the alignment of the flexible appendages. At this setpoint, the gimbal rotation induced by Z-axis modes

is orthogonal to both the X-axis actuators and the sensors so that the residues of the Z-axis modes in the X-axis frequency responses are negligible. However, as the commanded gimbal axes rotate away from this configuration, the Z-axis modes begin to couple into the gimbal X-axis actuators and sensors and these modes become significant in the X-axis frequency responses.

Nonlinear Model Validation

The nonlinear modeling process is verified by comparing the nonlinear model, linearized at a particular set of joint angles, with a linear model assembled with the same set of joint angles. Figure 5.8 shows the singular values of the geometrically nonlinear model, linearized at the joint angles $\Theta = [0^o, 60^o, 0^o, 60^o]$, plotted against the singular values of the linear model. The two models are virtually indistinguishable.

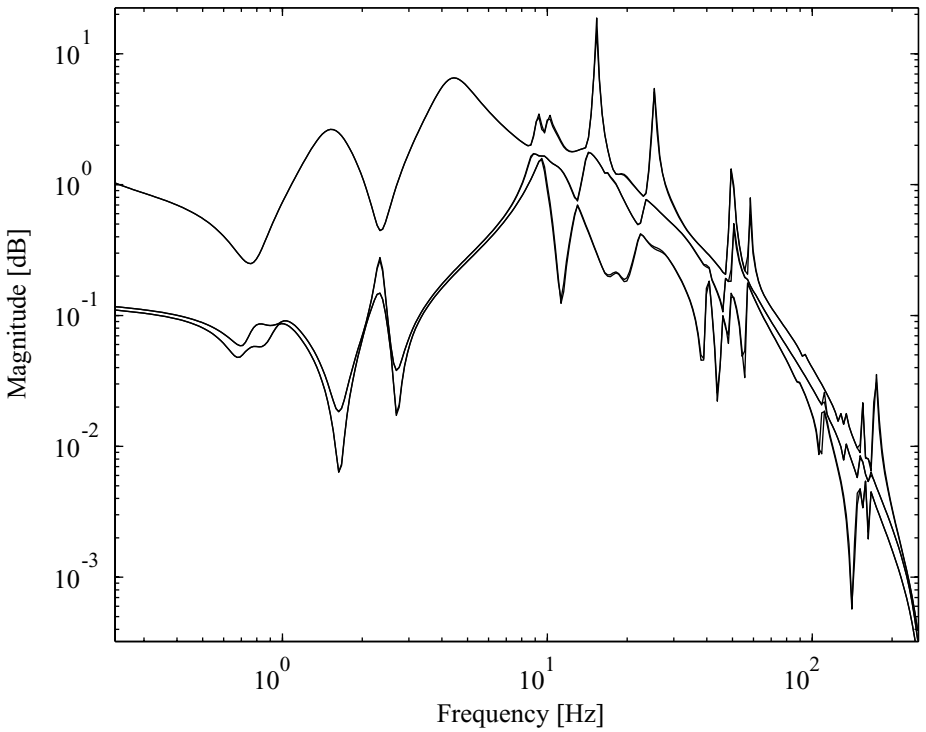

Figure 5.8 Singular values from reaction wheels to bus rate gyros, for the linear MBP FEM (solid) and the nonlinear FEM linearized (dashed) about $\Theta = [60,0,60,0]$. The two sets of singular values overlay indistinguishably. Note that there are three singular values for each system since there are three independent reaction wheels and rate gyros.

5.2.3 Model Updating

Parameter updating

Several model parameters were evaluated for their updating utility. These included: the moduli of the MBP Lexan struts, moduli of the G10 fiberglass of the flexible appendages, the mass properties of the aluminum appendage nodes, properties of the suspension cables, inertia properties of the gimbal assembly, and appendage attachment screw rotary stiffnesses.

The analysis showed that the moduli of the Lexan struts, the moduli of the G10 fiberglass of the appendages, and the appendage attachment screw rotary stiffnesses had the greatest influence on the frequency responses. Further physical insights suggested that the Lexan strut moduli should be linked (so that one modulus should describe all four struts), and that appendage attachment screw stiffnesses should be left independent (so that there are two individual stiffness per screw). Finally the G10 fiberglass modulus was not included in the update set since the effect of modifying this modulus was similar to modifying the stiffness of the appendage attachment screws.

Figure 5.9 shows the convergence of the updating cost with iteration for an optimization run with the parameter set chosen as above. The top two plots show that both the weighted and unweighted costs converge to a lower value. The weighted cost is the sum of the costs corresponding to the sensors, actuators, performance and disturbance inputs; these are the channels that are used for the nonlinear optimization. The unweighted cost combines all channels, including those channels that were not included in the nonlinear optimization. The convergence of both the weighted and unweighted costs indicates that the solution has converged to a physically meaningful set of parameters.

The final plot in Figure 5.9 compares the cost of the FE model at the start of a given iteration, to the cost of the updated model at the end of the same iteration. The convergence of this plot is a good indicator that the linearization approximations have succeeded in finding the minimum at each iteration. Figure 5.10 shows the corresponding convergence of

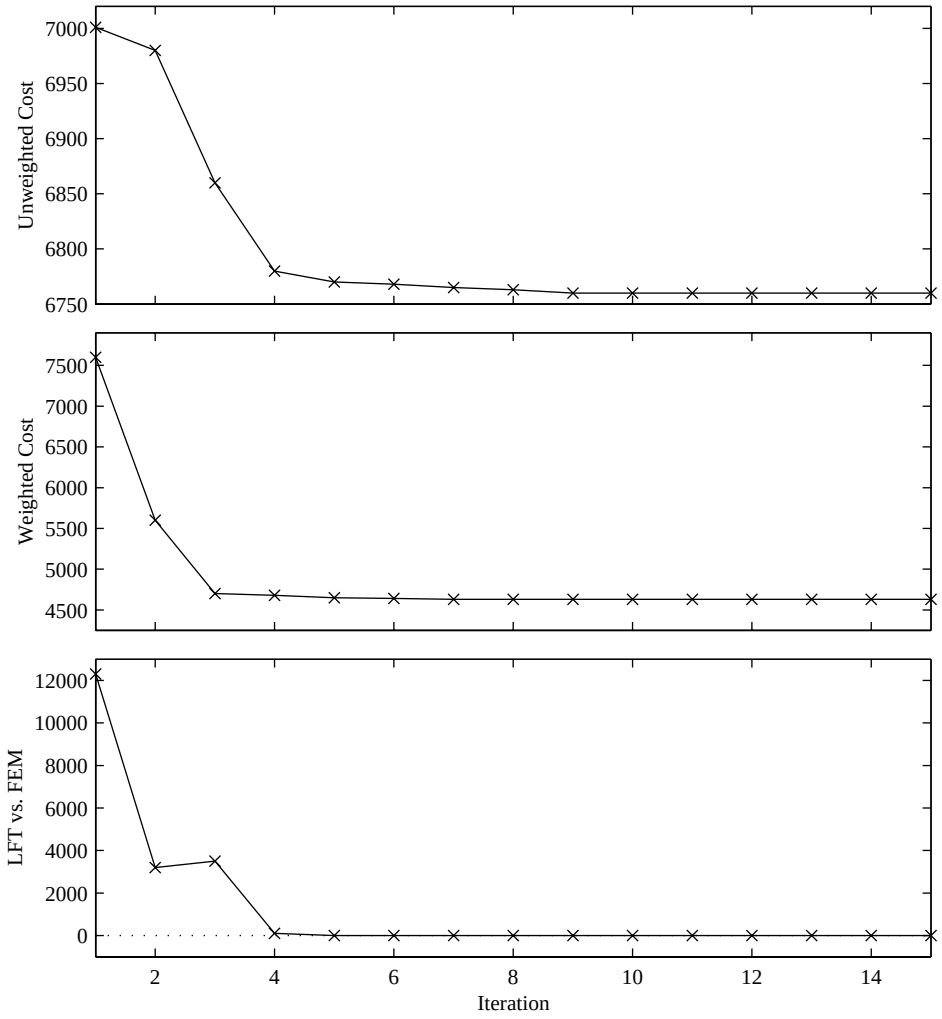

Figure 5.9 Cost convergence for typical parameter updating procedure. The top plot shows the unweighted costs (all actuators and sensors). The middle plot shows the weighted costs (actuators and sensors of the control problem only). The final plot compares the cost between the LFT updated model, and the model with the FE analysis performed with the updated parameters.

the individual parameters for the same optimization run, indicating that the parameter set is not overdetermined.

A sample frequency response is shown in Figure 5.11 comparing experimental data with an updated model. The fit is good at most frequencies, and qualitatively at least as close as measurement model fits.

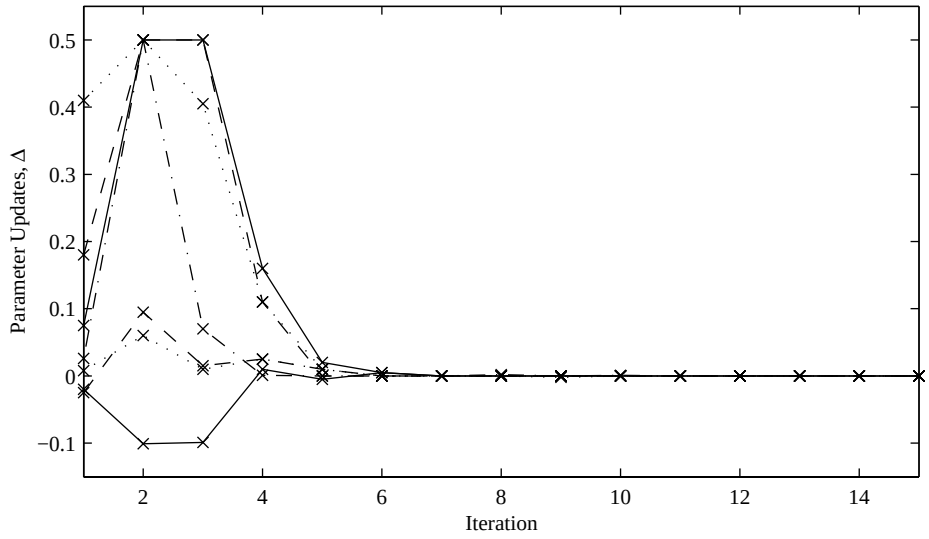

Figure 5.10 Update parameter convergence for typical parameter updating procedure.

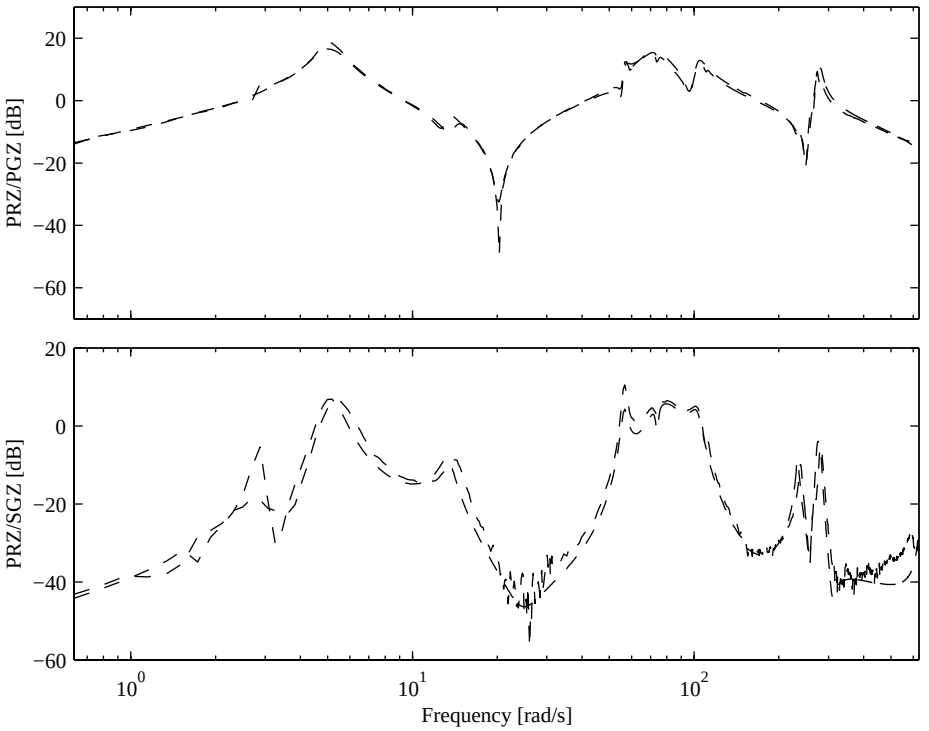

Figure 5.11 Sample frequency response for updated model compared with experimental data. The updated model is shown as a dotted line, and the experimental data is shown as a solid line.

Parameter Error Models

Four data sets were used to create statistics for the selected parameters. Each data set was used to perform separate updates. The update parameters are collected in Table 5.2 and statistics are computed to create the physical parameter model.

TABLE 5.2 Parameter Uncertainty Model

Parameter	Data Set				Mean	Var.	Dev.
	010301	010209	010109	001214			
Young's Modulus [N/m^2]							
Lexan ends	2.688e9	2.491e9	2.427e9	3.062e9	2.745e9	1.297e8	0.047
Lexan center	2.666e9	2.515e9	2.736e9	2.317e9	2.527e9	8.192e7	-0.032
Piezoceramic	3.188e9	4.074e9	3.055e9	5.013e9	4.034e9	4.058e8	0.101
G10	1.86e10	1.86e10	2.04e10	2.17e10	2.015e10	6.741e8	0.033
Appendage Attachment Screw Stiffness [Nm/rad]							
Pri. App. X	413.9	312.3	124.7	163.0	269.3	58.6	-0.218
Pri. App. Z	720.5	723.8	421.1	375.2	549.5	81.6	-0.148
Sec. App. X	745.6	567.0	382.3	292.1	518.9	88.0	-0.170
Sec. App. Z	827.5	851.1	530.7	412.8	632.0	95.0	-0.150

The parameter uncertainties are dominated by the uncertainty in the stiffness of the appendage attachment screws. Further investigation of the appendage attachment mechanisms revealed a hardware design issue which allowed the appendages to be attached at a slightly different configuration for successive assembly/dis-assembly cycles. This issue was not known before the model updating process, and was only discovered as a result of the model updating process.

The appendage attachment screw stiffness uncertainties do not represent errors in the model. Rather they represent of a true uncertainties in the system, since the attachment mechanism does not consistently connect the appendage to the payload can at each assembly. This is in fact a powerful demonstration of the ability of the model updating procedure presented in this work to capture physical parameter errors.

5.3 1-g Controller Design

A linear controller design for the MIT/MACE-II experiment is presented, demonstrating some of the synthesis and analysis tools presented in Chapter 3.

5.3.1 Control Design Objective

The control topology is summarized Figure 5.12. There is a disturbance at the secondary gimbal (SGX), and control at the primary gimbal (PGX). The performance and sensor are collocated at the primary rate gyro (PRXI), integrated to obtain a displacement signal.

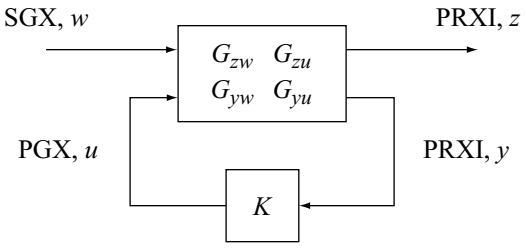

Figure 5.12 MIT/MACE-II control problem for demonstration of linear design tools.

The MACE control problem is then to reject disturbances from SGX to PRXI using the control loop consisting of PGX and PRXI. This disturbance rejection requirement is bandlimited between 0.5 and 50 Hz (or approximately 3 rad/s to 300 rad/s). In practice this objective is usually specified in terms of a decibel improvement. However the objective here is to determine the controller that provides the maximum performance while maintaining a reasonable level of robustness, and in doing so determine the factors that limit the amount of control that can be applied. This in turn can be used as a guideline for 0-g design.

Two sets of experimental frequency responses are used for control design; one for evaluation and one for design. The evaluation data is taken as the first frequency responses. A 53 state design model is identified from the second set of frequency responses using DynaMod[1], a MATLAB-based system identification tool. There are significant physical

differences between the two experimental data sets; the hardware was completely disassembled in-between the system identifications.

The evaluation data is compared to the design model in Figure 5.13. The zero in the PRXI/PGX loop at 20 rad/s is noticeably different between the two models. This is also manifested as a shift of the modes at 60 rad/s. There are also significant differences in the modes at 250 rad/s. These differences are primarily a result of the design of the appendage attachment screws (see the model updating procedure in Section 5.2.3 for more discussion), which strongly couple to the modes of the flexible appendages.

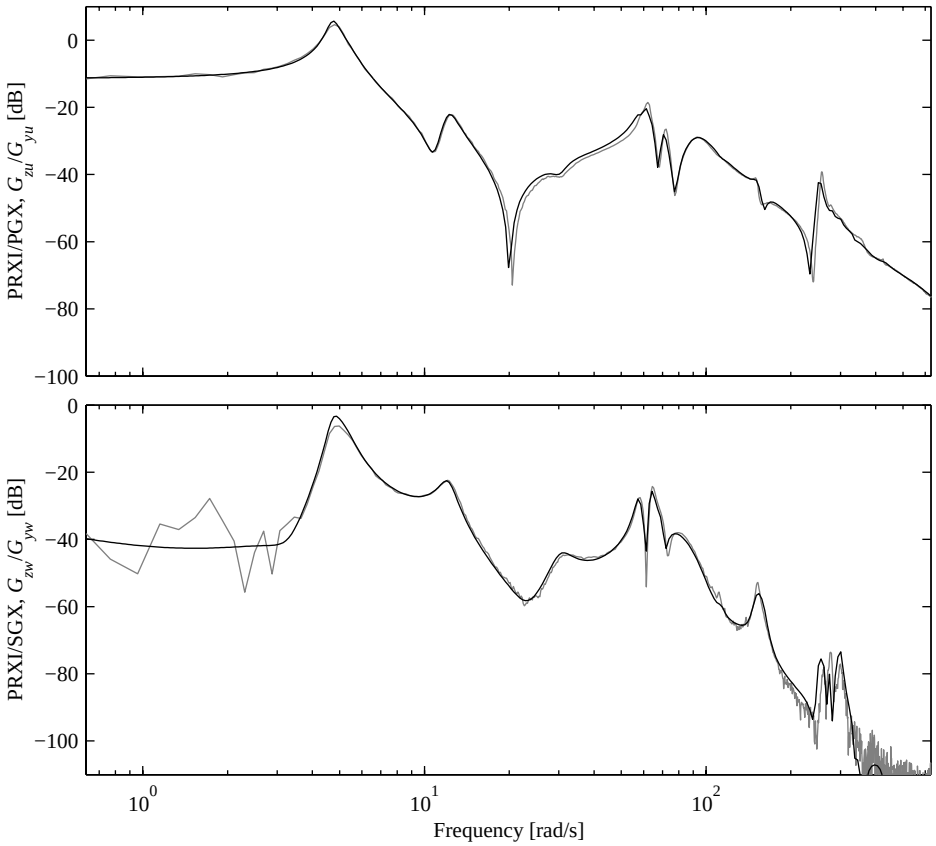

Figure 5.13 Control design model (black) and evaluation (gray) data for 1-g linear control.

1. © Midé Technology Corporation, Medford, MA.

5.3.2 Controller Design

As described in Chapter 3, the design of linear controllers is an iterative process. The synthesis and analysis procedure for two iterations are described in this section, first for an LQG design and then an SWLQG design. The LQG design does not prove to be stable so sensitivity weights are incorporated to produce a robust SWLQG controller.

Weights are selected for LQG design, and these are shown in Figure 5.14. The choice of the weights includes a frequency dependent de-weighting at low frequency, since the MACE performance specification does require the rejection of disturbances below 3 rad/s. The relatively low magnitude of the control cost, $G_{zu}^{(u)}$, compared to the state penalty, $G_{zu}^{(z)}$, suggests that the closed loop performance can be predicted by comparing the disturbance to sensor frequency response, $G_{yw}^{(w)}$, with the sensor noise, $G_{yw}^{(u)}$.

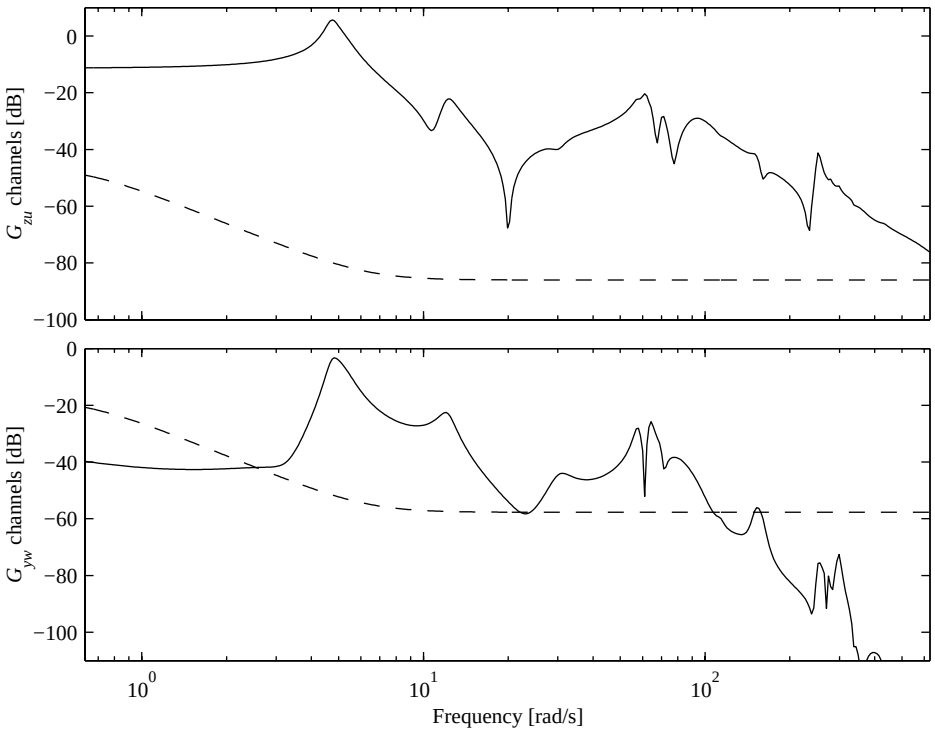

Figure 5.14 Control design weights for 1-g LQG linear control design. The top plot shows the state penalty (solid) and the control cost (dashed), and the bottom plot shows the disturbance to sensor frequency response (solid) and the sensor noise (dashed).

The LQG controller is then synthesized and the closed loop system is analyzed for both the design model and evaluation data. This analysis is presented in the three plots of Figure 5.15; the Nichols chart, maximum singular values of the sensitivity transfer function, and the performance frequency response. The plot of closed loop performance corresponds well with the prediction of the graphical heuristic; the closed loop performance roughly follows the minimum of the $G_{yw}^{(w)}$ and $G_{yw}^{(u)}$ frequency responses.

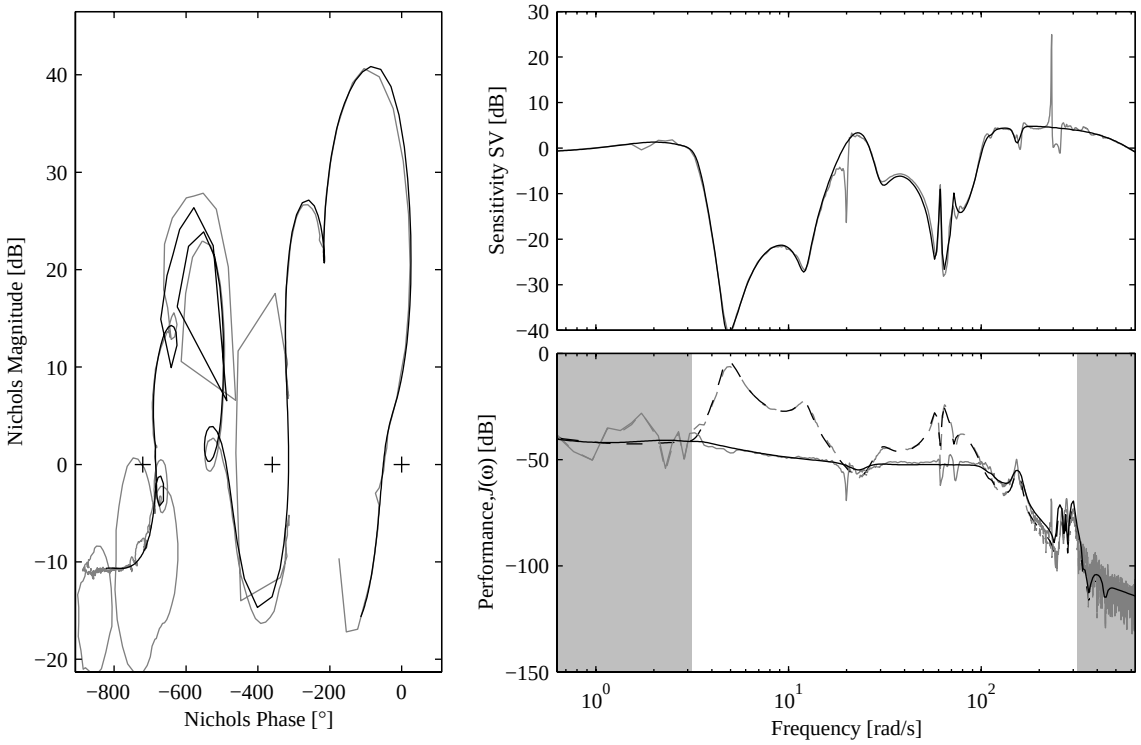

Figure 5.15 Analysis plots for 1-g LQG linear controller. The three plots show the controller evaluated on the control design model (black) and the evaluation data (gray). For the performance, the open loop response is also shown (dashed lines). The shaded area of the performance plot indicates regions that are not included in the performance calculations.

Although the choice of these weights produces good performance (24 dB reduction) on the evaluation data, and acceptable bandwidth (220 rad/s), the Nichols chart and the maximum singular values of the sensitivity transfer function indicate two stability problems. The first problem is evident from the Nichols chart. There is a pass-over of the critical point at –360º, indicating that the actual closed-loop system will be unstable at that fre-

quency. This pass-over occurs at approximately 20 rad/s. The second problem involves the 25 dB spike at 230 rad/s of the maximum singular values of the sensitivity transfer function. This indicates that there will be robustness issues that frequency. This is also reflected in the Nichols chart as a close pass-by of the critical point at –720º.

Note that the sensitivity plot does not indicate stability problems at 20 rad/s, even though the Nichols chart suggests that the system will be unstable. This a good example of control analysis where the sensitivity singular value based analysis does not provide a clear picture of the stability robustness of the closed loop, and a reminder that both the sensitivity transfer function and the Nichols chart should be examined in the control analysis.

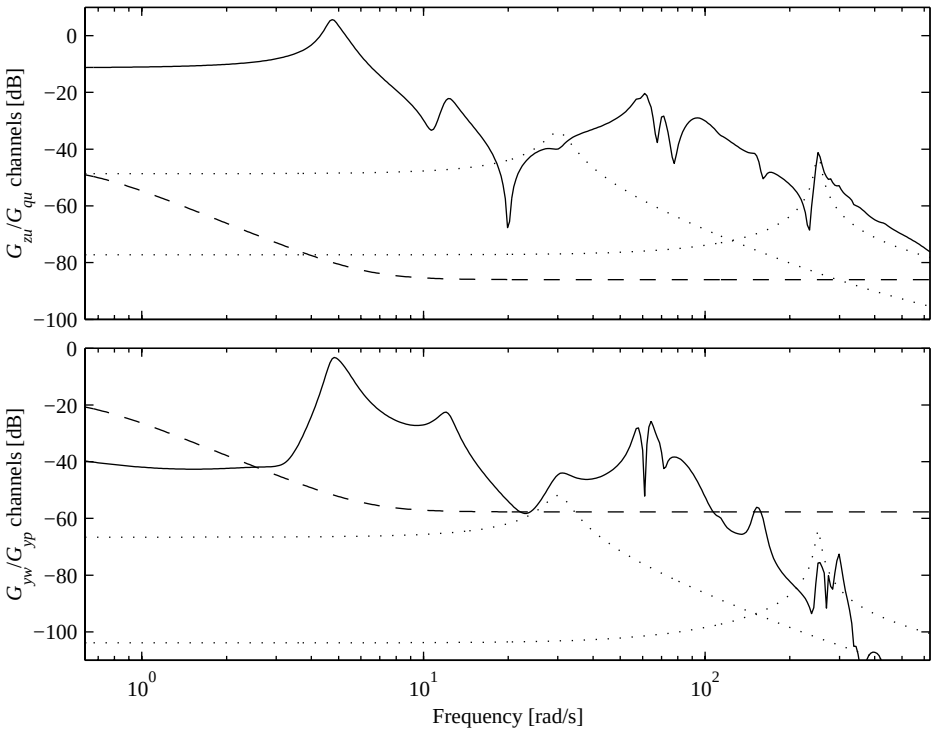

Figure 5.16 Control design weights for 1-g SWLQG linear control design. The top plot shows the state penalty (solid), the control cost (dashed), and the sensitivity weights (dotted). The bottom plot shows the disturbance to sensor frequency response (solid) and the sensor noise (dashed), and the sensitivity weights (dotted).

Sensitivity weights must now be incorporated to address the stability issues of the LQG controller. After several design iterations, a satisfactory group of sensitivity weights are selected. The control design weights, consisting of both the LQG and SWLQG weights are shown in Figure 5.16. Two sets of sensitivity weights had to be added, one for each of the stability problems of the LQG design (at 20 rad/s and then at 230 rad/s).

The SWLQG controller is then synthesized and the closed loop system is analyzed for both the design model and evaluation data. This analysis is presented in the three plots of Figure 5.17; the Nichols chart, maximum singular values of the sensitivity transfer function, and the performance frequency response. Satisfactory performance (~22 dB) and bandwidth (~110 rad/s) are achieved. The addition of the sensitivity weights has reduced the performance by approximately 2 dB when compared to the original LQG design. With the same LQG weight structure, this is to be expected for SWLQG, since the sensitivity weights are effectively penalizing the cost in a different "direction" than the LQG weights.

The Nichols chart and the maximum singular values of the sensitivity transfer function indicate that stability problems of the LQG controller have been removed with the addition of the sensitivity weights. The Nichols chart shows no pass-overs of the critical points, and the maximum singular values of the sensitivity transfer function are all below 10 dB, indicating that the SWLQG controller is robustly stable.

5.3.3 Experimental Validation

The SWLQG controller presented in the previous section was implemented on MIT/MACE-II in 1-g and found to be stable. The closed loop frequency responses are compared with the open loop frequency response and the predicted closed loop response in Figure 5.18. The analysis tools are able to predict the closed loop response accurately. The integrated performance is predicted to be 22.1 dB, and the actual performance is experimentally determined to be 21.9 dB.

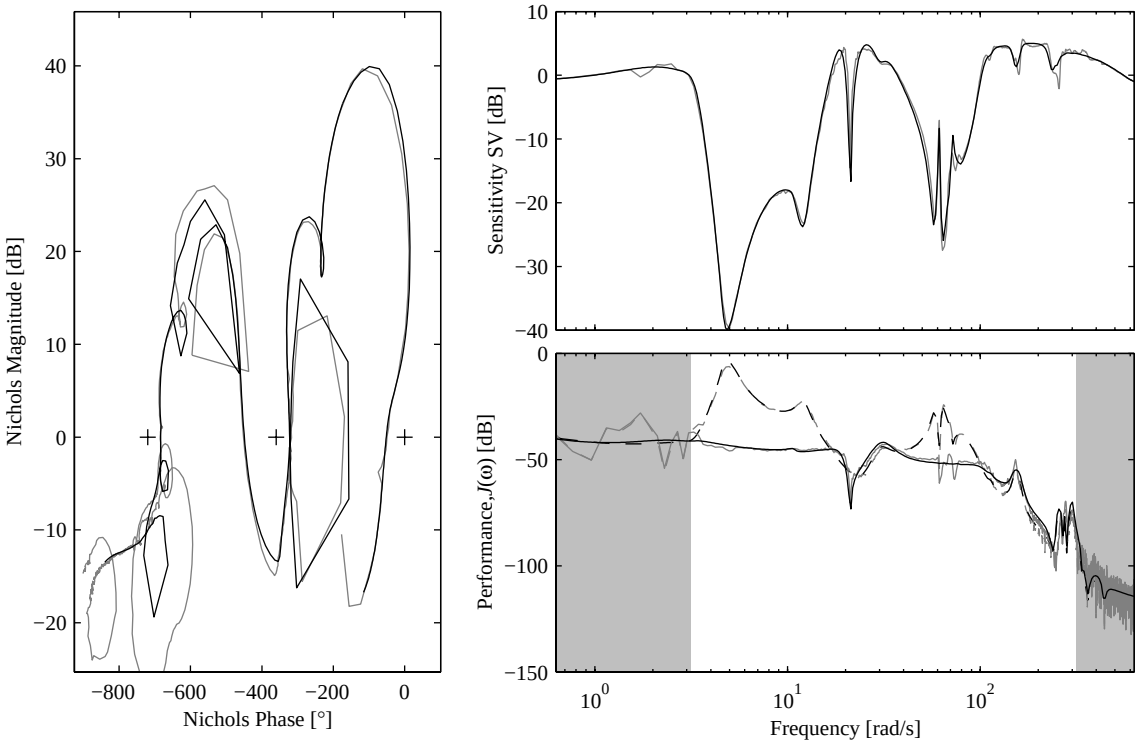

Figure 5.17 Analysis plots for 1-g SWLQG linear controller. The three plots show the controller evaluated on the control design model (black) and the evaluation data (gray). For the performance plot, the open loop response is also shown (dashed lines). The shaded area of the performance plot indicates regions that are not included in the performance calculations.

As mentioned earlier, one of the control objectives is to determine the factors that limit the amount of control that can be applied. For LQG, this factor proved to be the mismatch of the design model and evaluation data. LQG controllers exhibit a tendency to cancel out zeros [Campbell, 1993]; however the mismatch in the zero at 20 rad/s of the PRXI/PGX frequency response (Figure 5.13) limits the ability of LQG to cancel this zero out. The sensitivity to modeling errors is a fundamental limitation of LQG, since it does not account for model errors in the solution procedure.

As for the SWLQG controller, further examination of the Nichols chart of Figure 5.17 indicate that the sensitivity weights effectively mask the differences between the evaluation data and control design model. This is evidenced by the observation that the pass-

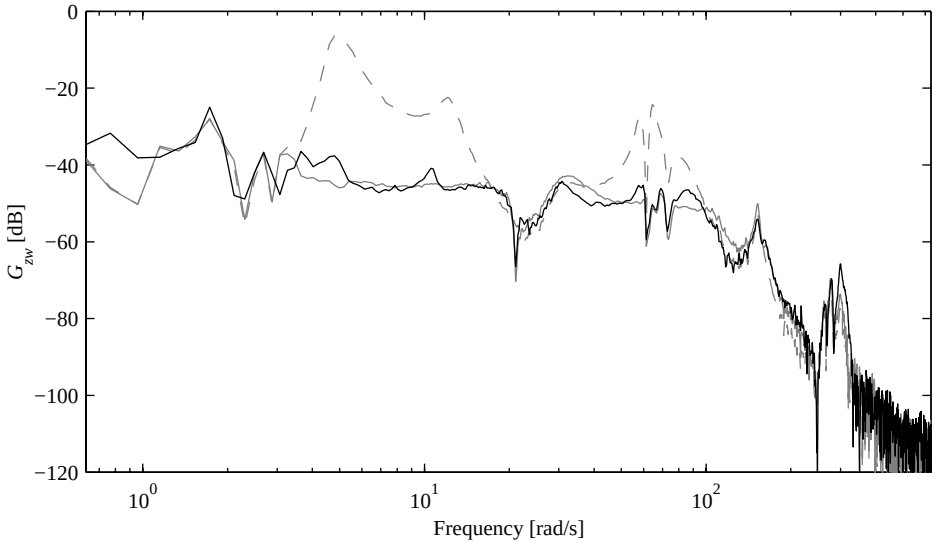

Figure 5.18 Predicted performance (grey) of 22.1 dB compared with actual performance (black) of 21.9 dB. Open loop (servos only) data is shown as a dashed line.

overs of the critical points are no closer for the evaluation data than for the control design model. The limitation to control is not due to modeling errors.

During the design iterations, attempts to increase the control authority tended to destabilize the SWLQG controller (while preserving the stability of the closed loop). As discussed in Section 4.1.2, unstable controllers are generally undesirable; the requirement for *stable* controllers turned out to be the limitation to 1-g control.

5.4 0-g Controller Design

The gain scheduling framework of this thesis is now applied to the 0-g nonlinear control of MIT/MACE-II. Several different options for the implementation of a gain scheduled controller, as described in Chapter 4, are compared. Comparison is also made to a benchmark linear controller designed using the Multiple Model method (Section 3.2.3).

5.4.1 Control Design Objective

As with the linear control design process, the first step is to define a control objective. The control topology used for the nonlinear problem will be the same as for the linear control problem example of the Section 5.3.1, with a disturbance at the secondary gimbal (SGX), control at the primary gimbal (PGX), and performance and sensor collocated at the integrated primary rate gyro (PRXI).

The nonlinear operating space selected as part of the control objective is designed to explore the interesting nonlinearities of the MIT/MACE-II experiment. To simplify the presentation of results, the multi-dimensional nonlinear space is mapped to a normalized setpoint. This mapping is shown Figure 5.19; the X-axis gimbals vary linearly from –50º to +50º, and the Z-axis gimbals vary from –40º to +40º. The limits on these variations are chosen to reflect physical constraints of the hardware.

Representative linearized setpoint models for the for the control design example presented in this section are shown in Figure 5.7. As previously discussed in Section 5.2.2, the nonlinearity of the plant is reflected in eigenvalues changes and variations in modal residues. The control design model used for the nonlinear control design in the following sections consists of 145 linearized models evaluated at evenly distributed setpoints, each with 100 states. Each setpoint consists of 3 samples that are distributed through the uncertainty space.

5.4.2 Gain Scheduled Controller Designs

The gain scheduled control design process described in Chapter 4 is summarized here. The first step is to determine a set of control design weights that produce the desired linear performance over the entire nonlinear and uncertainty space, or the analysis space. A family of linear controllers is synthesized at a set of design setpoints with these control design weights. The state matrices of this family of controllers are then smoothed and reduced. A sampling of this smoothed/reduced controller is then incorporated into an

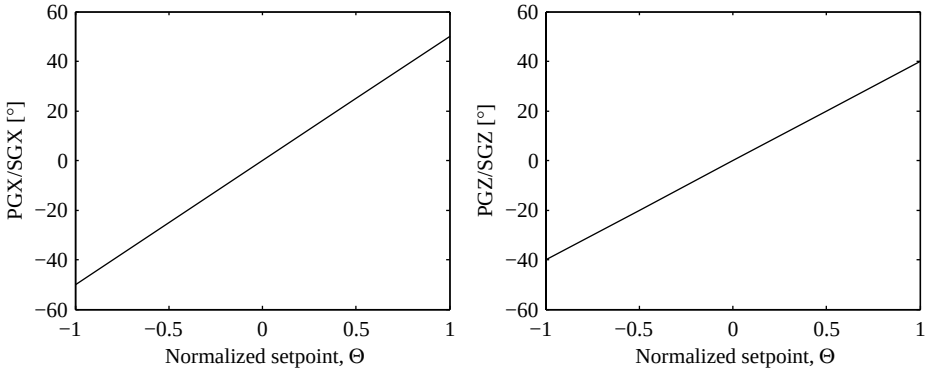

Figure 5.19 Transformation of one-dimensional setpoint space to normalized setpoint coordinates.

interpolative scheme to produce the final gain scheduled controller that is valid at all setpoints.

Controller Smoothing and Reduction

The smoothing and reduction procedures are validated in this section. An initial family of controllers is designed using SWLQG techniques, using the same control design weights across the all setpoints to ensure smoothness. SWLQG controllers are then computed for the 145 setpoints which the nonlinear control design model has been sampled. Each controller sample is a 104 state linear system (the linearized control design model has 100 states, and 4 frequency weighting states are added as well).

The state matrices of the SWLQG controllers are smoothed and reduced to 46 states. Figure 5.20 compares the full order to the smoothed/reduced controllers at selected setpoints; there are only very small differences. This indicates that, for these selected setpoints, the smoothing and reduction process has been successful in significantly reducing state order while preserving most of the input-output behavior.

The nonlinear analysis is performed for both the initial controllers and smoothed/reduced controllers at all 145 setpoints. Note that the analysis of a single setpoint involves the analyses at different point in the uncertainty space; for this controller design there are 3

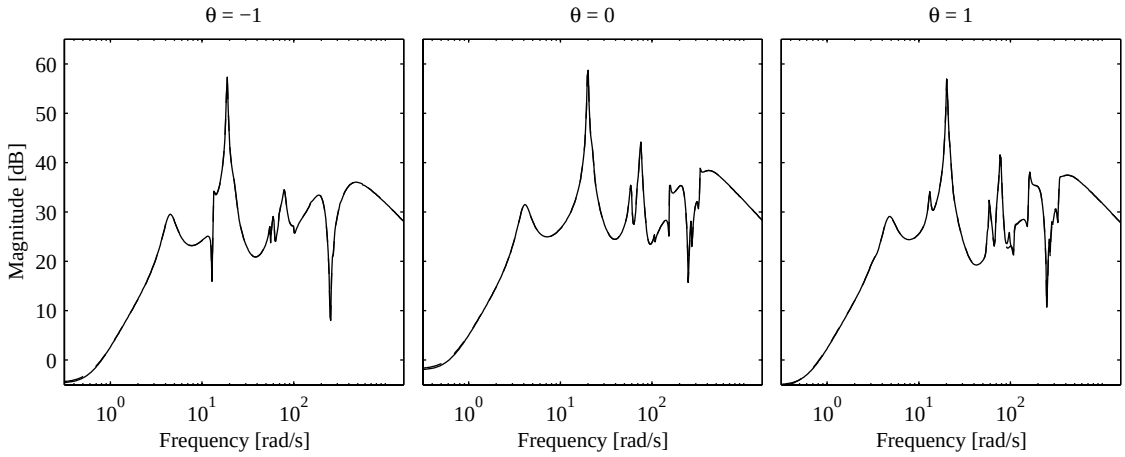

Figure 5.20 Full order and reduced order controller designs at different setpoints. The 104 state full order controller is shown as a solid line, and the 46 state smoothed/reduced controller is shown as a dashed line.

samples of the uncertainty space, so a total of 435 linear analyses are performed for each controller.

The analysis results for both controllers is summarized in Figure 5.21. The top row of plots shows the robustness of the controller, and the bottom row of plots shows the performance. A suitable measure of robustness is given by the largest singular value (over frequency) of the sensitivity transfer function. The performance of the controller is the bandlimited disturbance rejection (in dB with respect to open loop performance). The horizontal axes in each plot is the normalized setpoint, and each of the three columns corresponds to a different sample of the uncertainty space. The following conclusions are drawn from these plots:

- Both sensitivity and performance vary with configuration. This is because LQG/SWLQG performance cannot be explicitly specified.
- Sensitivity singular values are all below 10 dB over the entire design space. This indicates that both controllers have good robustness properties.
- The robustness and performance of the two controllers are very similar. This indicates that the reduction has not compromised the structure of the ideal controller.

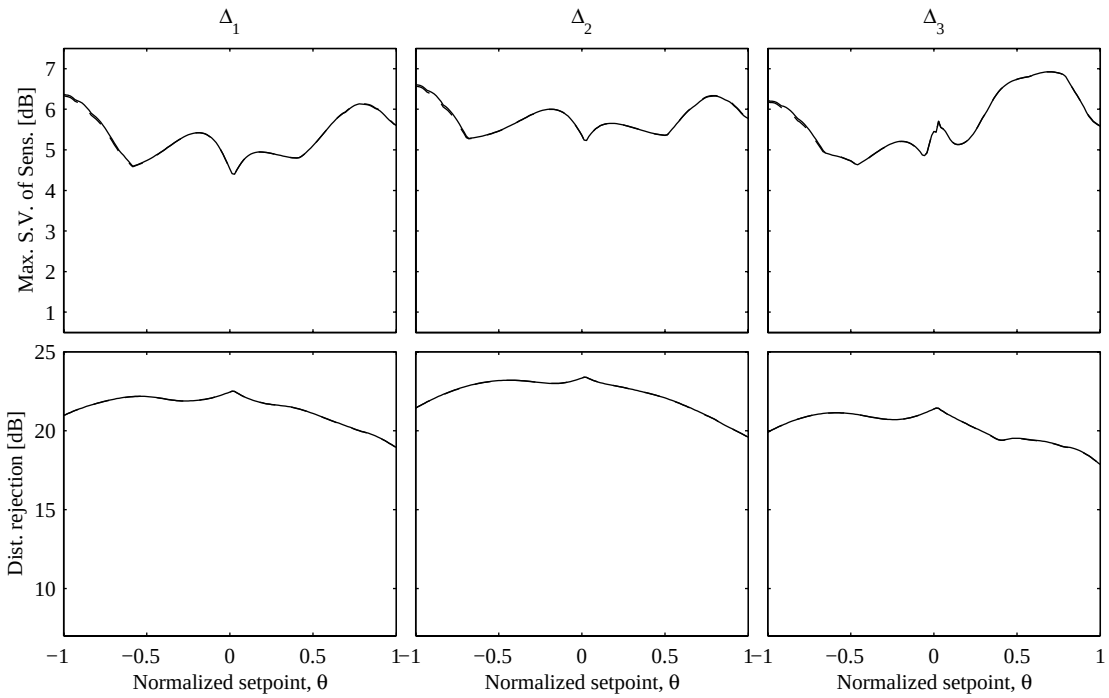

Figure 5.21 Robustness and performance of the full order and reduced order gain scheduled controllers. Each column of plots correspond to analysis at different samples of the uncertainty space Δ_1, Δ_2 and Δ_3. The ideal controller is shown as a solid line, and the smoothed and reduced controller is shown as a dashed line. The robustness and performance of the two controllers are virtually indistinguishable.

Controller size

The reduced controller has 46 states, 1 output, and 1 input so that there are potentially $(46+1)^2 = 2209$ varying elements in the [A, B, C, D] matrices of the state-space representation. One of the benefits of the smoothing process is to create a block diagonal structure that reduces the number of non-zero elements to a much more manageable 178 elements. Furthermore, many of these elements are duplicated (a property of the real modal realization) or non-varying (from the direct realization of a mixed mode) so that there are a total 156 unique varying elements.

Interpolation schemes

These 156 varying elements are the focus of the interpolation schemes. The objective is to capture the element variation with a relatively small set of numbers. Three different

schemes are examined in this section: nearest neighbor, linear and shapefunction, and are described in detail in Section 4.4.

Nearest Neighbor Interpolation

The nonlinear analysis of a gain scheduled controller implementation using a nearest neighbor interpolation scheme using 13 reference setpoints is presented in Figure 5.22. The analysis shows that both the performance and the robustness measures do not deviate from the ideal controller by more than 1 dB.

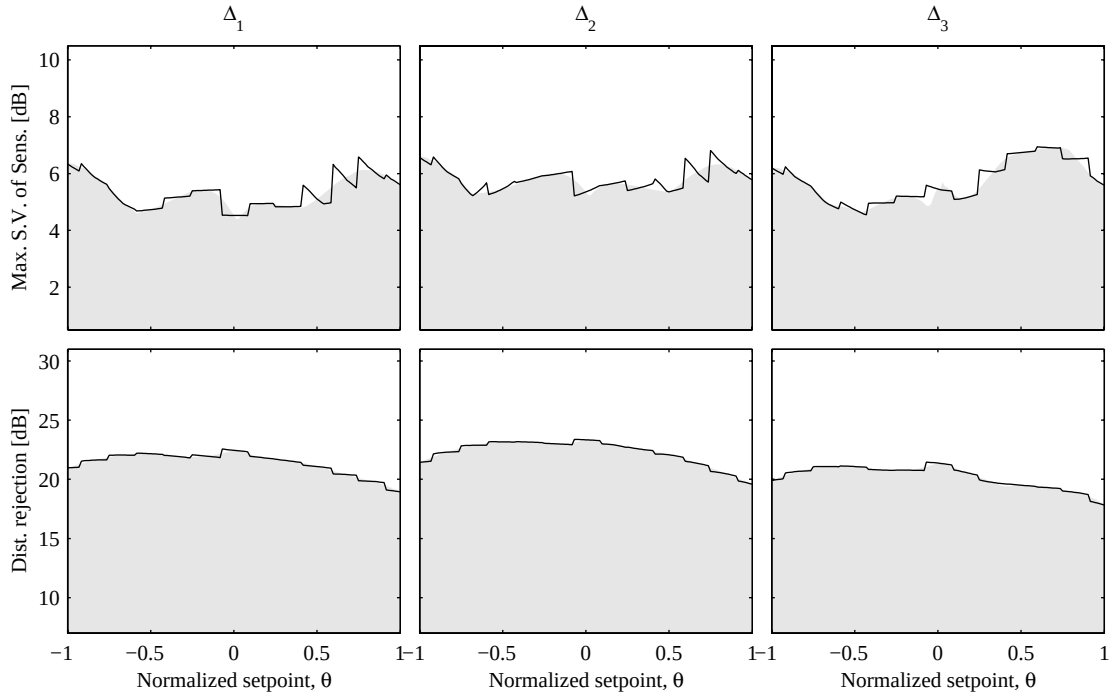

Figure 5.22 Robustness and performance of the nearest neighbor interpolation scheme (solid black line). For comparison the robustness and performance of the ideal controller is shown as a shaded gray patch.

However there are several discontinuities in all the robustness and performance plots. These discontinuities correspond to the switching that occurs as different reference setpoints become “nearest”, and can manifest themselves as switching transients under slewing maneuvers.

Linear Interpolation

A direct improvement on the nearest neighbor interpolation scheme is the linear interpolation scheme. A linear interpolation is performed with the surrounding reference setpoints. The nonlinear analysis of a gain scheduled controller implementation with a linear interpolation using 13 reference setpoints is presented in Figure 5.23. The analysis shows that both the performance and the robustness measures do not deviate from the ideal controller by more than 0.5 dB, which is an improvement over the nearest neighbor interpolation scheme. Furthermore the robustness and performance measures vary continuously.

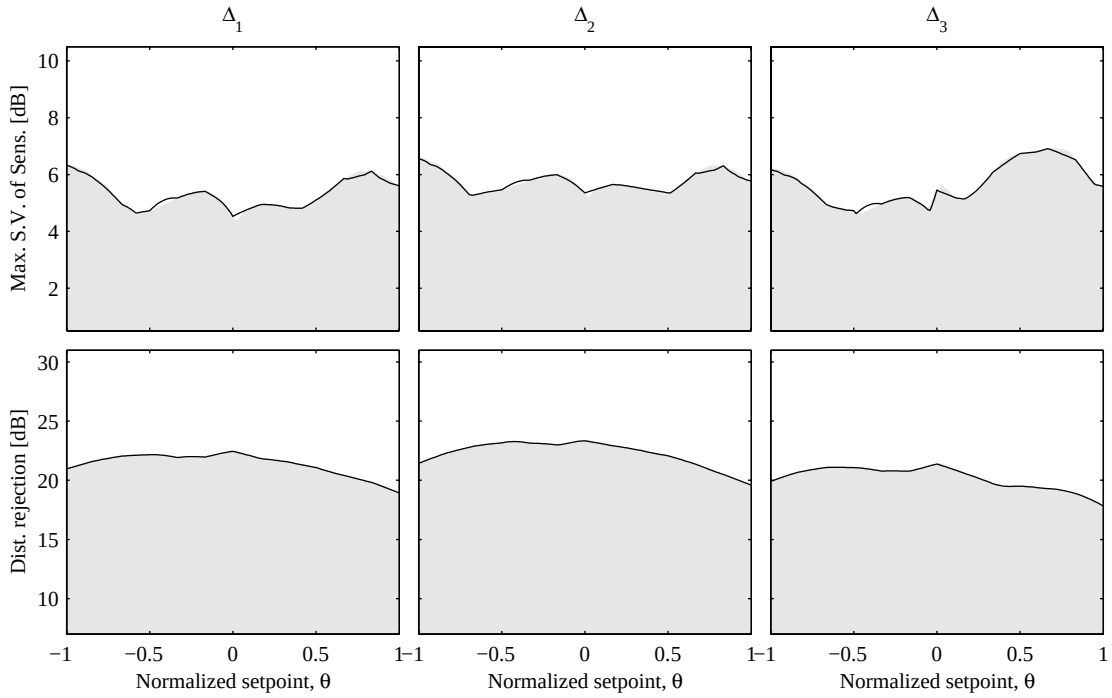

Figure 5.23 Robustness and performance of the linear interpolation scheme (solid black line). For comparison the robustness and performance of the ideal controller is shown as a shaded gray patch.

Global Shapefunction Interpolation

A global shapefunction interpolation scheme is implemented using 20 Chebyshev polynomials per element[1]. The nonlinear analysis of this gain scheduled controller is shown in Figure 5.24. As with the previous interpolation schemes, the performance of this gain

scheduled controller closely tracks the performance of the ideal controller. However, the robustness measure peaks above 12 dB close to $\theta = 0^o$. This occurs for all samples of the uncertainty space, and is a strong indicator that there are robustness issues close to $\theta = 0^o$.

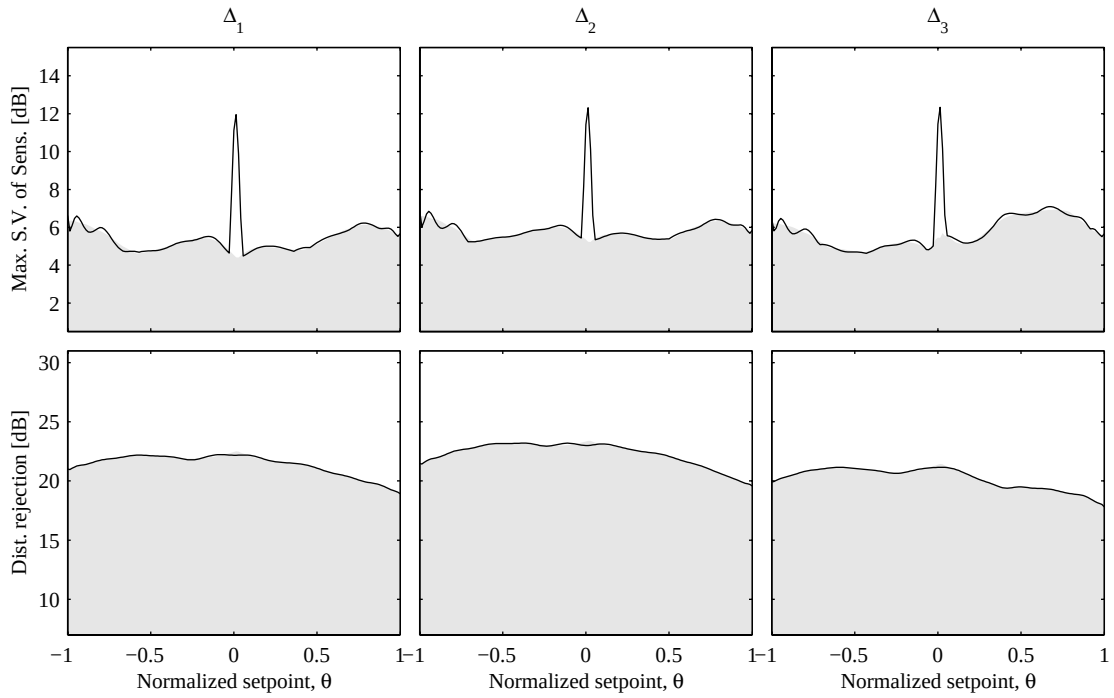

Figure 5.24 Robustness and performance of the shapefunction interpolation scheme (solid black line). For comparison the robustness and performance of the ideal controller is shown as a shaded gray patch.

The linear interpolation proved to be the most effective interpolation scheme for gain scheduled control. The next section compares the performance of this linear interpolation scheme with a benchmark LTI controller.

5.4.3 LTI Design for Nonlinear Plant

As a benchmark to compare the gain scheduled design framework of this thesis, a Linear Time Invariant (LTI) controller is designed using the Multiple Model method presented in

1. A global shapefunction interpolation scheme was also implemented with sinusoidal shapefunctions. The nonlinear analysis produced very similar results to the Chebyshev polynomial controller presented here.

Section 3.2.3. This linear controller is analyzed over the same design space as the gain scheduled controllers of the previous section. This analysis is presented Figure 5.25, together with the analysis of the ideal controller, and a gain scheduled controller that uses a 7 setpoint linear interpolation scheme. Robustness analysis for the multiple model controller suggests that there are stability problems for the uncertainty sample Δ_1 for setpoints larger than $\Theta = +0.4$, where the robustness measure exceeds 10 dB. The performance of the linear controller is consistently poorer than the gain scheduled controller deviating by more than 2 dB at several points in the design space.

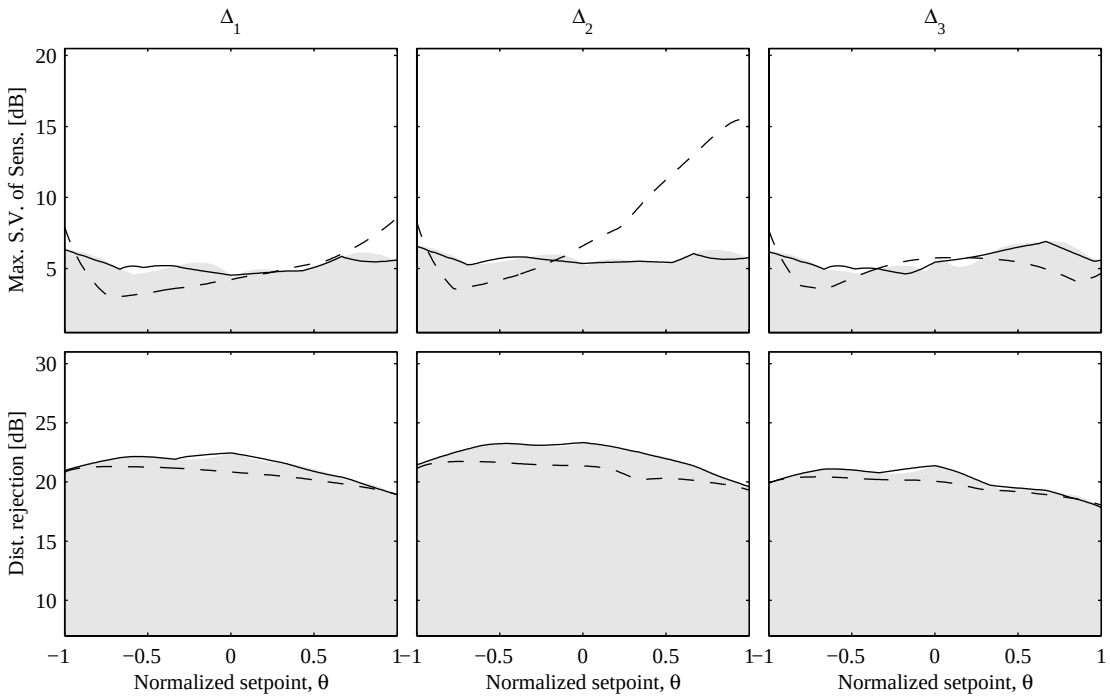

Figure 5.25 Comparison of linear multiple model controller (dashed) with gain scheduled controller (linear interpolated with 7 reference setpoints, solid line). For comparison the robustness and performance of the ideal controller is shown as a shaded gray patch.

5.4.4 Comparing Uncertainty and Nonlinearity

The results of the previous section show only a slight improvement in performance of the gain scheduled controller over the benchmark LTI controller. The gain scheduled controller improves on the multiple model controller by no more than 3 dB, and the limiting fac-

tor to control was the uncertainty in the system. In order to demonstrate some of the limitations and capabilities of the gain scheduled design framework developed in this thesis, this section presents control designs for MIT/MACE-II with decreasing uncertainty levels. A further motivation for decreasing the uncertainty is that the uncertainty is driven by the appendage attachment screws. This uncertainty could be reduced with relatively minor hardware modifications.

The first set of control designs is performed for MIT/MACE-II with the identified uncertainty. Three uncertainty samples are chosen: the nominal system designated as $\Delta = 0$; a system with all uncertain parameters shifted down by one standard deviation, $\Delta = -1$; and a system with all uncertain parameters shifted up by one standard deviation, $\Delta = +1$. Two control designs are compared; an LTI controller designed using Multiple Model techniques and a gain scheduled control design with linear interpolation. The robustness and performance measures of the closed loop analysis are shown in Figure 5.26.

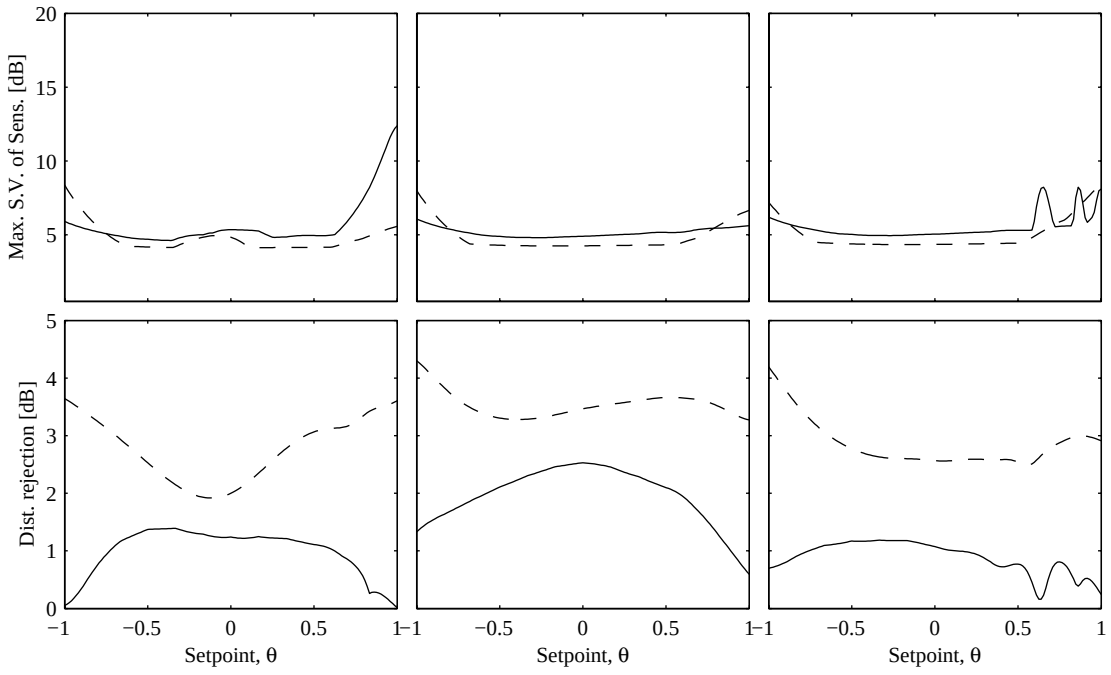

Figure 5.26 Comparison of linear multiple model controller (dashed) with gain scheduled controller (linear interpolated with 7 reference setpoints, solid line) for a system with large uncertainty, $\Delta = -1$, 0 and +1.

At this level of uncertainty, the LTI controller has better performance better over the entire evaluation space. Further more, the sensitivity singular values of the gain scheduled controller exceed 10 dB at $\Delta = -1$ and $\theta > 0.8$, indicating that there are stability robustness concerns.

The second set of control designs is for MIT/MACE-II with a reduced level of uncertainty. Three samples are again taken but with the parameter variations being 30% of the identified standard deviation. These uncertainty samples are designated as $\Delta = -0.3, 0, 0.3$. The analyses for two control designs are shown in Figure 5.27; a LTI multiple model controller and a gain scheduled controller using linear interpolation.

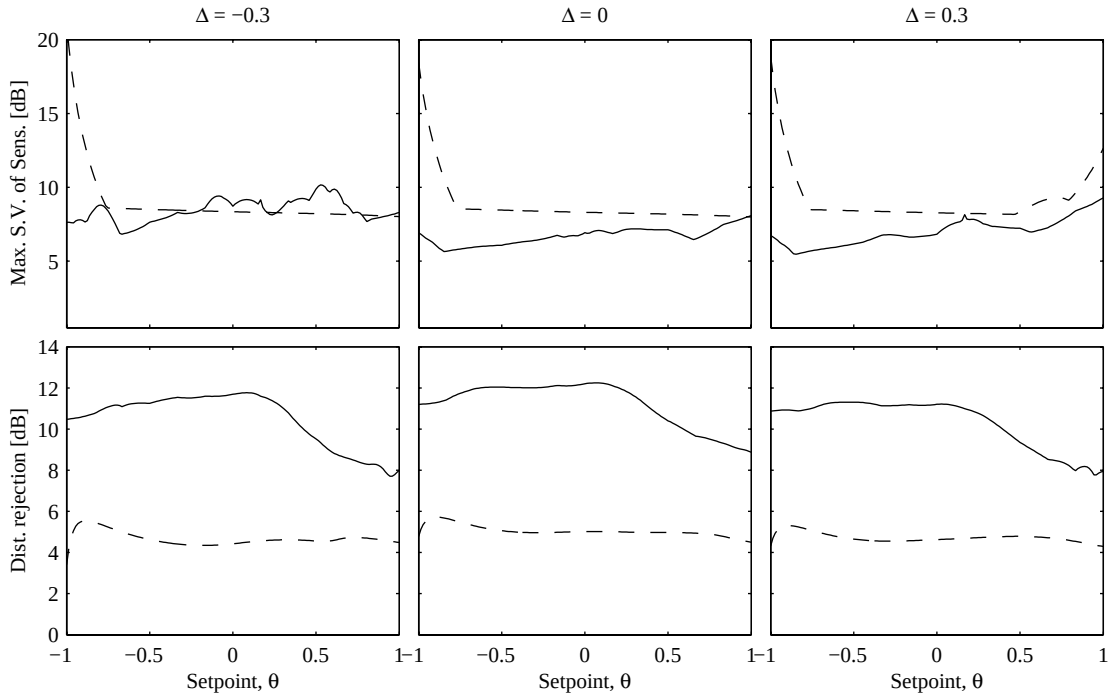

Figure 5.27 Comparison of linear multiple model controller (dashed) with gain scheduled controller (linear interpolated with 7 reference setpoints, solid line) for a system with medium uncertainty, $\Delta = -0.3, 0$ and $+0.3$.

At this level of uncertainty the gain scheduled controller performs significantly better over the entire evaluation space. The performance of the gain scheduled controller varies between 8 dB and 12 dB, with the best performance achieved at $\theta = 0$. This is an

improvement of 4 dB to 8 dB over the LTI controller. Furthermore the sensitivity singular values for the gain scheduled controller are all below 10 dB, indicating robust stability.

The final set of control designs is for MIT/MACE-II at a relatively small level of uncertainty, with the variations in the uncertainty samples being reduced to 10% of the identified standard deviation. These samples are designated as $\Delta = -0.1, 0, 0.1$. The same LTI and gain scheduled control designs are performed on this system, and the robustness and performance measures are shown in Figure 5.28.

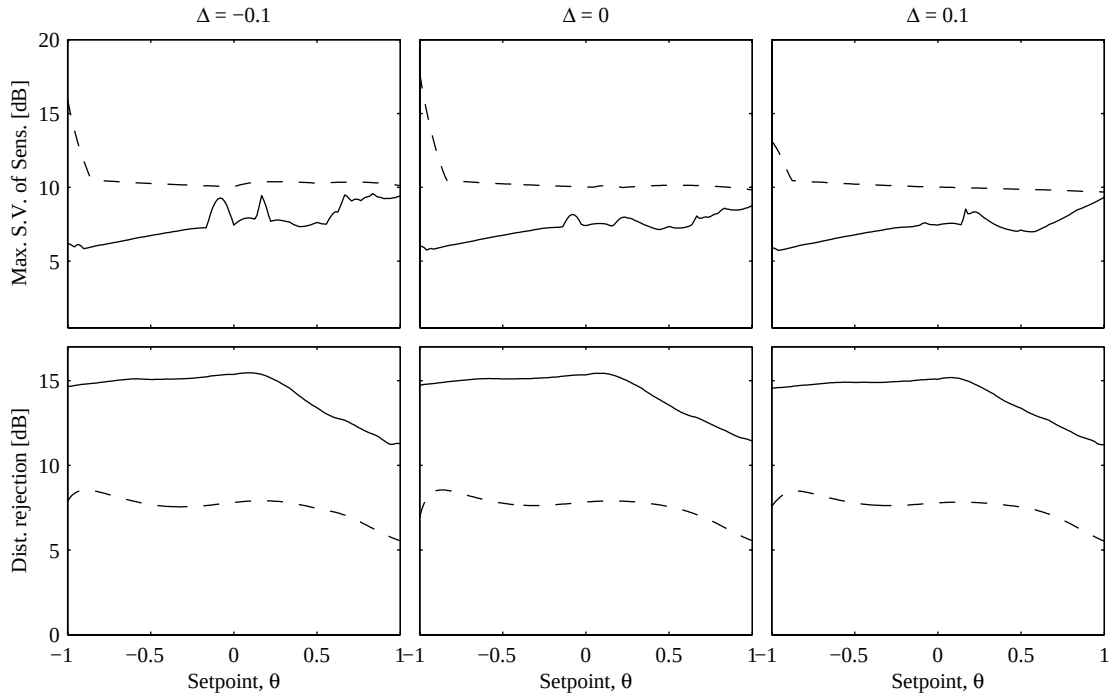

Figure 5.28 Comparison of linear multiple model controller (dashed) with gain scheduled controller (linear interpolated with 7 reference setpoints, solid line) for a system with small uncertainty, $\Delta = -0.1, 0$ and $+0.1$.

The performance of the gain scheduled controller has been further improved in comparison to the gain scheduled controllers for the two previous systems with greater uncertainty; the gain scheduled controller now achieves between 11 dB and 15 dB of performance. The LTI controller for this system achieves between 6 dB and 8 dB of performance.

A summary of the achieved performance of both the multiple model controller and the gain scheduled controller for the varying levels of uncertainty is provided in Figure 5.29. The horizontal axis represents increasing uncertainty, and the vertical axis is the achieved performance. The plot clearly shows that a gain scheduled scheme out-performs the LTI controller for small levels of uncertainty (below 30% of the identified uncertainty of MIT/ MACE-II). At the full level of uncertainty, the LTI controller out-performs the gain scheduled controller.

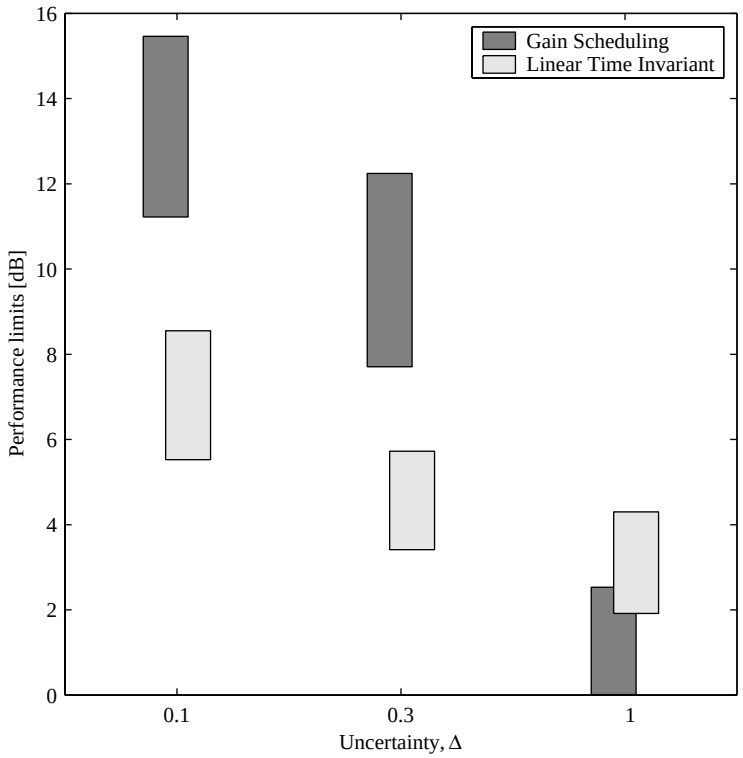

Figure 5.29 Achieved performance for gain scheduled controller design using linear interpolation and for LTI Multiple Model control. Each vertical bar represents the range of performance that is achieved over the entire evaluation space. The horizontal axis represents increasing uncertainty. The dark bars show the gain scheduled performance and the light bars show the LTI performance.

The analysis using systems with different levels of uncertainty can be summarized as follows. At larger levels of uncertainty the Multiple Model control design achieves higher performance, as it is more suited to dealing with the uncertainties. Where the uncertainty

is small, the nonlinearity dominates the problem. Gain scheduling can use the knowledge of this nonlinearity to provide greater performance than LTI controllers.

5.4.5 Limitations to Control

This section presents a discussion on the limiting factors of the gain scheduled design framework developed in this thesis. The analysis of the previous section suggests that two limiting cases should be considered; the first is where uncertainty dominates the control design ($\Delta = \pm 1.0$), and the second is where nonlinearity dominates the control design ($\Delta = \pm 0.1$).

Uncertainty Dominant

The gain scheduled controller for the system with $\Delta = \pm 1.0$ is considered in the following discussion. The frequency dependent performance at the setpoint $\theta = 0$ is shown in Figure 5.30 for the three samples of the uncertainty space, $\Delta = -1,0,+1$. The controller is only able to provide performance at the low frequencies. Most of the reduction achieved in the performance variable occurs below the bandwidth of the performance metric (below 3 rad/s). The controller cannot achieve significant performance in the cluster of modes between 50 and 80 rad/s; this region contributes greatly to the overall performance metric. Hence there is little overall performance.

Further insight can be gain by examining the plant frequency responses. The Gyu frequency response at the setpoint $\theta = 0$ is shown in Figure 5.31 for the three samples of the uncertainty space. Although the overall structure of the frequency response does not change with the uncertainty sample, a closer look at the bandwidth between 50 and 200 rad/s shows shifts in both the poles and zeros. Because the poles and zeros are lightly damped and closely spaced, a small shift can lead to large changes in the complex frequency response. For example, at 65 rad/s at $\Delta = -1$ there is a distinct zero. At $\Delta = +1$, at the same frequency there is now a lightly damped pole. This "switching" of the pole and zero implies there would be effectively be a sign change in the control loop.

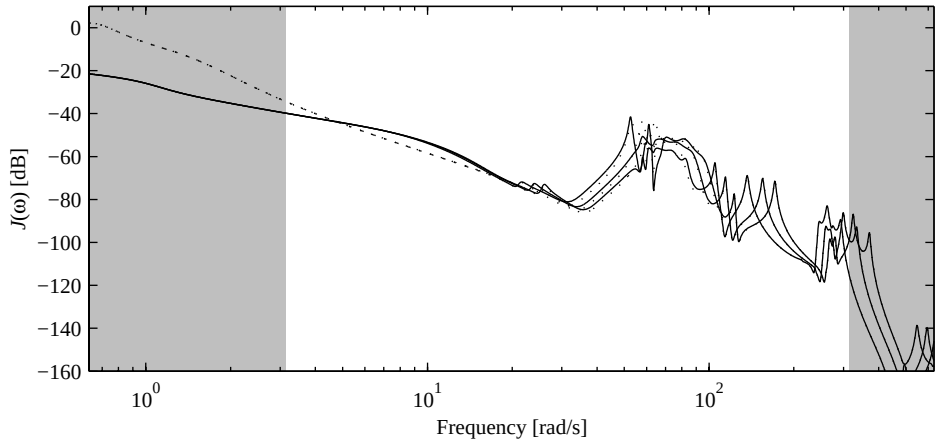

Figure 5.30 Frequency dependent performance for the gain scheduled controller at θ = 0. The closed loop response is shown as a solid line, and the open loop response is shown as a dotted line. All three samples of the uncertainty space are shown. The grey shaded areas indicate regions that do not contribute to the bandlimited performance metric.

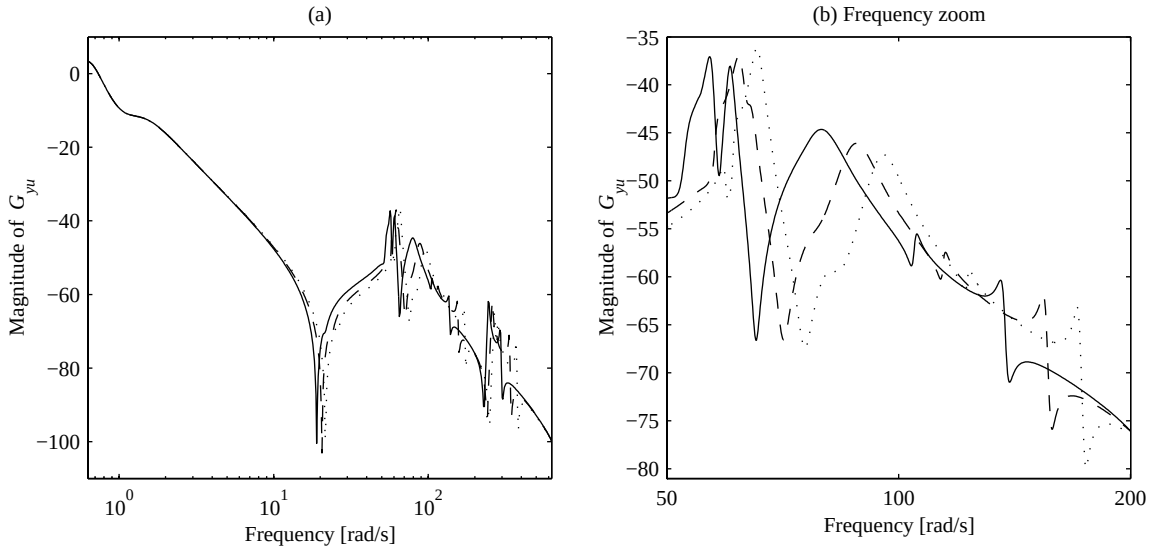

Figure 5.31 Frequency response Gyu for the plant with large (Δ = ±1) uncertainty. (a) shows the three different uncertainty samples at the full analysis range of frequencies. The solid line corresponds to Δ = +1, the dashed line to Δ = 0, and the dotted line to Δ = –1. (b) shows the same response but zoomed in to a narrower band of frequencies (50 to 200 rad/s).

If the control problem was fully analogous, this would not by itself pose a problem. Large gains could be applied without concern as there is no net phase loss for fully analogous

systems. However, the disturbance and actuator are located on opposite ends of the MBP, and the closed loop response for the non-analogous control problem must be considered,

$$G_{zw}\big|_{CL} = G_{zw} - G_{zu}K(I + G_{yu}K)^{-1}G_{yw}. \tag{5.1}$$

This equation shows that closed loop performance, $G_{zw}\big|_{CL}$, is achieved only when G_{zw}, is negated through a combination of G_{zu}, G_{yu}, G_{yw}, and the controller K. Simply applying large gains through the controller does not guarantee that the closed loop performance will be reduced. In particular, the sign changes caused by the switching of poles and zeros in the G_{yu} frequency response imply that a controller designed to provide performance at one uncertainty sample, may in fact be pushing the wrong direction when applied to a different uncertainty sample. This could possibly decrease the performance and even destabilize the closed loop. Since all points in the uncertainty space must be accounted for, the safest option is not to attempt control at all in that particular bandwidth.

To re-iterate, the modes between 50 and 80 rad/s dominate the performance metric for this control problem. However, at this level of uncertainty, the controller cannot account for the uncertain variations that are manifested as shifts in the densely spaced poles and zeros. Hence the overall performance is limited by the *uncertainty* of the system.

Nonlinearity Dominant

The gain scheduled controller for the system with $\Delta = \pm 0.1$ is considered in the following discussion. Although the uncertainty level is relatively small, sensitivity weights were applied to stabilize the closed loop system over the samples of the uncertainty space.

The linear analysis plots of performance, Nichols chart and the maximum singular values of the sensitivity transfer function are shown for the setpoint $\Delta = +1$ in Figure 5.32. Two controllers are analyzed: the final SWLQG controller design, and also the LQG controller design using the same LQG weighting matrices. The Nichols chart indicates that the LQG controller is in fact unstable with a pass-over of the –1080º critical point; this is also

reflected as a sharp spike in the singular values of the sensitivity transfer function at 160 rad/s.

The analysis plots of Figure 5.32 also indicate that the effect of the sensitivity weighting is to stabilize the closed loop without sacrificing significant performance. At this setpoint ($\theta = 0$) the performance of the SWLQG controller on the nominal system is 15.3 dB whereas the performance of the LQG controller is 15.75 dB. The SWLQG design is also able to provide *robust* performance; for example, at the same setpoint $\theta = 0$, the closed-loop system for three uncertainty samples achieves between 15 and 15.5 dB.

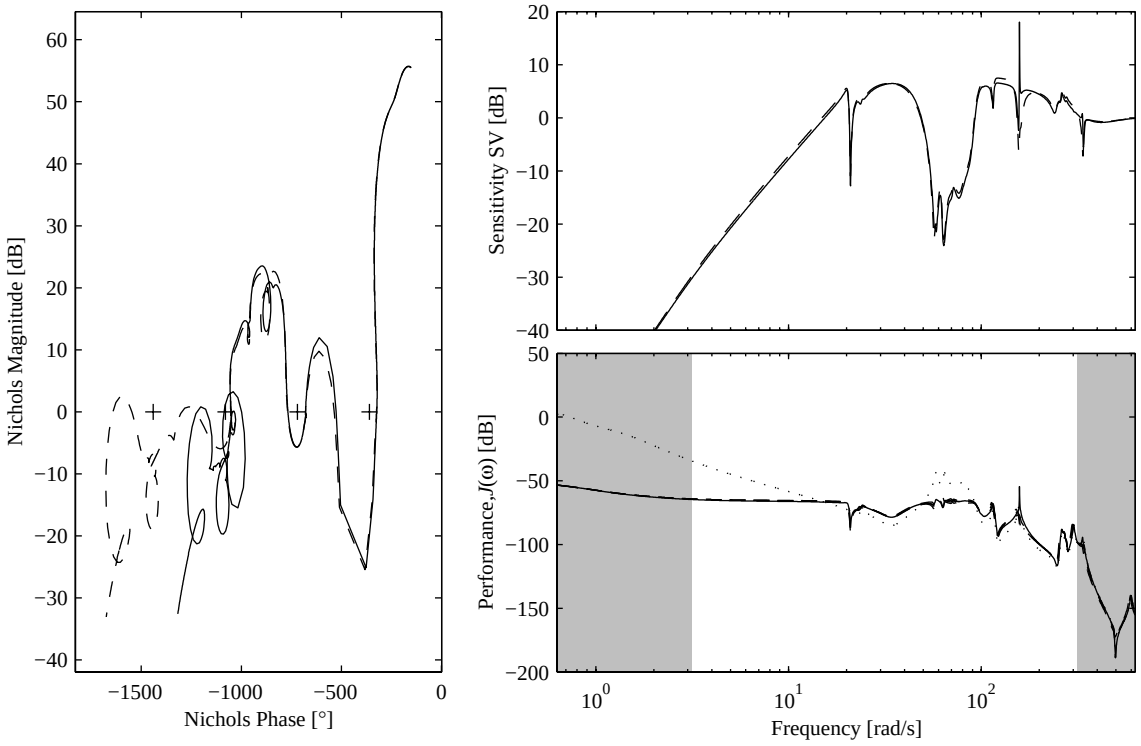

Figure 5.32 Comparison of LQG and SWLQG controller designs, for the uncertainty sample $\Delta = +1$ and the setpoint $\theta = 0$. Nominal system performance is 15.75 dB for the LQG controller and 15.3 dB for the SWLQG controller. The LQG analysis is shown as a solid line, and the SWLQG analysis is shown as a dashed line.

The conclusion thus far is that the limitation to control for this controller design is not the uncertainty, since the SWLQG controller performs robustly and provides nearly the same

performance as the LQG controller. The limitation to the control is seen during control design; further increases in gains result in a controller that has *unstable* open-loop poles. Although an open-loop unstable controller is feasible as an LQG controller, certain practical issues preclude it from implementation.

The presence of unstable poles in the controller solution can be explained as follows: Consider the Nichols chart of Figure 5.32. The curve corresponding to the Nyquist criterion for the SWLQG controller passes under several critical points as it loses phase with increasing frequency. In order to achieve more performance, more gain must be applied; however, this would entail moving closer to the critical points. LQG circumvents this restriction by adding phase with an unstable pole, so that the curve "fits" between two critical points and gain can be applied without moving closer to the critical points.

To summarize, the limitation to control for the uncertainty dominant problem is not the uncertainty, but rather it is related to the phase loss of the plant, which requires the presence of unstable controller poles to provide additional performance. This phase loss is determined by the structural topology of the control problem, so that the limitation to control is effectively due to the structural flexibility between the control actuator and the disturbance input.

5.5 Chapter Summary

5.5.1 Modeling

The modeling approach of Chapter 2 has been successfully applied to the MIT/MACE-II experiment to provide a 0-g model that incorporates the geometric nonlinearity due to large gimbal motions. This approach incorporates the modeling effort of the MACE-I program. Furthermore, data from extensive testing with the hardware is incorporated to produce an updated Finite Element model. This 1-g data is also used to develop a physical parameter uncertainty model that can be directly propagated to 0-g. The nonlinear model

and physical parameter model are combined to form a control design model that is used to design 0-g gain scheduled controllers without 0-g *in situ* test data.

5.5.2 1-g Control

The linear design tools of Chapter 3 have been successfully applied to the 1-g MIT/MACE-II control problem. This design example demonstrates the utility of the graphical heuristics used in the synthesis of LQG and SWLQG controllers. The resulting controller is implemented on the hardware and resulting performance is predicted within 0.3 dB.

5.5.3 0-g Control

The gain-scheduling framework is applied to the 0-g control design model. Several different interpolation schemes are examined with a nonlinear analysis. This nonlinear analysis demonstrates that the linear interpolation scheme performs better than a linear Multiple Model controller with improved robustness.

A further analysis is performed that compares nonlinearity with uncertainty. At the identified level of uncertainty, the LTI controller outperforms the gain scheduled controller. As the level of uncertainty is decreased, the gain scheduled controller performs significantly better than the LTI controller.

Finally, an analysis is presented to determine the limitations to control for different levels of uncertainty. At low levels of uncertainty, the limitation to control is determined to be the non-analogous nature of the control problem, since the control actuator and the disturbance input are non-collocated.

Chapter 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Thesis Summary

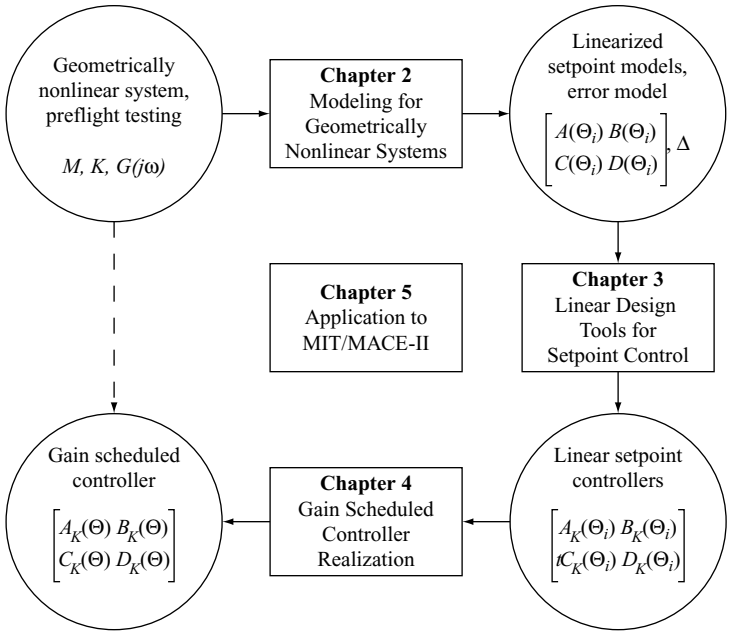

Figure 6.1 Thesis overview.

6.1.1 Modeling for Geometrically Nonlinear Systems

A methodology for modeling a geometrically nonlinear system for robust control design was presented. The methodology incorporates both geometric nonlinearity and a parametric error model. The features of the modeling process include:

- The LF form allows uncertainty and nonlinearity to be modeled *independently.*
- The LF form allows linearized models to be rapidly generated without repeated FE analysis. This is particularly useful for controller validation, where closed loop analysis must be performed at many linearized setpoints.
- The linear state space form for both model updating and nonlinear modeling allows additional dynamics (such as servos and sensor dynamics) to be trivially incorporated, which is crucial to creating a description of the dynamics as seen by the control computer.
- Both model updating and nonlinear modeling are compatible with commercial FEM packages. The use of a commercial FEM package such as NASTRAN allows complex physical structures to be captured accurately, with tractable modeling effort. Such models are typically available for flight systems.
- Errors in the model are represented by physical parameter uncertainties, allowing for direct extrapolation from experimental to *in situ* environments across the nonlinear configuration space.
- The primary disadvantage is that the higher order rate-dependent terms in the equations of motion are not described. This is not critical to the slow time variations of geometrically nonlinear systems.

6.1.2 Linear Design Tools for Setpoint Control

Design tools for the synthesis and analysis of linear controllers have been presented. Graphical heuristics have been developed to aid in the design of LQG and SWLQG controllers.

Although more sophisticated linear control design techniques can be found in the literature, the closed-form nature of the solution to LQG and SWLQG makes them attractive choices for gain-scheduled control. A requirement for gain scheduling methodology developed in this thesis is that for a small change in the plant, the corresponding change in controller designs should also be small. Iterative methods used in other design techniques generally do not possess this “smoothness” property. An additional feature of the LQG/SWLQG solution is that the solution time to Riccati equation is fixed and relatively short, which is important because gain scheduling requires many linear designs to be performed.

6.1.3 Gain Scheduled Controller Realization

A procedure for designing gain scheduled controllers for geometrically nonlinear systems was developed. The uncertain, nonlinear control design model of Chapter 2 is used as a starting point, and the linear design tools of Chapter 3 are incorporated. The procedure consists of three major steps. The first is the design of a family of linear controllers for linearized models of the nonlinear system. The designs are performed at several reference setpoints. The second step involves smoothing the state space matrices of this family of linear controllers. The final step is to implement an interpolation scheme that can interpolate between the reference setpoints so that controllers can be determined at an arbitrary setpoint. Although the design process does not directly achieve specified performance and robustness requirements, several options for iteration are incorporated in the design process to allow the control designer to gradually refine the controller to meet these requirements.

6.1.4 Application to MIT/MACE-II

The overall modeling and gain scheduling framework developed in this thesis was applied to the MIT/MACE-II experiment. The modeling approach of Chapter 2 has been successfully applied to the experiment to provide a 0-g model that incorporates the geometric nonlinearity due to large gimbal motions. This approach incorporates the modeling effort of the MACE-I program. Furthermore, data from extensive testing with the hardware is incorporated to produce an updated Finite Element model. This 1-g data is also used to develop a physical parameter uncertainty model that can be directly propagated to 0-g. The nonlinear model and physical parameter model are combined to form a control design model that is used to design 0-g gain scheduled controllers without 0-g *in situ* test data.

The linear design tools of Chapter 3 have been successfully applied to the 1-g MIT/MACE-II control problem. This design example demonstrates the utility of the graphical heuristics used in the synthesis of LQG and SWLQG controllers. The resulting controller is implemented on the hardware and resulting performance is predicted within 0.3 dB.

The gain-scheduling framework is applied to the MIT/MACE-II 0-g control design model. Several different interpolation schemes are examined with a nonlinear analysis. This nonlinear analysis demonstrates that the linear interpolation scheme performs better than a linear Multiple Model controller with improved robustness.

Although the performance for MIT/MACE-II with the identified uncertainty is poor, further analyses show that this uncertainty was the limit to performance. In particular, gain scheduling is shown to be much more effective as the level of uncertainty in the system is reduced. As the uncertainty is reduced, the limitation to control is the topology of the control problem.

All of the original thesis objectives have been achieved, and they are restated here: The overall objective of the thesis is to develop and demonstrate a framework for configuration dependent control of a geometrically nonlinear system. Specific objectives:

- Develop a framework for modeling a geometrically nonlinear system. This model includes an error model for the design of a robust controller, and must be done without *in situ* test data.
- Develop a framework for designing robust gain scheduled controllers for a geometrically nonlinear system. These controllers are designed to achieve better performance than linear time invariant (LTI) controllers for the same geometrically nonlinear system.
- Implement the modeling and control framework on an experimental testbed. This testbed is the Middeck Active Control Experiment Reflight (MIT/MACE-II), which was flown on STS-106 for deployment on the International Space Station in September 2000.

6.2 Contributions

The following contributions were made during the course of meeting the thesis objectives:

- Commercial Finite Element packages have been incorporated with a powerful nonlinear modeling methodology. The use of interface accelerations and forces allows the description of the nonlinearity to be placed in an LF form.
- An LF form has been developed to represent changes in modal and physical parameters. This form can be used for a parameter updating scheme that

matches experimental frequency responses to modeled frequency responses, and includes a term that de-emphasizes noisy signals. Repeated updating across multiple datasets is used to develop a parameter uncertainty database. The physical parameter description allows for the error model to be developed without *in situ* test data even though the 1-g model is linear and the 0-g model has nonlinear variations.

- Both the nonlinearity and uncertainty description enter through an LF description so that they can be independently incorporated into the model. Furthermore, this linearity implies that servos can be added to the model before performing the Linear Fractional Transformation that is required to obtain a linearized model at any point in the uncertainty or nonlinearity space.
- A graphical heuristics has been developed to assist the control designer in selecting weighting matrices for LQG control design. Using simple design rules, the control designer can predict the closed loop response without explicit numerical computation.
- A graphical heuristic has been developed to assist the control designer in selecting weighting matrices for SWLQG control design. Using simple design rules, the control designer can determine whether the sensitivity weights chosen will affect the eventual design of the SWLQG controller.
- A procedure has been developed to smooth the state matrices for a gain scheduled controller. This procedure consists of two tools; first a mode sort tool that determines the nonlinear pole loci of a time-varying system. Second the tool must realize real, complex or mixed modes into a unique realization. This guarantees the smoothness of the state matrices.
- A reduction tool for time-varying systems. This reduction tool uses modal rankings to preserve the smoothness of the time-varying system, and does not require possibly un-smoothing transformations such as a state-space balancing. A logarithmic cost has been introduced to rank the modes, which is effective for control design.
- A process for designing a gain scheduled controller has been developed. This process combines the uncertain nonlinear model with robust linear design tools to create a gain scheduled controller. The control designer can iterate at various points in the design process to ensure that the final controller meets design requirements.
- Linear design tools have been implemented on MIT/MACE-II in 1-g to improve the servoed performance by 22 dB, approaching a structure imposed performance limit.
- Gain scheduling controllers that outperform a benchmark linear controller have been designed for MIT/MACE-II in 0-g. Interpolation schemes of

varying complexity and refinement have been investigated. A simple linear interpolation scheme has been shown to provide performance close to that of linear controllers design at every setpoint. Confidence in the gain scheduled controllers is gained through a detailed analysis that evaluates the linearized closed loop system over a large sampling of the uncertain and nonlinear design space.

6.3 Recommendations

- A time domain parameter identification method (perhaps based on an EKF implementation of the LF parameter uncertainty description) would be a useful adjunct to the frequency-domain update process presented in this thesis. An EKF based method would provide a cross-check against errors introduced by the Fourier transform (including bias effects and smearing of nonlinear plant response), as well as generating confidence intervals for identified parameters.
- Although it is not the limiting factor to control, the modeling of damping on the nonlinear system can be extended. Because of the complex nature of the physical damping mechanism, the propagation of experimental test data to the 0-g *in situ* test environment is not well understood.
- To provide a higher degree of confidence in the gain scheduled controller, as well as a guideline as to the limiting rate of variation in the system, nonlinear time simulations can include the rate terms of the nonlinear equations of motion. Because of the formulation of the nonlinear model, the rate terms also enter through an external feedback matrix in LF form [Blaurock, 1997].
- The linear design tools presented in Chapter 3 do not take into account phase effects. Further analysis of the problem may reveal similar graphical insights into the limitations of control incurred by phase losses.
- As numerical computational methods become more efficient and computing power becomes less expensive, advanced gain scheduling techniques may become accessible. This includes the gain scheduling work of [Leith and Leithead, 1999b] for velocity based linearizations and LMI based design tools [Apkarian and Gahinet, 1995].
- There are numerous interpolation methods that can be applied to the gain-scheduling implementation. Further trade studies should be performed to determine the relative importance of controller state order and the accuracy of the interpolation scheme, since these two factors both consume real-time computing power.
- Because the interpolation methods presented combine controller parameters linearly, analytical gradients to a global cost function can be determined using Multiple Model gradients, or gradients derived for the controller tun-

ing scheme of [Mallory, 2000]. Preliminary attempts at applying gradient search techniques for the optimization of interpolated controllers resulted in minimal performance gains in spite of great computational effort. However, it is not clear whether the minimal performance gains could be improved with a better choice of optimization routine.

REFERENCES

[Apkarian and Gahinet, 1995] Apkarian, P., and P. Gahinet, "A Convex Characterization of Gain-Scheduled H_∞ Controllers", *IEEE Transactions on Automatic Control*, **40**, No. 2, pp. 853-864, 1995.

[Ashkenazi and Bryson, 1982] Ashkenazi, A., and A. E. Bryson Jr., "Control Logic for Parameter Insensitivity and Disturbance Attenuation", *AIAA Journal of Guidance, Control, and Dynamics,* **5**, No. 4, pp. 383-388, 1982.

[Bathe, 1982] Bathe, K. J., *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[Belcastro, 1998] Belcastro, C. M., "Parametric Uncertainty Modeling: An Overview", *Proceedings of the American Control Conference*, 1998.

[Balas et al., 2001] Balas, G. J., J. C. Doyle, K. Glover, A. Packard, and R. Smith, *μ-Analysis and Synthesis Toolbox*, The MathWorks, Inc., Natick, MA, 2001.

[Blaurock, 1997] Blaurock, C. A., *Modeling of Geometrically Nonlinear Flexible Structures for Control*, Ph. D. thesis, Massachusetts Institute of Technology, September 1997, SERC Report #7-97.

[Blaurock et al., 1999] Blaurock, C. A., J. H. Yung, S. P. Kenny and D. W. Miller, "Nonlinear Modeling and Control for the Middeck Active Control Experiment Reflight", *Proceedings of the AIAA Space Technology Conference & Exposition*, Albuquerque, NM, Sept. 28-30, 1999.

[Blaurock, 2001] Blaurock, C. A., *MACE II Finite Element Model Updating*, Internal Memorandum, Midé Technology Corporation, Medford MA, 2001.

[Blaurock et al., 2001] Blaurock, C. A., J. H. Yung and D. W. Miller, "0G Predictive Nonlinear Modeling for Control", *Proceedings of the AIAA Space Technology Conference & Exposition*, Albuquerque, NM, Aug. 28-30, 2001.

[Campbell, 1993] Campbell, M., *Neo-Classical Control of Structures*, S. M. thesis, Massachusetts Institute of Technology, Februrary 1993, SERC Report #4-93.

[Campbell, 1996] Campbell, M., *Uncertainty Modeling for Structural Control Analysis and Synthesis*, Ph. D. thesis, Massachusetts Institute of Technology, September 1996, SERC Report #2-96.

[Crawley et al., 1995] Crawley, E. F., M. S. Barlow, M. C. van Schoor, B. P. Masters, and A. S. Bicos, "Measurement of the modal parameters of a space structure in zero

gravity", *Journal of Guidance, Control, and Dynamics*, **18**, no. 3, pp. 385-394, 1995.

*[*D'Eleuterio and Sincarsin 1991*]* D'Eleuterio, G. M. T., and Sincarsin, G. B., "Articulational dynamics of variable-geometry truss structures", *Proceedings of the 2nd Joint Japan/U.S. Conference on Adaptive Structures*, Nagoya, Japan, Nov. 12-14, 1991.

[Desoer, 1969] Desoer, C. A., "Slowly Varying Controller dx/dt=A(t)x", *IEEE Transactions on Automatic Control*, **14**, pp. 780-781, 1969.

[Elmqvist, 1994] Elmqvist, H., *Dymola – Dynamic Modeling Language – User's Manual*, Dynasim AB, Lund, 1994.

[Geradin et al., 1989] Geradin, M., A. Cardona, and D. Granville, "Numerical Simulation in the Deployment of Space Structures", *Proceedings of the ASME/JSME Pressure Vessels and Piping Conference*, Honolulu, HI, pp. 61-66. 1989.

[Gutierrez, 1999] Guttierez, H. L., "Performance Assessment and Enhancement of Precision Controled Structures During Conceptual Design", Ph. D. thesis, Massachusetts Institute of Technology, February 1999, SERC Report #1-99.

[Glaese, 1994] Glaese, R. M., *Development of Zero-Gravity Structural Control Modesl from Analysis and Ground Experimentation*, S. M. thesis, Massachusetts Institute of Technology, January 1994, SERC Report #3-97.

[Grocott, 1994] Grocott, S. C. O., *Comparison of Control Techniques for Robust Perfomance on Uncertain Structural Systems*, S. M. thesis, Massachusetts Institute of Technology, January 1994, SERC Report #2-94.

[Hoppensteadt, 1966] Hoppensteadt, F. A., "Singular Perturbations on the Infinite Interval", *Transactions of the American Mathematical Society*, **123**, pp. 521-535, 1966.

[Hyde and Glover, 1993] Hyde, R. A., and Glover, K., "The Application of Scheduled H_∞ Controllers to a VSTOL Aircraft", *IEEE Transactions on Automatic Control*, **38**, No. 7, pp. 1021-1039, 1993.

[Ingham and Crawley, 2001] Ingham, M. D., and E. F. Crawley, "Microdynamic characterization of modal parameters for a deployable space structure", *AIAA Journal*, **39**, no 2, pp. 331-338, 2001.

[Jacques, 1995] Jacques, R. N., *On-line System Identification and Control Design for Flexible Structures*, Ph. D. thesis, Massachusetts Institute of Technology, May 1995, SERC Report #3-95.

[Kailath, 1980] Kailath, T., *Linear Systems*, Prentice-Hall Inc., Englewood Cliffs, NJ, 1980.

[Kelkar and Alberts, 1991] Kelkar, A. G., and Alberts, T. E., "Optimal Control/Structure Integrated Design of a Flexible Space Platform with Articulated Appendages", *Proceedings of the SPIE Conference on Structures, Sensing and Control*, Bellingham, WA, pp. 243-253, 1991.

[Khalil and Kokotovic, 1991] Khalil, H. K., and P. V. Kokotovic, "On Stability Properties of Nonlinear Systems with Slowly Varying Inputs", *IEEE Transactions on Automatic Control*, **36**, No. 2, pp. 229, 1991.

[Khorrami et al., 1994] Khorrami, F., S. Jain, and A. Tzes, "Experiments on Rigid Body-Based Controllers with Input Preshaping for a Two-Link Flexible Manipulator", *IEEE Transactions on Robotics and Automation*, **10**, No. 1, pp. 55-65, 1994.

[Kwakernaak and Sivan, 1972] Kwakernaak, H., and R. Sivan, *Linear Optimal Control Systems*, Wiley Interscience, New York, 1972.

[Leith and Leithead, 1999a] Leith, D. J., and W. E. Leithead, *Survey of Gain-Scheduling Analysis & Design*, Technical Report, Department of Electronics & Electrical Engineering, University of Strathclyde, 1999.

[Leith and Leithead, 1999b] Leith, D. J., and W. E. Leithead, "Input-Output Linearization by Velocity-Based Gain-Scheduling", *International Journal of Control*, **72**, pp. 229-246, 1999.

[Lim and How, 1997] Lim, S. and J. P. How, "Analysis of LPV systems Using a Piecewise Affice Parameter-Dependent Lyapunov Function", *Proceedings of the Conference on Decision and Control*, San Diego CA, pp. 978-983, 1997.

[Ljung, 1999] Ljyung, L., *System Identification: Theory for the User*, Prentice-Hall Inc., Upper Saddle River, NJ, 1999.

[Mallory, 2000] Mallory, G. W., *Development and Experimental Validation of Direct Controller Tuning for Spaceborne Telescopes*, Ph. D. thesis, Massachusetts Institute of Technology, April 2000, SERC Report #1-2000.

[Mann, 2001] Mann, H., *DYNAST Manual,* http://icosym.cvut.cz/dyn/manual/, DYN, Prague, Czech Republic, 2001.

[Meirovitch, 1986] Meirovitch, L., *Elements of Vibration Analysis*, McGraw-Hill, New York, 1986.

[Miller et al., 1996] Miller, D. W., E. F. Crawley, J. P. How, K, Liu, M. E. Campbell, S. C. O. Grocott, R. M. Glaese, T. D. Tuttle, G. Stover, J. A. Woods-Vedeler, J.

deLuis, E. Bokhour, R. Grimes, K. Scholle, C. Krebs, and R. Renshaw, *The Middeck Active Control Experiment (MACE): Summary Report*, June 1996, SERC Report #7-96.

[Moore, 1981] Moore, B. C., "Principal component analysis in linear systems: Controllability, observability, and model reduction", *IEEE Transactions on Automatic Control*, **26**, No. 1, pp. 17-32, 1981.

[Mordfin and Sivakumar, 2000] Mordfin, T. G. and S. S. K. Tadikonda, "Truth Models for Articulating Flexible Multibody Dynamic Systems", *Journal of Guidance, Control, and Dynamics*, **23**, no. 5, pp. 805-811, 2000.

[Mottershead and Friswell, 1993] Mottershead, J. E., and M. I. Friswell, "Model Updating in Structural Dynamics: A Survey", *Journal of Sound and Vibration*, **167**, no. 2, pp. 347-375, 1993.

[Mueller-Karger and Townsend, 1997] Mueller-Karger C. M., and M. A. Townsend, "Dynamics of robot arms and chains via Kane's equations", *Proceedings of the 5th Pan American Congress of Applied Mechanics*, San Juan, Puerto Rico, pp. 470-473, 1997.

[Nichols et al., 1993] Nichols, R., R. Rechert, W. Rugh, "Gain Scheduling for H_∞ Controllers: A Flight Control Example", *IEEE Transactions on Control Systems Technology*, **1**, No. 1, pp. 69-78, 1993.

[Oakley and Cannon Jr., 1990] Oakley, C. M., and R. H. Cannon, Jr., "Anatomy of an Experimental Two-Link Flexible Manipulator under End-Point Control", *Proceedings of the 29th Conference on Decision and Control*, Honolulu, HI, pp. 507-513, 1990.

[Sesak, 1974] Sesak, J. R., *Sensitivity Constrained Linear Optimal Control Analysis and Synthesis*, Ph. D. thesis, University of Wisconsin, 1974.

[Shamma and Athans, 1990] Shamma, J. S., and M. Athans, "Analysis of Gain Scheduled Control", *IEEE Transactions on Automatic Control*, **35**, No. 8, pp. 898-907, 1990.

[Slotine and Li, 1991] Slotine, J.-J., and W. Li, *Applied Nonlinear Control*, Prentice-Hall Inc., Englewood Cliffs, NJ, 1991.

[Skogestad and Postlethwaite, 1996] Skogestad, S., and I. Postlethwaite, *Multivariable Feedback Control*, John Wiley and Sons, New York, 1996.

[Tanaka, Ikeda and Wang, 1998] Tanaka, K., T. Ikeda, and H. Wang, "Fuzzy Regulators and Observers: Relaxed Stability Conditions and LMI-based designs", *IEEE Transactions on Fuzzy Systems*, **6**, pp. 250-265, 1998.

[Visser and Imregun, 1991] Visser, W. J., and M. Imregun, "A Technique to Update Finite Element Models Using Frequency Response Data", *Proceedings of the 9th International Modal Analysis Conference*, pp. 462-468, 1991.

[Wu et al., 1995] Wu, F., X. H. Yang, A. Packard, and G. Becker, "Induced L_2-norm Control For LPV Systems with Bounded Parameter-Variation", *Proceedings of the American Control Conference*, Seattle WA, pp. 2379-2383, 1995.

[Yurkovich et al., 1989] Yurkovich, S., F. Pacheco, and A. Tzes. "On-Line Frequency Domain Information for Control of a Flexible-Link Robot with Varying Payload", *IEEE Transactions on Automatic Control*, **34**, No. 12, pp. 1300-1304, 1989.

[Zhou et al., 1996] Zhou, K., J. C. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice-Hall Inc., Upper Saddle River, NJ, 1996.

Appendix A

GAIN SCHEDULED CONTROLLER IMPLEMENTATION

This appendix presents a few different implementation options that are amenable to a typical nonlinear controller architecture. The choice of the implementation is motivated by considering the requirements for controlling a geometrically nonlinear structure with limited computational resources.

A typical nonlinear controller architecture is based on two asynchronous loops and is shown in Figure A.1. The foreground loop operates with a fixed rate, typically related to the sampling rate of the actuators and sensors, and is used for operating a controller with a given set of parameters. The rate of the foreground loop is typically fixed, and depends on realtime elements such as the actuator and sensor sampling rates. The background loop has an arbitrary rate, and is used to determine and update the controller parameters. The rate of the background loop depends on the update mechanism and is typically much slower than the rate of the foreground loop. Many forms of nonlinear control have been implemented on this architecture.

Gain scheduling schemes are well suited to this architecture. A discrete state-space representation with state-space matrices $[A_K(\Theta), B_K(\Theta), C_K(\Theta), D_K(\Theta)]$ that depend on the scheduling parameter vector Θ can be implemented as,

$$\begin{bmatrix} x_{i+1} \\ u_i \end{bmatrix} = \begin{bmatrix} A_K(\Theta) & B_K(\Theta) \\ C_K(\Theta) & D_K(\Theta) \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}, \tag{A.1}$$

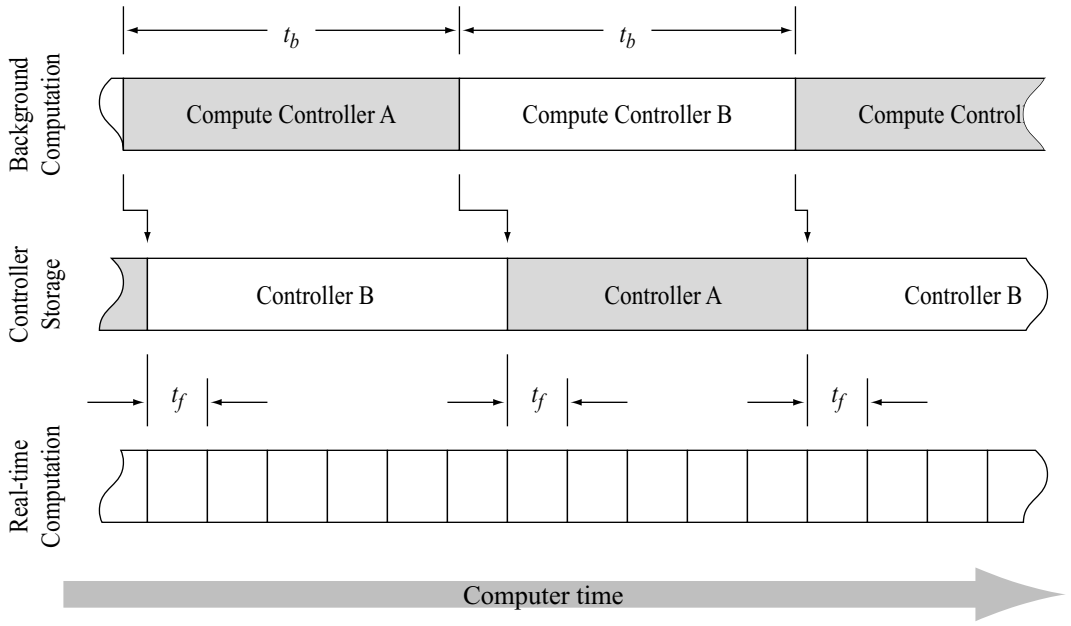

Figure A.1 Typical nonlinear controller architecture. The background update has of time period t_b, and the realtime controller has a period of t_f. The background controller updates, puts the new set of controller parameters into storage, and when this is complete, the real-time computation uses these new parameters at the next real-time cycle.

so that the control sensors are represented by y_i, the control actuators by u_i, and the updated state vector by x_{i+1}. The matrix multiplication, corresponding to the state update and calculation of the control output, is executed in the foreground loop. The updating of the varying state-space matrices, or the controller update, is executed in the background loop. With limited computation, the state order of the controller, the bandwidth of the controller, and the controller update rate are three factors that must be traded off to maximize overall performance.

An immediate modification to improve the execution rate of the foreground loop is to isomorphically transform the controller so that the matrix $A_K(\Theta)$ is block-diagonal; the matrix multiplication can be hard-coded to take advantage of this sparse structure. Systems that have distinct eigenvalues can generally be transformed so that the $A_K(\Theta)$ matrix is 2x2 block-diagonal. With this transformation the computation for the linearized controller now scales linearly with state order as opposed to quadraticly for a general state-space system. Furthermore, a requirement of sevearl gain scheduling implementations is

that the state-space matrices be smoothly varying with the scheduling parameter. The tools presented in Chapter 2 can provide this block diagonal smoothly varying realization.

Three metrics are used to evaluate each implementation of gain scheduling. These metrics arise a because of the limited capabilities of the realtime computing hardware....because the computing power that is typically available for analysis and synthesis is generally orders of magnitude above the capability of ultra-reliable and application-qualified real-time control hardware.

- *Foreground computation*—Want to compare to equivalent linear controller, by maintaining same effective state order.
- *Background computation*—How long it takes to update a controller. Faster update means better slew ability.
- *Storage requirements*—Storage of floating point controllers.

These metrics are now applied to different categories of gain scheduling architectures. The categories correspond roughly to the stage at which the linearized controller is assembled; *a priori*, actively in the background, or some combination thereof.

A.1 Controller Switching/Nearest Neighbor Interpolation

Controller switching is essentially a lookup table extended to a state-space system. The configuration space is gridded, and linear controllers are determined *a priori* for each element of the grid. Within each range the same controller is executed in the foreground loop; when a different range of the configuration space is reached, the controller is switched for the controller corresponding to the new range. This is shown in Figure A.2.

Since the controllers are determined *a priori* there is very little background computation involved; the update rate is primarily limited by the number of controllers that can be stored. If a configuration space is to be gridded with n_c controllers in each of the n_d dimensions of the space, then a total of $n_c^{n_d}$ controllers must be stored. This issue of storage is the limitation of controller switching. The fewer controllers that can be stored, the

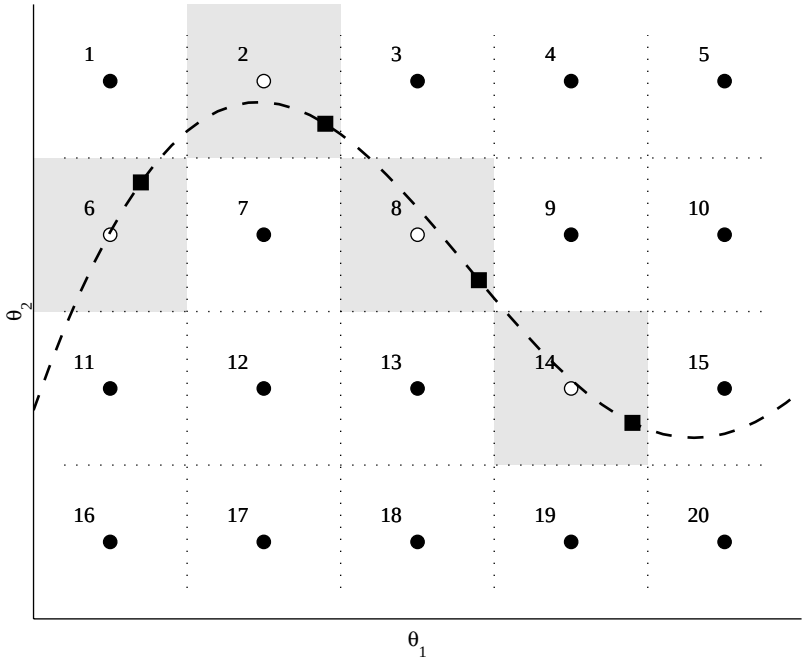

Figure A.2 Controller switching for a two dimensional configuration space. Each controller, designed *a priori*, is indicated by the solid black circle. As the system, shown by a solid black square, moves through configuration space, shown as a dashed locus, the appropriate controller is switched in. The grey square corresponds to the the entire grid element that the current controller (white circle) will be used over.

more potential there is for switching transients as the differences between controllers must become greater.

A.2 Output Interpolation

Address switching transients problem when controller grid is sparse.... the output interpolation architecture. Like controller switching, this gain scheduling architecture grids the configuration space and determines linear controllers *a priori* for each element of the grid. All of the linear controllers are stored in the realtime hardware.

To alleviate the problem of switching transients the outputs of multiple controllers are executed simultaneously and interpolated. This interpolation guarantees a smooth transition between different elements of the gridded configuration space, since the influence of

a single controller is gradually phased in and out as the system enters and exits the corresponding element of the gridded configuration space. An example of output interpolation in a two dimensional configuration space is shown in Figure A.3.

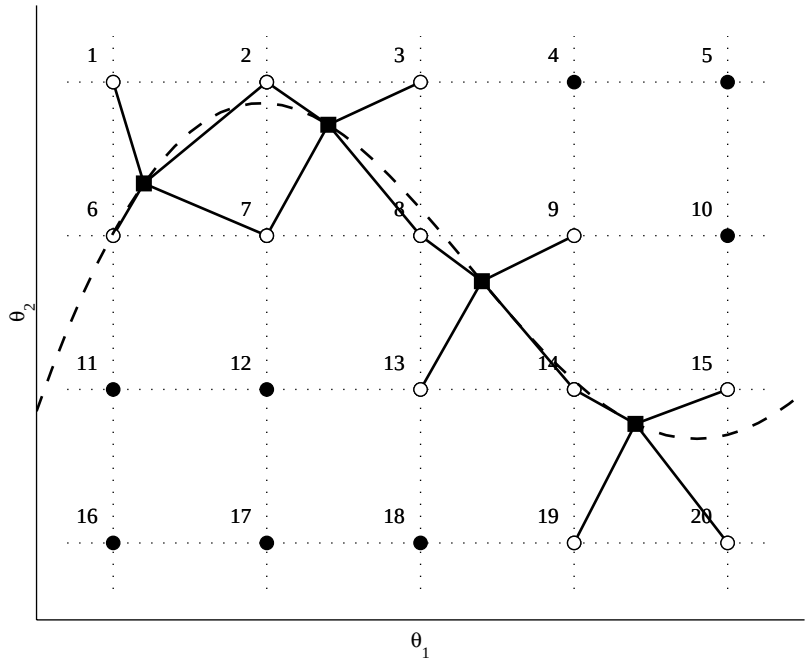

Figure A.3 Output interpolation for a two dimensional configuration space. Each controller, designed *a priori*, is indicated by the solid black circle. The system, shown by a square, moves through configuration space, shown as a dashed locus. The total control is derived from different combinations of controllers depending on the location in configuration space. 4 simultaneous controllers are required to cover the configuration space.

This implementation also requires that a total of $n_c^{n_d}$ controllers must be stored. However, the requirement of a high gridding density is reduced because of the output interpolation. The limitation to this scheme is the level of foreground computation. To interpolate between n_d dimensions 2^{n_d} controllers must be simultaneously executed, directly increasing the level of foreground computation required.

A.3 Controller Recalculation

Controller switching and output interpolation each have their disadvantages. Either an issue of storage, or an issue of too much foreground computation. This motivates the storage of just the nonlinear model, and to recalculate the controller as required (Figure A.4). This requires that the background computation perform all the steps of contoller synthesis:

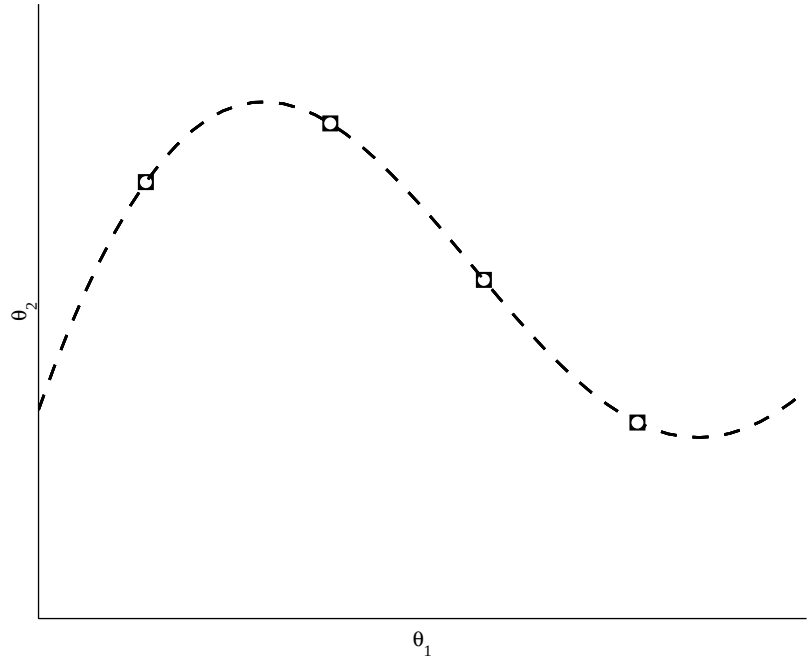

Figure A.4 Controller recalculation for a two dimensional configuration space. Controllers are not pre-calculated and do not need to be stored. Instead a compact nonlinear model is stored and the controller recalculated as necessary.

- Generating a linearized local model. The nonlinear model can be in the form presented in Chapter 2, so that the model assembly just involves the feedback of the configuration parameter matrix. The necessary control weightings (determined *a priori*) appropriate for LQG/SWLQG must also be applied.
- Solving the dual Riccati equations and assembling the linear controller.
- Transforming the controller, including converting the controller to a block diagonal realization, a controller reduction, and smoothing with respect to the currently executing controller.

- Discretizing the controller. This is facilitated by the block diagonal structure of the A matrix; Appendix A details the derivation of formulas for the discretization of a 2x2 block diagonal system.

The storage requirements for this scheme are vastly improved over the controller switching and output interpolation schemes; instead of storing multiple controllers, only a single state-space system corresponding to the model needs to be stored. The major drawback of this scheme is clearly the amount of background computation involved. If the realtime hardware cannot generate the controller quickly enough, this implementation will suffer the same problem as the controller switching implementation.

A.4 Parameter Interpolation Strategies

The three previous implementations have been limited either by storage or by computational requirements. A more suitable implementation scheme would be to try to perform some computation offline, but somehow store the results of the computation in a compact form so that they can be *interpolated*. Various interpolation schemes are discussed in Chapter 4.

Two distinct interpolation strategies are considered:

A.4.1 Gain Interpolation

The most computationally expensive calculation to perform in the controller generation is the solution of the two Riccati equations. Gain interpolation aims to relieve this by precomputing the solutions and storing the gains onboard the realtime hardware.

The amount of foreground computation remains the same as for controller switching and controller recalculation; just a single linear controller is executed. For the background computation, the tasks are a modification of the controller recalculation implementaion:

- Generating a linearized local model, for use in the model-based controller. Note that the disturbance and performance channels can be omitted.
- Interpolating the gains and assembling the controller.

- Transforming the controller, including converting the controller to a block diagonal realization, a possible controller reduction, and smoothing with respect to the currently executing controller.
- Discretizing the controller.

A nonlinear model of the control inputs and outputs must be stored, as well the parameterized controller gains.

A.4.2 Controller Interpolation

The controller interpolation implementation improves upon gain interpolation by assembling and transforming the controllers offline to save on even more background computation. Instead of storing the controller gains and the nonlinear model in the realtime hardware, an entire controller is stored.

As with the gain interpolation implementation, the foreground computation required for this implementation is just the execution of a single linear controller. The tasks for the background computation reduce to interpolating the controller matrices and discretizing the controller. Only the interpolated controller needs to be stored.

A.5 Summary of Implementation Strategies

A functional flow of each gain scheduling implementation is shown in Figure A.5, which summarizes the computation required offline, and in the background and foreground. The main difference between the schemes is at which point in time (offline or in the background) that controller generation is performed. The controller switching and output interpolation schemes place most of the controller generation effort offline, whereas all of the controller generation is performed in the background for the controller recalculation implementation. The two parameter interpolation strategies are compromises between these two extremes.

The other metric for evaluating a gain scheduling implementation is the amount of storage required. Table A.1 provides a summary of the floating point storage requirements of

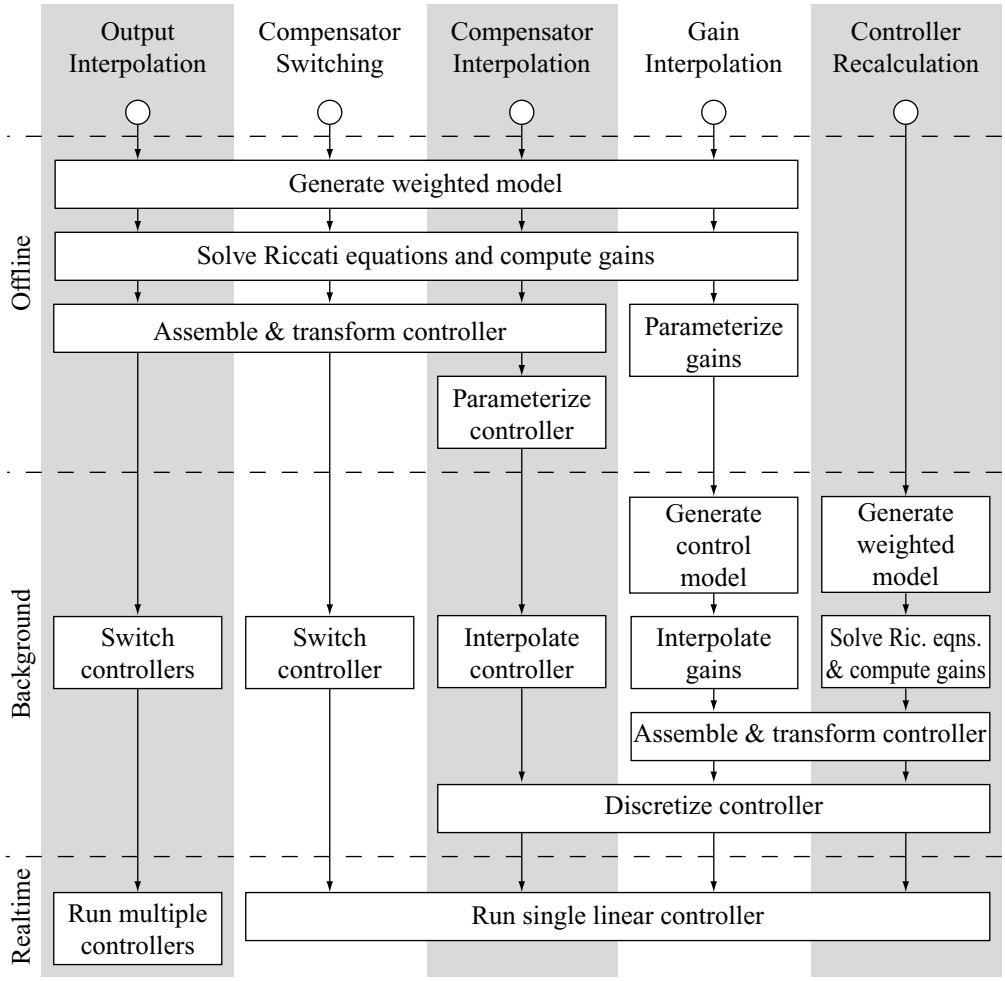

Figure A.5 Gain scheduling implementations summarized.

each implementation, in terms of the state order of the system, the number of control and performance inputs and outputs, (list them all!). These variables are defined in Table A.2.

TABLE A.1 Floating Point Storage Requirements for Gain Scheduling Implementations. See Table A.2 for variable definitions.

Implementation	Storage components	Approx. storage [floats]
Output Interpolation	• Distinct controllers	• $n_r^{n_d} n_{st}(3 + n_y + n_u)$
Switching	• Distinct controllers	• $n_r^{n_d} n_{st}(3 + n_y + n_u)$
Controller Recalculation	• Full weighted model	• $n_{st}(3 + n_y + n_u + n_z + n_w + 2n_\theta)$
Gain Interpolation	• Control model • Parameterized gains	• $n_{st}(3 + n_y + n_u + 2n_\theta)$ • $n_{st} n_f (n_y + n_u)$
Controller Interpolation	• Parameterized controller matrices	• $n_{st} n_f (3 + n_y + n_u)$

TABLE A.2 Gain-Scheduling Variable Definitions

Variable	Description	Variable	Description
n_k	State order of controller	n_z	Number of performance channels
n_p	State order of model	n_w	Number of disturbance channels
n_θ	Order of nonlinearity	n_d	Dimension of configuration space
n_y	Number of sensors	n_r	Gridding resolution per dimension
n_u	Number of actuators	n_f	Number of shapefunctions

Appendix B

IMPLEMENTATION OF THE TUSTIN TRANSFORMATION

This appendix presents the derivation of simple algebraic formulas to compute the Tustin transform of a state-space system with 2 states for the purposes of real-time implementation. This can be trivially extended to the 2x2 block diagonal gain-scheduled controller realizations of Chapters 4 and 5.

B.1 Derivation of the Tustin Transformation

The continuous system is given by:

$$G(s) = C[sI - A]^{-1}B + D. \tag{B.1}$$

We wish to approximate this continuous system on a digital computer via the Tustin transform to obtain the discrete system:

$$\tilde{G}(z) = C_z[zI - A_z]^{-1}B_z + D_z. \tag{B.2}$$

The derivation of the Tustin transform involves substituting the relation $s \Rightarrow \frac{2}{h}\left(\frac{z-1}{z+1}\right)$ into the transfer function description of the continuous system of Equation B.1, and manipulating that into the form of the discrete transfer function of Equation B.2. We have:

$$\begin{aligned} G(z) &= C\left[\frac{2}{h}\left(\frac{z-1}{z+1}\right)I - A\right]^{-1}B + D \\ &= C\left[\frac{2}{h}\left(1 - \frac{2}{z+1}\right)I - A\right]^{-1}B + D \\ &= \frac{h}{2}C\left[\left(1 - \frac{2}{z+1}\right)I - \frac{Ah}{2}\right]^{-1}B + D \\ &= \frac{h}{2}C\left[\left(I - \frac{Ah}{2}\right) + \frac{2}{z+1}I\right]^{-1}B + D \end{aligned} \tag{B.3}$$

The Matrix Inversion Lemma is the relation:

$$(A_{11} - A_{12}A_{22}^{-1}A_{21})^{-1} = A_{11}^{-1} + A_{11}^{-1}A_{12}(A_{22} - A_{21}A_{11}^{-1}A_{12})^{-1}A_{21}A_{11}^{-1} \tag{B.4}$$

We can use this relation by taking $A_{12} = A_{21} = I$. For compactness, also define $A_1 = A_{11}$, $A_2 = A_{22}$. We now have the following using relation:

$$(A_1 - A_2^{-1})^{-1} = A_1^{-1} + A_1^{-1}(A_2 - A_1^{-1})^{-1}A_1^{-1}, \tag{B.5}$$

which can be applied to the term in the square brackets of Eq. B.3, using $A_1 = \left(I - \frac{Ah}{2}\right)$ and $A_2 = \frac{2}{z+1}I$ to obtain:

$$\begin{aligned} G(z) &= \frac{h}{2}C\left[\left(I - \frac{Ah}{2}\right)^{-1} + \left(I - \frac{Ah}{2}\right)^{-1}\left(\frac{z+1}{2}I - \left(I - \frac{Ah}{2}\right)^{-1}\right)^{-1}\left(I - \frac{Ah}{2}\right)^{-1}\right]B + D \\ &= \frac{h}{2}C\left(I - \frac{Ah}{2}\right)^{-1}\left[I + \left(\frac{z+1}{2}I - \left(I - \frac{Ah}{2}\right)^{-1}\right)^{-1}\left(I - \frac{Ah}{2}\right)^{-1}\right]B + D \\ &= \frac{h}{2}C\left(I - \frac{Ah}{2}\right)^{-1}\left[\left(\frac{z+1}{2}I - \left(I - \frac{Ah}{2}\right)^{-1}\right)^{-1}\left(I - \frac{Ah}{2}\right)^{-1}\right]B + \frac{h}{2}C\left(I - \frac{Ah}{2}\right)^{-1}B + D \\ &= \frac{h}{2}C\left(I - \frac{Ah}{2}\right)^{-1}\left[\frac{z+1}{2}I - \left(I - \frac{Ah}{2}\right)^{-1}\right]^{-1}\left(I - \frac{Ah}{2}\right)^{-1}B + \frac{h}{2}C\left(I - \frac{Ah}{2}\right)^{-1}B + D \end{aligned} \tag{B.6}$$

Focus on the term between the square brackets:

$$\begin{aligned}\left[\frac{z+1}{2}I-\left(I-\frac{Ah}{2}\right)^{-1}\right] &= \left[\frac{z}{2}I+\frac{I}{2}-\left(I-\frac{Ah}{2}\right)^{-1}\right] \\ &= \left[\frac{z}{2}I+\frac{1}{2}\left(I-\frac{Ah}{2}\right)\left(I-\frac{Ah}{2}\right)^{-1}-\left(I-\frac{Ah}{2}\right)^{-1}\right] \\ &= \left[\frac{z}{2}I+\frac{1}{2}\left(I-\frac{Ah}{2}\right)^{-1}\left(\left(I-\frac{Ah}{2}\right)-2I\right)\right] \\ &= \frac{1}{2}\left[zI-\left(I-\frac{Ah}{2}\right)^{-1}\left(I+\frac{Ah}{2}\right)\right]\end{aligned} \tag{B.7}$$

Substitute this back into the expression for $G(z)$:

$$\begin{aligned}G(z) &= \frac{h}{2}C\left(I-\frac{Ah}{2}\right)^{-1}\left[\frac{1}{2}\left[zI-\left(I-\frac{Ah}{2}\right)^{-1}\left(I+\frac{Ah}{2}\right)\right]\right]^{-1}\left(I-\frac{Ah}{2}\right)^{-1}B+\frac{h}{2}C\left(I-\frac{Ah}{2}\right)^{-1}B+D \\ &= hC\left(I-\frac{Ah}{2}\right)^{-1}\left[zI-\left(I-\frac{Ah}{2}\right)^{-1}\left(I+\frac{Ah}{2}\right)\right]^{-1}\left(I-\frac{Ah}{2}\right)^{-1}B+\frac{h}{2}C\left(I-\frac{Ah}{2}\right)^{-1}B+D\end{aligned} \tag{B.8}$$

and we now have the discrete compensator in the state-space form:

$$\begin{aligned}A_z &= \left(I-\frac{Ah}{2}\right)^{-1}\left(I+\frac{Ah}{2}\right),\ B_z = \left(I-\frac{Ah}{2}\right)^{-1}B, \\ C_z &= hC\left(I-\frac{Ah}{2}\right)^{-1},\ D_z = \frac{h}{2}C\left(I-\frac{Ah}{2}\right)^{-1}B+D = \frac{h}{2}CB_z+D\end{aligned} \tag{B.9}$$

B.2 Implementation for a Two State System

For a system that is block diagonalized into 2-state blocks, we can individually transform each block. The Tustin transformation can be explicitly determined in terms of the elements of the state space matrices. Each input-ouput pair of 2x2 block-diagonal system has the form:

$$A^{(2x2)} = \begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix},\ B^{(2x2)} = \begin{bmatrix}b_1\\ b_2\end{bmatrix} \text{ and } C^{(2x2)} = \begin{bmatrix}c_1 & c_2\end{bmatrix}. \tag{B.10}$$

Remember that b_1 and b_2 are row vectors, and c_1 and c_2 are column vectors. The Tustinized system is given as:

$$\Delta \equiv \left(\frac{h}{2}\right)^2 (a_{11}a_{22} - a_{12}a_{21}) - \left(\frac{h}{2}\right)(a_{11} + a_{22}) + 1 \tag{B.11}$$

$$A_z^{(2x2)} = \frac{1}{\Delta}\begin{bmatrix} \left(\frac{h}{2}\right)^2 (a_{12}a_{21} - a_{11}a_{22}) + \left(\frac{h}{2}\right)(a_{11} - a_{22}) + 1 & 2\left(\frac{h}{2}\right)a_{12} \\ 2\left(\frac{h}{2}\right)a_{21} & \left(\frac{h}{2}\right)^2 (a_{12}a_{21} - a_{11}a_{22}) + \left(\frac{h}{2}\right)(a_{22} - a_{11}) + 1 \end{bmatrix} \tag{B.12}$$

$$B_z^{(2x2)} = \frac{1}{\Delta}\begin{bmatrix} \left(\frac{h}{2}\right)(a_{12}b_2 - a_{22}b_1) + b_1 \\ \left(\frac{h}{2}\right)(a_{21}b_1 - a_{11}b_2) + b_2 \end{bmatrix} \tag{B.13}$$

$$C_z^{(2x2)} = \frac{1}{\Delta}\left[\left(\frac{h}{2}\right)^2 (c_2a_{21} - c_1a_{22}) + \left(\frac{h}{2}\right)c_1 \left(\frac{h}{2}\right)^2 (c_1a_{12} - c_2a_{11}) + \left(\frac{h}{2}\right)c_2\right] \tag{B.14}$$

$$D_z^{(2x2)} = \frac{1}{\Delta}\left[\left(\frac{h}{2}\right)^2 (c_1a_{12}b_2 + c_2a_{21}b_1 - c_1a_{22}b_1 - c_2a_{11}b_2) + \left(\frac{h}{2}\right)(c_1b_1 + c_2b_2)\right] + D \tag{B.15}$$

The derivation is complete. Equations B.11 through B.15 are a simple algebraic equivalent to the Tustin transformation for a 2 state system.