19941228 079

APPLICATIONS OF $l_1$ AND MIXED $H_2/l_1$ OPTIMIZATION

THESIS
Mark Stephan Spillman
Captain, USAF
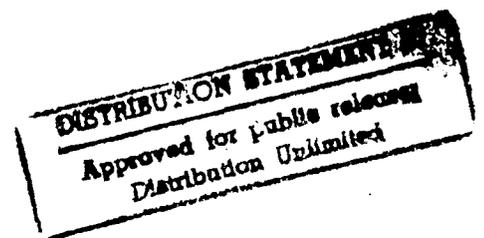
AFIT/GAE/ENY/94D-12

**DEPARTMENT OF THE AIR FORCE**

**AIR UNIVERSITY**

# AIR FORCE INSTITUTE OF TECHNOLOGY

**Wright-Patterson Air Force Base, Ohio**

AFIT/GAE/ENY/94D-12

# APPLICATIONS OF $\ell_1$ AND MIXED $H_2/\ell_1$ OPTIMIZATION

THESIS
Mark Stephan Spillman
Captain, USAF

AFIT/GAE/ENY/94D-12

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

# APPLICATIONS OF $\ell_1$ AND MIXED H$_2$/$\ell_1$ OPTIMIZATION

## THESIS

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Aeronautical Engineering

Mark Stephan Spillman, B.S.

Captain, USAF

December, 1994

## *Acknowledgments*

There are many people who have influenced the completion of this work. First and foremost, I would like to thank my thesis advisor, Dr. Brett Ridgely, for all the support he's given me. His insights, advice and editorial contributions were invaluable throughout this research project. His greatest contribution, however, was as a source of inspiration. Dr. Ridgely truly loves what he teaches, and his energy and enthusiasm passes on to his students.

I am deeply indebted to Dave Jacques for the countless hours of advice and help that he provided. Without his explanations of the $l_1$ optimization theory, I never would have been able to complete this thesis. I would also like to thank Dr. Baker for his suggestions on how to solve a crucial mathematical problem, and Dr. Bob Canfield for his theoretical contributions and comments on my thesis.

Most importantly, I would like to thank my loving wife, Kathy. Your support and encouragement have made this whole endeavor bearable. I know there's no way I can make up for all the time we've missed together in the last year and a half, but I'm looking forward to trying.

Finally, I would like thank the two people who have always been a source of encouragement and advice, my Mother and Father. I dedicate this thesis to you.

<div style="text-align: right;">Mark Stephan Spillman</div>

# Table of Contents

## List of Figures

vii

## List of Tables

AFIT/GAE/ENY/94D-12

## *Abstract*

This thesis explores the use of $\ell_1$ and mixed $H_2/\ell_1$ optimization methods to design flight control systems. $\ell_1$ optimization is used to handle tracking issues in the design of digital compensators. Control deflection and rate limitations, overshoot and undershoot limitations and steady-state error requirements are discussed. Model-matching techniques which produce acceptable tracking results with lower order controllers are also examined. New numerical methods for continuous $H_2/L_1$ and discrete $H_2/\ell_1$ optimization are presented. These methods are used to design an aircraft controller in continuous and discrete time and the results are compared.

# APPLICATIONS OF $\ell_1$ AND MIXED H$_2$/$\ell_1$ OPTIMIZATION

## I.  Introduction

### 1.1.  Overview

Much of the research in optimal flight control design in recent years has focused on H$_2$ and H$_\infty$ optimization.  While each method is well suited for specific classes of inputs and outputs of a system, neither method adequately handles the complete flight control problem.  H$_2$ optimization, for example, attempts to minimize the energy of a system's output to a white Gaussian noise input.  The resulting H$_2$ design is adept at handling noises, but can have poor stability margins.  Further, additional work is usually required to produce an H$_2$ design with good tracking.  H$_\infty$ optimization attempts to minimize the energy of a system's output to an unknown but bounded energy input.  The resulting H$_\infty$ design can have excellent stability margins and tracking performance, but be notably deficient at handling noises.  Both methods are extremely poor at handling "hard" magnitude and time domain constraints on the system, such as control deflection limitations, control rate limitations and overshoot restrictions in the system response.

The $\ell_1$ optimization method  minimizes the maximum magnitude of a system's output to an unknown but bounded magnitude input.  Since this optimization method is also a time domain approach, it can handle "hard" magnitude and time domain constraints on the system.  Little research has been done on using $\ell_1$ optimization to improve a system's stability margins; however, it has been shown ([DDB94]) that $\ell_1$ optimization can produce systems with good tracking.  Unfortunately, $\ell_1$ designs can be deficient at handling noises.

Although $\ell_1$ optimization still requires further research, it is clear that this optimization method alone will not address the complete flight control problem.  The next

logical step to a complete design methodology is to mix some of the optimization methods discussed above. Many researchers have worked in this area, but few have incorporated $\ell_1$ optimization into a mixed design approach. A notable exception is Jacques ([JR94]), who has developed a fixed-order mixed $H_2/\ell_1/H_\infty$ numerical optimization method for discrete systems. This approach, however, is relatively new and, thus, questions still remain on how to best utilize this mixed design methodology. A mixed design approach for continuous systems incorporating the continuous version of the $\ell_1$ problem, known as $L_1$ optimization, has not been developed.

Given that some mixed optimization method is desired, the next question is whether this method should be utilized in continuous or discrete time. Even though aircraft are inherently continuous systems, most aircraft controllers are actually implemented digitally. One way of producing a discrete controller in this case is to use a continuous design approach and discretize the resulting controller. Another alternative is to discretize the continuous plant and use a discrete design methodology to obtain a discrete controller directly. Perhaps the best way to determine which approach is better is to produce designs using each approach and compare the results.

This thesis explores the use of $\ell_1$ and fixed-order, mixed $H_2/\ell_1$ optimization. A fixed-order, mixed $H_2/L_1$ design approach for continuous systems is also developed to compare the two methods of producing discrete controllers mentioned above.

### 1.2. Review of Related Work

Dahleh and Diaz-Bobillo ([DDB94]) have done the most comprehensive work on $\ell_1$ optimization. They pose the $\ell_1$ optimization problem as a linear programming problem and solve it exactly for one block systems. Three methods for finding approximate solutions to multi-block problems are also presented in [DDB94]. Dahleh and Diaz-Bobillo propose a few methods of incorporating control deflection limitations, control rate limitations and overshoot restrictions in the $\ell_1$ optimization problem; however, many

implementation details are omitted and few comparisons between the different methods are shown.

Several researchers have worked on fixed-order, mixed-norm optimization methods for continuous systems, but Walker's approach ([Wal94]) solves the most general mixed-norm problem. [Wal94] discusses fixed-order, mixed $H_2/H_\infty$ optimization where the $H_2$ and $H_\infty$ subproblems can be posed with dissimilar transfer functions. Singular and multiple $H_\infty$ constraints can also be incorporated into his design method. Walker formulates a method of incorporating $L_1$ constraints into his algorithm using a forward Euler transformation of the continuous $L_1$ problem, but does not implement the method.

As mentioned in the previous section, Jacques ([JR94]) has developed a numerical, fixed-order, mixed-norm control synthesis method for discrete systems. Jacques' work essentially extends Walker's method to discrete time systems, and incorporates $\ell_1$ optimization. The $\ell_1$ portion of Jacques' algorithm, however, does not handle systems with both fast and slow dynamics due to the large number of sample periods required to estimate the pulse responses of these systems. Since most aircraft problems have fast and slow dynamics (such as actuator dynamics and phugoid modes), this restriction in Jacques' algorithm severely limits the number and type of aircraft control problems that can be considered.

## 1.3. Research Objectives

The purpose of this research is first to thoroughly investigate and compare different magnitude and time domain constraints that can be added to the $\ell_1$ optimization problem to produce systems with good tracking characteristics. Next, fixed-order, mixed $H_2/L_1$ and $H_2/\ell_1$ optimization methods will be developed which handle realistic flight control problems (i.e. systems with both fast and slow dynamics). Finally, both mixed

approaches will be used to compare two different methods of producing discrete aircraft controllers.

## 1.4. Outline

This thesis consists of 6 chapters including this introduction. Chapter 2 presents important background theory that is used throughout this thesis. Topics include state-space systems and transfer functions, transformations from continuous to discrete time, the Youla parameterization, linear programming and duality theory, $\ell_1$ and $L_1$ optimization methods, and $H_2$ optimization (both continuous and discrete).

Chapter 3 discusses using $\ell_1$ optimization to solve tracking problems. Unweighted and weighted sensitivity minimization problems are considered, along with constraints on the control deflections and rates, steady-state error, and maximum overshoot and undershoot values. Each technique is demonstrated on a realistic flight control problem. Two different model matching methods are also presented which yield acceptable tracking results with lower order controllers.

Mixed $H_2/L_1$ optimization is developed in Chapter 4. New numerical methods of evaluating the $L_1$ norm and its gradient are also presented. The $H_2/L_1$ algorithm is used to design a series of continuous aircraft controllers which demonstrate the trade-offs between pure $H_2$ and $L_1$ designs.

In Chapter 5, the mixed $H_2/\ell_1$ optimization problem is presented. New numerical methods of evaluating the $\ell_1$ norm and its gradient are developed which offer dramatic improvements in terms of computational efficiency over the methods presented in [JR94]. The new $H_2/\ell_1$ algorithm is tested on a discrete version of the aircraft control problem presented in Chapter 4, and several mixed $H_2/\ell_1$ designs are compared. The aircraft model is then tested with a discretized controller from $H_2/L_1$ optimization, and the results are compared to a similar $H_2/\ell_1$ design. Finally, Chapter 6 presents the author's conclusions and recommendations for further research.

# II. Background Theory

## 2.1. Chapter Overview

This chapter covers some of the basic theory used to conduct the research in this thesis. State-space systems and transfer functions are discussed in Section 2.2. Section 2.3 covers three transformation methods from continuous to discrete time. Section 2.4 covers the Youla parameterization. Concepts in linear programming and duality which lay the foundation for discussing $l_1$ optimization are discussed in Section 2.5. The $l_1$ optimization method is subsequently developed in Section 2.6. Section 2.7 covers $L_1$ optimization. The chapter ends with discussions of continuous $H_2$ optimization in Section 2.8 and discrete $H_2$ optimization in Section 2.9.

## 2.2. State-Space and Transfer Functions

All finite dimensional linear systems, whether continuous or discrete, can be written in state-space form. While the state-space form of a system is not a unique representation, it is often used in controller synthesis since matrix manipulations are easily handled by digital computers.

The continuous state-space form of a linear, time-invariant system is written as

$$
\begin{aligned}
\dot{x}(t) &= A_c x(t) + B_c u(t) \\
y(t) &= C_c x(t) + D_c u(t)
\end{aligned}
\tag{2.1}
$$

and the discrete state-space model is written as

$$
\begin{aligned}
x(k+1) &= A_d x(k) + B_d u(k) \\
y(k) &= C_d x(k) + D_d u(k).
\end{aligned}
\tag{2.2}
$$

In both cases, $A \in \Re^{n \times n}$, $B \in \Re^{n \times p}$, $C \in \Re^{q \times n}$ and $D \in \Re^{q \times p}$ are constant matrices, and $x \in \Re^n$, $u \in \Re^p$ and $y \in \Re^q$ are the state, control and output vectors, respectively. In the discrete state-space model x, u and y are sequences with index $k \in \{0,1,2,...\}$.

The state-space forms above are considered minimal if A has the smallest possible number of states. A state-space system is minimal if and only if (A,B) is controllable and (A,C) is observable.

The Laplace transform is used to transform continuous state-space realizations to input-output transfer functions. The z and $\lambda$ transforms perform the same function for discrete state-space realizations. Given any continuous time signal, f, starting at t=0 and sampled at T intervals, the z transform, $\tilde{F}(z)$, is defined by

$$\tilde{F}(z) = \sum_{k=0}^{\infty} f(kT)z^{-k}, \tag{2.3}$$

where $z = e^{sT}$. Note that this form of the z transform can only be used on *causal* systems, i.e. systems with current outputs that are not influenced by future inputs. This is not a significant restriction for this thesis, however, since causality will be assumed for all systems considered. The $\lambda$ transform, $\hat{F}(\lambda)$, of a continuous signal starting at t=0 is defined as

$$\hat{F}(\lambda) = \sum_{k=0}^{\infty} f(kT)\lambda^{k}. \tag{2.4}$$

Using the Laplace transform, the continuous state-space realization can be put in transfer function form,

$$T_{yu}(s) = C_c(sI - A_c)^{-1}B_c + D_c. \tag{2.5}$$

The z transform on a discrete state-space realization produces

$$T_{yu}(z) = C_d(zI - A_d)^{-1}B_d + D_d, \tag{2.6}$$

and the $\lambda$ transform produces

$$T_{yu}(\lambda) = C_d\left(\frac{1}{\lambda}I - A_d\right)^{-1}B_d + D_d. \tag{2.7}$$

Since this work considers both continuous and discrete systems, great care is taken to differentiate between the two and the different discrete domains. In places where

equations or results apply to all cases, the clarifying notation will be omitted. In addition, transfer functions in this thesis will occasionally be written in a convenient shorthand notation,

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = C(\xi I - A)^{-1}B + D, \tag{2.8}$$

where $\xi = s$, z or $1/\lambda$.

Control system design is standardly done in terms of two transfer functions, P and K, where P is the weighted plant and K is the compensator to be found (see Figure 2.1).
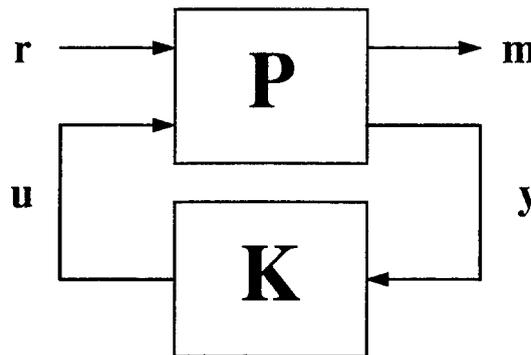


Figure 2.1. Nominal Feedback System

Note from Figure 2.1,

$$\begin{bmatrix} m \\ y \end{bmatrix} = P\begin{bmatrix} r \\ u \end{bmatrix} \qquad u = Ky \tag{2.9}$$

where m is a vector of controlled outputs and r is a vector of exogenous inputs. If P is partitioned as

$$P = \begin{bmatrix} P_{mr} & P_{mu} \\ P_{yr} & P_{yu} \end{bmatrix} \tag{2.10}$$

then the closed-loop transfer function, $T_{mr}$, can be determined from the lower fractional transformation (LFT)

2-3

$$T_{mr} = F_\ell(P, K) = P_{mr} + P_{mu}K(I - P_{yu}K)^{-1}P_{yr}. \tag{2.11}$$

The system shown in Figure 2.1 is *well posed* if and only if $(I-P_{yu}K)$ is invertible. This condition is assumed throughout this work.

P and K can also be written in terms of the state-space matrices as

$$P = \begin{bmatrix} A & B_r & B_u \\ \hline C_m & D_{mr} & D_{mu} \\ C_y & D_{yr} & D_{yu} \end{bmatrix} \qquad K = \begin{bmatrix} A_k & B_k \\ \hline C_k & D_k \end{bmatrix}. \tag{2.12}$$

Since the system is assumed to be well posed, both $(I-D_kD_{yu})$ and $(I-D_{yu}D_k)$ are invertible and the closed-loop transfer function can be written as

$$T_{mr} = \begin{bmatrix} \mathbf{A} & \mathbf{B}_r \\ \hline \mathbf{C}_m & \mathbf{D}_{mr} \end{bmatrix} \tag{2.13}$$

where

$$\mathbf{A} = \begin{bmatrix} A + B_u(I - D_kD_{yu})^{-1}D_kC_y & B_u(I - D_kD_{yu})^{-1}C_k \\ B_k(I - D_{yu}D_k)^{-1}C_y & A_k + B_k(I - D_{yu}D_k)^{-1}D_{yu}C_k \end{bmatrix} \tag{2.14}$$

$$\mathbf{B}_r = \begin{bmatrix} B_r + B_u(I - D_kD_{yu})^{-1}D_kD_{yr} \\ B_k(I - D_{yu}D_k)^{-1}D_{yr} \end{bmatrix} \tag{2.15}$$

$$\mathbf{C}_m = \begin{bmatrix} C_m + D_{mu}D_k(I - D_{yu}D_k)^{-1}C_y & D_{mu}(I - D_kD_{yu})^{-1}C_k \end{bmatrix} \tag{2.16}$$

$$\mathbf{D}_{mr} = \begin{bmatrix} D_{mr} + D_{mu}D_k(I - D_{yu}D_k)^{-1}D_{yr} \end{bmatrix}. \tag{2.17}$$

Stability for both continuous and discrete systems is determined by the eigenvalues of the **A** matrix. If all the eigenvalues of a continuous system lie in the open left-half of the complex plane, the system is stable. Discrete systems in the z-domain must have all their eigenvalues inside the open unit disk for stability, while discrete systems in the $\lambda$-domain must have all their eigenvalues outside the open unit disk. A distinct relationship

exists between the z and $\lambda$ domains, namely $z = 1/\lambda$. The next section discusses relationships between the continuous and discrete systems.

## 2.3. Continuous to Discrete

Most often, controller designs are developed for continuous systems but are actually implemented with digital controllers. The resulting mixed system is known as a sampled data or a hybrid system. One effective way to design a digital controller for such a system is to discretize the system and complete the controller design in the discrete time domain. To do this effectively, however, an appropriate method must be used to discretize the system.

There are several ways to transform a system from continuous to discrete. Some transformations maintain the integrity of the stability boundary, others do not. Some transformations are better for matching frequency responses and others for time responses. Since discrete signals only contain samples of their continuous counterparts, all the transformations tend to distort some information. The three transformations that are important in this work are presented below with their corresponding limitations.

### 2.3.1. Zero Order Hold Equivalence

In actual practice, a discrete signal does not directly drive a continuous plant. The discrete signal is passed through a digital-to-analog converter (DAC) which produces a continuous output from the discrete input. Most DAC devices convert a binary computer output to a voltage level and then hold the level for T seconds until the next computer output. These types of devices are known as zero order holds (ZOH) and represent the most common method for transforming a system from discrete time to continuous time.

The ZOH method is also known as the step invariant method because it matches the step response of a discrete system to a continuous one. For example, the step responses of the continuous system,

$$H(s) = \frac{1}{s+1}, \tag{2.18}$$

and its ZOH equivalence are shown in Figure 2.2.

The ZOH transformation in state-space form is

$$A_d = e^{A_c T}, \quad B_d = \int_0^T e^{A_c \tau} B_c \, d\tau, \quad C_d = C_c, \quad D_d = D_c, \tag{2.19}$$

where T is the sample period.
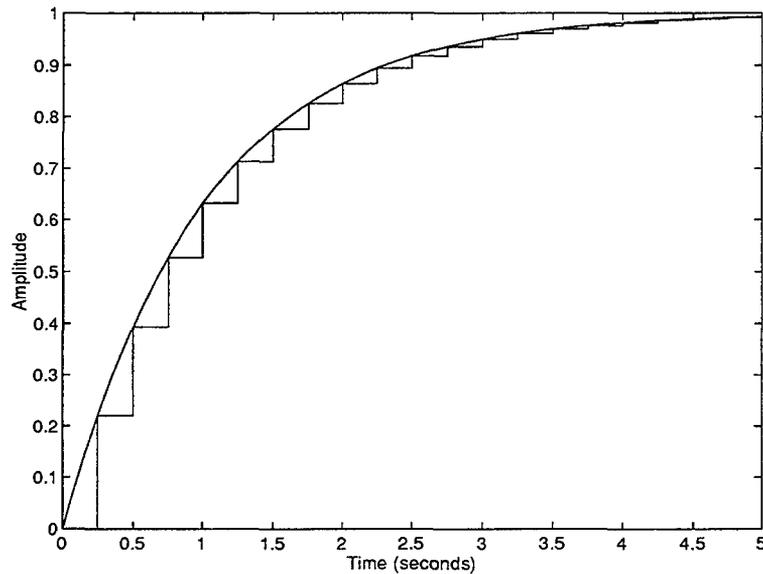


Figure 2.2. Step responses for a continuous system and its ZOH equivalent

The main limitation of ZOH equivalent systems is that their frequency responses do not match their continuous counterparts well unless the sample rate is very fast. For this reason, the ZOH method is seldom used to transform continuous filters to digital ones.

### 2.3.2. Forward Euler Rule

The forward Euler rule is one transformation method which attempts to maintain the frequency response of the continuous transfer function. This technique essentially replaces the Laplace variable, s, with a function of z. The rule is defined as

$$z = 1 + sT. \tag{2.20}$$

Since $z = e^{sT}$, it is easy to see that the forward Euler rule is simply the first two terms in the Taylor series expansion of the exponential.

The main limitation of using this method is that the $j\omega$ axis in the s plane does not map to the unit circle in the z-plane. In fact, the location of the stability boundary is dependent on the sample rate chosen to discretize the system. Consider the system in (2.18), which has a pole at -1. The forward Euler transformation of this system is

$$H(z) = \frac{T}{z - 1 + T}, \tag{2.21}$$

which has a pole at 1-T. Clearly, if the sample rate chosen is greater than or equal to 2, the resulting discrete system will be unstable.

### 2.3.3. Tustin or Bilinear Transformation

Like the forward Euler rule, the Tustin or bilinear transformation attempts to maintain the frequency response of the continuous transfer function. Unlike the forward rule, however, this transformation maintains the integrity of the stability boundary. The transformation is defined as

$$z = \frac{1 + sT/2}{1 - sT/2}. \tag{2.22}$$

To see that the stability boundary is correctly mapped in this transformation, let s=j$\omega$ in (2.22) and solve for the magnitude and phase of the resulting discrete transfer function

$$z = \frac{1 + j\omega T/2}{1 - j\omega T/2}, \quad |z| = 1, \quad \theta_z = 2 \tan \omega T/2.$$

Notice that as ω increases, the phase increases, but the magnitude is always 1. Thus, the jω axis in the s-domain correctly maps to the unit circle in the z-domain.

Since the Tustin transformation works well at maintaining frequency and stability information, it is the most common method used to discretize continuous filters. The main disadvantages of using this method, however, are that it does not maintain the step response of the continuous system and it is more difficult to implement.

A comparison of frequency responses for the system in (2.18) and its ZOH, forward Euler and Tustin transformations with T=0.5 sec is shown in Figure 2.3.
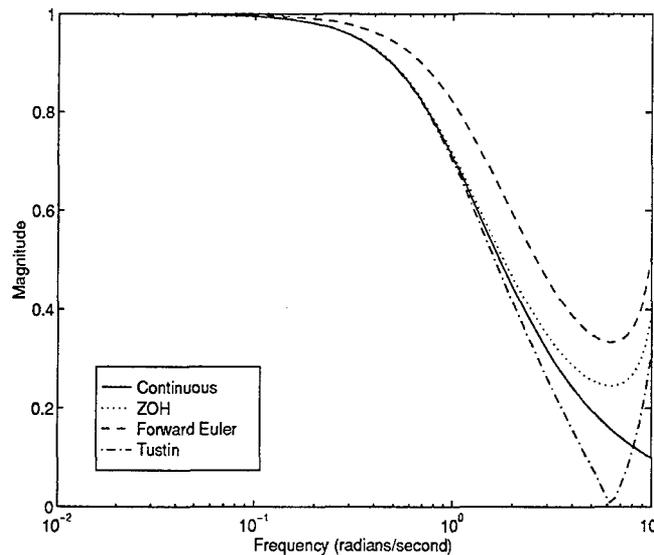


Figure 2.3. Frequency response comparisons

Note in this particular example that the sample rate is fast enough for the ZOH method to match the frequency response of the continuous system. However, the sample rate is too slow for the forward Euler rule to provide a good match.

## 2.4. Youla Parameterization

Any type of normed optimization on the closed-loop transfer function described in (2.13) involves a search over all possible stabilizing compensators. This search can, in general, be quite difficult since the closed loop transfer function is a nonlinear function of K [see (2.11)]. The Youla parameterization, however, defines the set of all possible real-rational compensators that stabilize the closed-loop transfer function in terms of a free parameter, Q. In fact, the resulting expression for K is affine (linear with an offset) in the parameter Q. Optimization problems can thus be done as a search for Q instead of K, which is much easier. The only restriction is that Q must be a stable, real-rational transfer function (the set of which are a convex set).

To understand the Youla parameterization further, consider the system in Figure 2.4, where $K = F_\ell(J,Q)$.
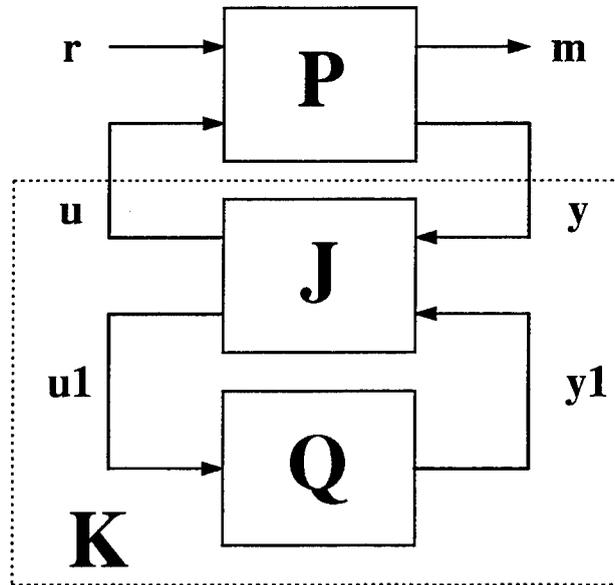


Figure 2.4. Youla parameterization

The closed-loop transfer function between m and r is given by

$$T_{mr} = F_\ell(P,K) = F_\ell(S,Q) \tag{2.23}$$

where

$$S = F_\ell(P, J). \tag{2.24}$$

If S is partitioned as

$$S = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}, \tag{2.25}$$

(2.23) becomes

$$T_{mr} = S_{11} + S_{12}Q(I - S_{22}Q)^{-1}S_{21}. \tag{2.26}$$

It can be shown ([Mac89]) that the Youla parameterization of K produces $S_{22}=0$, which simplifies (2.26) to

$$T_{mr} = S_{11} + S_{12}QS_{21}, \tag{2.27}$$

which, for ease of notation, will be written as

$$T_{mr} = H - UQV$$

from this point on. Note that this expression is indeed affine in terms of Q.

It can also be shown ([Mac89]) that if F is a gain matrix which stabilizes $A+B_uF$, and L is a gain matrix which stabilizes $A+LC_y$, then H, U and V are given by

$$H = \left[ \begin{array}{cc|c} A + B_uF & -B_uF & B_r \\ 0 & A + LC_y & B_r + LD_{yr} \\ \hline C_m & -D_{mu}F & D_{mr} \end{array} \right] \tag{2.28}$$

$$U = \left[ \begin{array}{c|c} A + B_uF & B_u \\ \hline C_m + D_{mu}F & D_{mu} \end{array} \right] \tag{2.29}$$

$$V = \left[ \begin{array}{c|c} A + LC_y & B_r + LD_{yr} \\ \hline C_y & D_{yr} \end{array} \right]. \tag{2.30}$$

The sizes of the transfer functions U and V, above, are often used to classify different types of problems. If U and V have the same number of rows as columns, then U and V are square and all-pass, or *inner*. Problems of this type are known as one block

problems. If U has fewer columns than rows, and/or V has fewer rows than columns, the problem is classified as a multi-block problem.

Note that, by definition, H, U and V are all stable transfer functions. Therefore, the closed-loop transfer function from m to r will be stable if and only if the parameter Q is stable.

## 2.5. *Linear Programming and Duality*

Since linear programming and duality concepts will be used to find $\ell_1$ solutions in the next section, it is important that they are reviewed. First, consider the standard linear programming problem,

$$
\begin{aligned}
\min_{x} \quad & c^T x \\
\text{subject to} \quad & Ax \geq b \\
& x_i \geq 0, \quad i = 1,\ldots,n
\end{aligned}
\tag{2.31}
$$

where x is a column vector of variables, c is a vector of coefficients, and A and b are a matrix and vector, respectively, which form constraints on the variables. Dimensionally, x and $c \in \Re^n$, $b \in \Re^p$ and $A \in \Re^{p \times n}$. Any optimization problem with a linear objective and linear equality and/or inequality constraints can be transformed into this form.

For each standard linear minimization problem in the above form, known as the *primal problem*, there corresponds a linear maximization problem known as the *dual problem*. The standard dual problem is written as

$$
\begin{aligned}
\max_{y} \quad & yb \\
\text{subject to} \quad & yA \leq c \\
& y_j \geq 0, \quad j = 1,\ldots,p
\end{aligned}
\tag{2.32}
$$

where $y \in \Re^p$ is a row vector of *dual* variables. Note that the dual problem simply reverses everything. In the primal problem, c was the cost function and b was the constraint. In the

dual problem, b is the cost function and c is the constraint. The inequality sign also changes between the two problems, and the unknown, y, becomes a row vector instead of a column vector.

The primal and dual problems are closely related through several key results of duality theory. First, it is known that if the primal or the dual problem has an optimal solution, then the other also has an optimal solution and their values are the same. Second, if x and y are any feasible vectors in the minimum and maximum problems, then

$$yb \leq c^T x. \tag{2.33}$$

This concept is known as weak duality. Finally, if the vectors x and y are feasible and $c^T x = yb$, then x and y are optimal.

The primal and dual variables are also related through the *complementary slackness* conditions. If x and y are feasible solutions, then both are optimal solutions of their respective problems if and only if for all i=1,...,n,

$$\text{i)} \quad x_i > 0 \Rightarrow (yA)_i = c_i,$$
$$\text{ii)} \quad (yA)_i < c_i \Rightarrow x_i = 0.$$

These conditions are also known as the alignment conditions.

## 2.6. $\ell_1$ Optimization

The goal of $\ell_1$ optimization is to minimize the maximum magnitude of the controlled output of a system given a bounded magnitude exogenous input. Vidyasagar ([Vid86]) first introduced this problem, but Dahleh and Pearson ([DP87]) are responsible for its more general solution. Dahleh and Pearson's method of solution involves posing the problem as a linear programming problem. The goal of this section is to explain their method of solution.

To simplify the explanation, the introductory development considers the case of one-block problems only. The changes necessary to find solutions to multi-block problems are discussed thereafter.

The system in Figure 2.1, where $r(k) \in \mathfrak{R}^{n_r}$ is an exogenous input sequence of unknown but bounded magnitude and $m(k) \in \mathfrak{R}^{n_m}$ is the output sequence to be controlled, represents the standard $\ell_1$ problem. If $\Phi$ is the closed-loop transfer function from m to r, then the objective of $\ell_1$ optimization can be written as

$$\min_{K \text{ stabilizing}} \|\Phi\|_1 = \min_{K \text{ stabilizing}} \left[ \max_{1 \leq i \leq n_m} \sum_{j=1}^{n_r} \sum_{k=0}^{\infty} |\phi_{ij}(k)| \right]. \tag{2.34}$$

Several steps must be taken in order to pose this as a linear programming problem. First, the nonlinear absolute value function in the norm calculation must be removed. This is accomplished by a standard change of variables from linear programming. Let $\Phi = \Phi^+ - \Phi^-$, where $\Phi^+$ and $\Phi^-$ are sequences of $n_m$ x $n_r$ matrices with positive entries. The norm calculation can then be replaced by

$$\max_{1 \leq i \leq n_m} \sum_{j=1}^{n_r} \sum_{k=0}^{\infty} \left( \phi_{ij}^+(k) + \phi_{ij}^-(k) \right), \tag{2.35}$$

which is equal to the norm if, for every (i,j,k), either $\phi^+$ or $\phi^-$ is zero. But either $\phi^+$ or $\phi^-$ must be zero at the optimal solution, so the substitution is valid. To see this fact more clearly, consider minimizing the sum of two non-negative numbers that must remain a fixed distance apart. For example, consider the problem

$$\begin{aligned} &\min (a + b) \\ &\text{subject to} \ \ a - b = 2, \ a \geq 0, \ b \geq 0. \end{aligned} \tag{2.36}$$

The solution to this problem is a=2, b=0. Note that, with the substitution of $\phi^+$ and $\phi^-$, (2.35) is linear but the number of variables has doubled.

Before searching for the variables $\Phi^+$ and $\Phi^-$ which minimize the one-norm of $\Phi$, constraints must be imposed to ensure that the resulting $\Phi$ will be stable and realizable. These two problems are handled by the Youla parameterization. Recall that

$$\Phi = H - UQV, \tag{2.37}$$

and is stable if and only if Q is stable. Also recall that U and V are inner for one block problems, and thus invertible. This means that Q can be solved for directly,

$$Q = U^{-1}(H - \Phi)V^{-1}, \tag{2.38}$$

which makes it easy to see that Q will be stable if and only if the transfer function (H-$\Phi$) cancels the unstable zeros of U and V. In other words, if the unstable zeros of U and V are denoted as $a_i$, then Q will be stable if and only if

$$\Phi(a_i) = H(a_i), \quad \text{for } 1 \le i \le N. \tag{2.39}$$

Further, if $\Phi$ is written as a function of $\lambda$,

$$\hat{\Phi}(\lambda) = \sum_{k=0}^{\infty} \Phi(k)\lambda^k, \tag{2.40}$$

then this constraint can be expressed as

$$\begin{bmatrix} 1 & a_1 & a_1^2 & \cdots \\ 1 & a_2 & a_2^2 & \cdots \\ 1 & \vdots & \vdots & \cdots \\ 1 & a_N & a_N^2 & \cdots \end{bmatrix} \begin{bmatrix} \hat{\Phi}(0) \\ \hat{\Phi}(1) \\ \hat{\Phi}(2) \\ \vdots \end{bmatrix} = \begin{bmatrix} H(a_1) \\ H(a_2) \\ \vdots \\ H(a_N) \end{bmatrix} \tag{2.41}$$

for the case of simple zeros. It can also be expressed as $A_{feas}\hat{\Phi} = b_{feas}$, which is linear in $\hat{\Phi}$.

With all of the above modifications, the $\ell_1$ optimization problem becomes

$$\min \quad \sum_{k=0}^{\infty} \hat{\Phi}^+(k) + \hat{\Phi}^-(k)$$

$$\text{subject to} \quad A_{feas}(\hat{\Phi}^+(k) - \hat{\Phi}^-(k)) = b_{feas}$$

$$\hat{\Phi}^+(k) \ge 0, \; \hat{\Phi}^-(k) \ge 0 \tag{2.42}$$

which is a linear programming problem with an infinite number of variables and a finite number of constraints. The corresponding dual problem has a finite number of variables and an infinite number of constraints. However, if there are no unstable zeros of U and V on the unit circle, at some large enough k, $a_i^k$ will be small enough that only a finite number of these constraints will be active. Thus, the dual problem is finite dimensional, and an exact solution can be found. Further, the existence of a solution to the dual problem guarantees the existence of the same solution to the primal problem. In fact, the $\ell_1$ optimization problem can be solved directly in the primal space by truncating the above series at a large enough value.

There are a few modifications that must be made to the above formulation for multi-block problems. First of all, U and V may not be invertible. However, from (2.29) and (2.30) it's clear that if $D_{mu}$ has full column rank and $D_{yr}$ has full row rank, then a left-inverse of U and a right-inverse of V will exist, which is all that is required. These two restrictions ensure that all the controls are penalized and that no measurements are perfect. Additionally, for multi-block problems it is the left unstable zeros of U and the right unstable zeros of V that must cancel with zeros of $(H - \Phi)$.

Multi-block problems have an infinite number of variables as well as an infinite number of constraints, and thus cannot be solved exactly. To counter this problem, Dahleh and Diaz-Bobillo ([DDB94]) proposed three ways to find approximate solutions. The first method, known as the *Finitely Many Variables* (FMV) approach, constrains the polynomial solution $\hat{\Phi}$ to a fixed length. The resulting compensator provides a sub-optimal but feasible solution to the problem. The second method, known as the *Finitely Many Equations* (FME) approach, truncates the number of dual variables, which is the same as solving the primal problem with a finite number of constraints. The solution to this problem is super-optimal and infeasible. The final and most viable method is known as the *Delay Augmentation* (DA) approximation. This method is generally considered the

best method to use for multi-block problems since it carries more information about the optimal solution than the other two approaches.

The DA approach embeds the multi-block problem into a larger one-block problem by augmenting pure delays to U and V. The resulting one-block problem, which contains extra degrees of freedom in Q, can then be solved exactly. While the solution to this problem is super-optimal and infeasible, it serves as an upper bound to the true optimal. To get a feasible solution, the extra degrees of freedom are simply stripped out of Q. The resulting solution is sub-optimal but provides a lower bound to the optimal solution. Thus, this method produces both a feasible solution and bounds on the optimal solution.

The $\ell_1$ optimization method is discussed further in Chapter 3. In parts of Chapter 3, the standard $\ell_1$ linear programming problem in (2.42) is augmented with constraints on the maximum magnitude of the controlled output to an exogenous step input. These additional constraints are posed using the vector $\ell_\infty$ norm,

$$\|m\|_\infty = \sup_k |m(k)| . \tag{2.43}$$

## 2.7. $L_1$ Optimization

The goal of $L_1$ optimization is the same as $\ell_1$ optimization, except that it is applied to continuous systems. While $\ell_1$ optimization attempts to minimize the absolute sum of the pulse response of a discrete system, $L_1$ optimization attempts to minimize the absolute integral of the impulse response of a continuous system.

The impulse response matrix for a continuous causal system,

$$T_{mr}(s) = \left[ \begin{array}{c|c} A_1 & B_r \\ \hline C_m & D_{mr} \end{array} \right], \tag{2.44}$$

with p inputs and q outputs is given by

$$H(t) = C_m e^{A_1 t} B_r + D_{mr} \delta(t), \tag{2.45}$$

where $\delta(t)$ is an infinite pulse of zero time duration applied at $t=0$. If $h_{ij}$ are the elements of $H(t)$, then the objective of $L_1$ optimization can be written as

$$\min_{K \text{ stabilizing}} \|T_{mr}(s)\|_1 = \min_{K \text{ stabilizing}} \left[ \max_i \sum_{j=1}^{p} \int_0^{\infty} |h_{ij}| dt \right]. \qquad (2.46)$$

Unlike $\ell_1$ optimization, there is no known way of solving the $L_1$ problem exactly. In fact, Dahleh and Diaz-Bobillo ([DDB94]) have shown that if a solution could be found, the optimal compensator would have infinite order. Theory has been developed to find approximate solutions to the $L_1$ optimization problem. Much of this theory involves discretizing the continuous system with a forward Euler transformation and performing $\ell_1$ optimization on the resulting discrete system. It can be shown ([Wal94]) that the $\ell_1$ norm of this discrete system is an upper bound to the $L_1$ norm of the original continuous system, and that the $\ell_1$ norm monotonically approaches the $L_1$ norm as the sample period is decreased to zero.

Unfortunately, this approach is not very practical for systems with slow modes. Since these systems generally have very slowly decaying pulse responses, a larger sample period is required to capture the pulse response with a reasonable number of samples. This larger sample period, however, leads to poor approximation of the $L_1$ norm. Further, it may not be possible to pick a large sample period and have the discrete system remain stable. Recall from Section 2.3.2 that the forward Euler transformation does not correctly map the stability boundary from continuous to discrete time, and larger sample periods increase the likelihood that the discrete system will be unstable.

Two numerical methods of computing the $L_1$ norm of a continuous system are presented in Chapter 4. Both methods avoid the above complications by approaching the problem more directly. The primary focus in Chapter 4 is fixed-order, mixed $H_2/L_1$ optimization for continuous systems. Continuous $H_2$ optimization is discussed in the next section.

## 2.8. *Continuous $H_2$ Optimization*

The goal of $H_2$ optimization is to find the internally stabilizing controller which minimizes the energy of the system output to a white Gaussian noise input. The continuous time development is shown here, and the discrete time version is shown in the next section.

Consider the standard $H_2$ problem in Figure 2.5, where w is a zero-mean white Gaussian noise input with unit intensity and z is the controlled output.
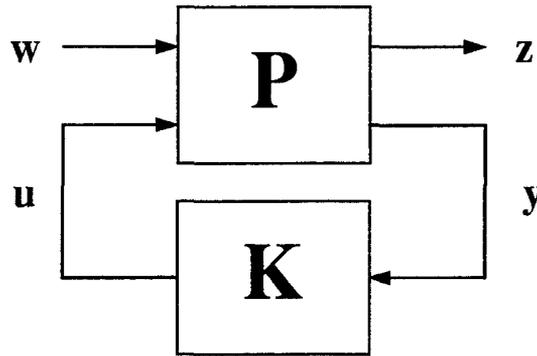


Figure 2.5. $H_2$ problem

Given this setup, the $H_2$ optimization objective can be written as

$$\inf_{K \text{ stabilizing}} \|T_{zw}\|_2 . \tag{2.47}$$

If $T_{zw}$ is written as

$$T_{zw} = \left[\begin{array}{c|c} A_2 & B_w \\ \hline C_z & D_{zw} \end{array}\right], \tag{2.48}$$

then the two-norm of $T_{zw}$ can be expressed as

$$\|T_{zw}(s)\|_2^2 = \mathrm{tr}[QC_z^T C_z], \tag{2.49}$$

where Q is the positive semidefinite solution to the Lyapunov equation

$$A_2 Q + Q A_2^T + B_w B_w^T = 0. \tag{2.50}$$

P can also be written in terms of state-space matrices as,

$$P = \left[\begin{array}{c|cc} A & B_w & B_u \\ \hline C_z & D_{zw} & D_{zu} \\ C_y & D_{yw} & D_{yu} \end{array}\right].$$ (2.51)

The following assumptions are now made on the state-space elements of P(s):

i) $D_{zw} = 0$

ii) $D_{yu} = 0$

iii) $(A, B_u)$ is stabilizable and $(C_y, A)$ is detectable

iv) $D_{zu}^T D_{zu} = I$ and $D_{yw} D_{yw}^T = I$

v) $\left[\begin{array}{cc} A - j\omega I & B_u \\ C_z & D_{zu} \end{array}\right]$ has full column rank for all $\omega$

vi) $\left[\begin{array}{cc} A - j\omega I & B_w \\ C_y & D_{yw} \end{array}\right]$ has full row rank for all $\omega$

Assumption i) is required in order to ensure that $T_{zw}$ will have a finite two-norm. Assumption ii) is not required, but it simplifies the problem. Assumption iii) is required for a stabilizing compensator to exist. Assumption iv) ensures that $D_{zu}^T D_{zu}$ and $D_{yw} D_{yw}^T$ have full rank which keeps the problem from being singular. This assumption can be relaxed to just full rank requirements on $D_{zu}^T D_{zu}$ and $D_{yw} D_{yw}^T$ through scaling. Finally, assumptions v) and vi) are required for the existence of stabilizing solutions to the algebraic Riccati equations used in the $H_2$ solution.

The unique controller which minimizes (2.47) is given by

$$K_{2\,opt}(s) = \left[\begin{array}{c|c} A_J & K_f \\ \hline -K_c & 0 \end{array}\right],$$ (2.52)

where

$$A_J = A - K_f C_y - B_u K_c$$ (2.53)

$$K_c = B_u^T X + D_{zu}^T C_z \tag{2.54}$$

$$K_f = Y C_y^T + B_w D_{yw}^T. \tag{2.55}$$

X and Y are the real, unique, symmetric, positive semidefinite solutions to the algebraic Riccati equations

$$(A - B_u D_{zu}^T C_z)^T X + X(A - B_u D_{zu}^T C_z) - X B_u B_u^T X + \hat{C}_z^T \hat{C}_z = 0 \tag{2.56}$$

$$(A - B_w D_{yw}^T C_y)^T Y + Y(A - B_w D_{yw}^T C_y) - Y C_y C_y^T Y + \hat{B}_w^T \hat{B}_w = 0, \tag{2.57}$$

where

$$\hat{C}_z = (I - D_{zu} D_{zu}^T) C_z \tag{2.58}$$

and

$$\hat{B}_w = B_w (I - D_{yw}^T D_{yw}). \tag{2.59}$$

The minimum two-norm which corresponds to $K_{2\,opt}(s)$ is denoted as $\underline{\alpha}$.

## 2.9. Discrete $H_2$ Optimization

While the goal of $H_2$ optimization does not change in the discrete time case, there are some distinct differences between the discrete and continuous time solutions. The purpose of this section is to highlight these differences.

$T_{zw}(z)$ can also be expressed in terms of the state-space matrices in (2.48). The two-norm of $T_{zw}$ in discrete time is given by

$$\|T_{zw}(z)\|_2^2 = \text{tr}[\mathbf{D}_{zw}\mathbf{D}_{zw}^T + \mathbf{C}_z Q \mathbf{C}_z^T], \tag{2.60}$$

where Q is positive semidefinite solution to the discrete Lyapunov equation

$$\mathbf{A}_2 Q \mathbf{A}_2^T + \mathbf{B}_w \mathbf{B}_w^T = Q. \tag{2.61}$$

Given the state-space description of P(z) in (2.51), let

$$\begin{bmatrix} C_z^T \\ D_{zu}^T \end{bmatrix} \begin{bmatrix} C_z & D_{zu} \end{bmatrix} = \begin{bmatrix} Q_c & S_c \\ S_c^T & R_c \end{bmatrix} \tag{2.62}$$

and

$$\begin{bmatrix} B_w \\ D_{yw} \end{bmatrix} \begin{bmatrix} B_w^T & D_{yw}^T \end{bmatrix} = \begin{bmatrix} Q_f & S_f \\ S_f^T & R_f \end{bmatrix}. \tag{2.63}$$

The following assumptions are now made on the state-space elements of the plant, $P(z)$, using the above definitions:

i) $(A, B_u)$ is stabilizable and $(C_y, A)$ is detectable

ii) $R_f, R_c > 0$

iii) $D_{zu}^T C_z = 0$ and $B_w D_{yw}^T = 0$

Assumption i) is required in order for a stabilizing compensator to exist. Assumption ii) ensures the existence of stabilizing solutions to the discrete algebraic Riccati equations. Assumption iii) is a standard assumption which is not required but simplifies the derivation.

The unique discrete $H_2$ optimal compensator is given by

$$K_{2opt}(z) = \left[ \begin{array}{c|c} A - AK_fC_y - B_uK_cK_fC_y & AK_f - B_uK_cK_f \\ \hline -K_c + K_cK_fC_y & -K_cK_f' \end{array} \right], \tag{2.64}$$

where

$$K_c = (R_c + B_u^T X B_u)^{-1}(S_c^T + B_u^T X A) \tag{2.65}$$

$$K_f = (YC_y^T + S_f^T)(C_y YC_y^T + R_f + C_y S_f^T + S_f C_y^T)^{-1}. \tag{2.66}$$

X and Y are the real, unique, symmetric, positive semidefinite solutions to the discrete algebraic Riccati equations

$$A^T X A + Q_c - (A^T X B_u + S_c)(R_c + B_u^T X B_u)^{-1}(B_u^T X A + S_c^T) = X \tag{2.67}$$

$$AYA^T + Q_f - (AYC_y^T + S_f^T)(R_f + C_y YC_y^T)^{-1}(C_y YA^T + S_f) = Y. \tag{2.68}$$

Notice from (2.52) and (2.64) that $D_k$ equals zero in the continuous time case but it is not equal to zero in the discrete time case. This fact is specifically pointed out because many references in the control literature incorrectly state that $D_k$ equals zero in

the discrete solution. In fact, assuming that $D_k$ equals zero generally results in a sub-optimal solution.

## 2.10. Chapter Summary

This chapter developed some essential background theory which will be required in Chapters 3-5. The theory on $L_1$ and continuous $H_2$ optimization will be important in Chapter 4, where mixed $H_2/L_1$ optimization for continuous systems is discussed. Discrete $H_2$ optimization theory is used in Chapter 5 to develop a mixed $H_2/\ell_1$ optimization method for discrete systems. The next chapter discusses using $\ell_1$ optimization to solve tracking problems.

# III. Using $\ell_1$ Optimization

## 3.1. Chapter Overview

This chapter focuses on using $\ell_1$ optimization to solve tracking problems. Section 3.2 describes some changes that had to be made to the $\ell_1$ optimization software prior to conducting most of the work in this chapter. Section 3.3 discusses $\ell_1$ optimization on a conceptual level, and brings up an important relationship between the unit pulse and step responses. Unweighted and weighted sensitivity minimization are discussed in Section 3.4. Sections 3.5-3.7 cover different constraints that can be added to the $\ell_1$ optimization problem to handle several tracking issues. The last section in this chapter discusses the use of $\ell_1$ optimization for model matching problems.

Throughout the chapter, a Single Input Single Output (SISO) longitudinal model of the AFTI F-16, shown in detail in Appendix A, is used to illustrate various tracking design issues. The tracking problem described in this chapter is defined as the ability to accurately command a 1g normal acceleration of the aircraft. The stabilator is the only control surface considered in the model, and it is limited to $\pm 25$ degrees deflection angle and $\pm 60$ degrees/sec deflection rate. In Section 3.8 a model matching design is completed for this system and also for a Multiple Input Multiple Output (MIMO) system involving a missile. The objective of the missile problem is stated in Section 3.8 and a detailed description of this system is given in Appendix B.

All simulations in this chapter are done with sampled-data systems (i.e. discrete controllers with continuous system models). Step inputs, applied one second after simulations are started, are used to evaluate tracking performance. Plots of the control rate are based on finite difference calculations over the discretization period of the system.

## 3.2. $l_1$ Optimization Software

MATLAB™ software used to perform $l_1$ optimization was first written by Diaz-Bobillo in 1991-92. This research code was used to run the examples in the book, *Control of Uncertain Systems: A Linear Programming Approach*, written by Dahleh and Diaz-Bobillo ([DDB94]). A copy of the software and a preprint of the book were used to conduct the research in this thesis.

After running several examples with Diaz-Bobillo's software, it became apparent that certain "bugs" existed. The first error occurred with certain simple one-block systems. While the main routine always displays a lower and upper bound to the one-norm for all systems, the two bounds should be the same for one-block systems since they can be solved exactly. Due to numerical differences in the ways the upper and lower bounds are calculated, the two bounds are never exactly the same. However, for several one-block systems there were wide disparities between the two values. The problem was traced to a routine which rounded off the unstable zero frequencies of U and V. The amount of round-off affected the interpolation conditions given in (2.41) and thus the norm calculations. This problem was corrected by eliminating the round-off algorithm altogether.

The second problem encountered with Diaz-Bobillo's code was that it did not directly ensure the required existence of a left inverse of U and a right inverse of V. It appears that Diaz-Bobillo attempted to address this issue by changing the discrete system to continuous time through a bilinear transformation. In many cases, this action leads to a system where $D_{mu}$ has full column rank and $D_{yr}$ has full row rank, which ensures that U has a left inverse and V has a right inverse. Unfortunately, these conditions must be met for a discrete time system, not a continuous one, and a bilinear transformation back to discrete time recreates the rank defect problem. Diaz-Bobillo's method to handle this issue is to replace the original discrete D matrix with the continuous D matrix. However, this results

in a new discrete system which can be significantly different than the original discrete system.

This problem was solved by simply forcing the user to input discrete systems where $D_{mu}$ has full column rank and $D_{yr}$ has full row rank. Effectively, these requirements ensure that all control usage is penalized and that no measurements are perfect. These conditions are also standard requirements for the state-space solution of $H_\infty$ problems presented in [DGKF89]. With this change, several transformations back and forth from continuous to discrete time were eliminated in Diaz-Bobillo's code. This proved to be a major overhaul of the computer code, however, since many parts of the old routine depended on the continuous time version of the original discrete system. Among other changes, new routines had to be developed to perform Youla parameterizations and stable projections of discrete systems.

Since much of the code had to be rewritten to incorporate the above changes, other simple modifications were also included to make the routine more user-friendly and efficient. As a result, the main routine is now a function file instead of a series of scripts, which frees up additional memory in the MATLAB™ workspace for the user. Data structures used in the $\mu$-Analysis and Synthesis Toolbox, which are more efficient and easier to use, were also incorporated. All the modifications to the original software were sent back to researchers at the Massachusetts Institute of Technology (MIT) who are preparing the $\ell_1$ optimization software for commercial release.

## 3.3. Understanding $\ell_1$ Optimization

In order to answer the question of how best to use $\ell_1$ optimization, it is important to first understand what $\ell_1$ optimization is trying to accomplish. The formal mathematical definition given in Chapter 2 is not important here; rather, a simple conceptual idea of how the method works is sufficient. By definition, $\ell_1$ optimization attempts to minimize the absolute sum of a system's sampled pulse response. Conceptually, this optimization

method works by pushing down on the pulse response from all sides. In other words, both peak-to-peak gains and long pulse responses are penalized since both tend to increase the absolute sum.

Since the primary interest in this chapter is how best to use $\ell_1$ optimization for tracking problems, it is instructive to examine the unit pulse and step responses of a simple discrete system. Consider the continuous system,

$$H(s) = \frac{1}{s+1} = \left[\begin{array}{c|c} -1 & 1 \\ \hline 1 & 0 \end{array}\right], \tag{3.1}$$

which discretized at 1/3 Hz using a ZOH is equivalent to

$$H(z) = \left[\begin{array}{c|c} .05 & .95 \\ \hline 1 & 0 \end{array}\right]. \tag{3.2}$$

The sampled unit pulse response of this system is shown in Figure 3.1.
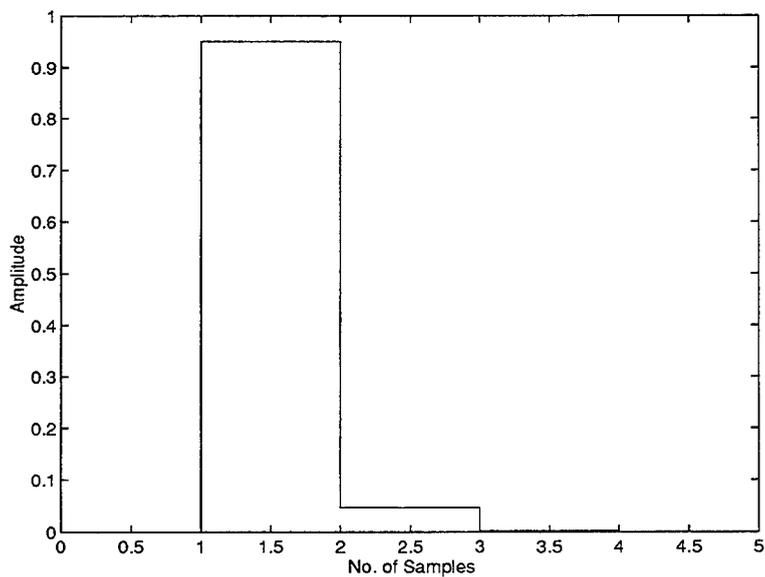


Figure 3.1. Pulse response of a discrete system

The one-norm of this system can be calculated by inspection. The total sum of samples 1-4 in Figure 3.1 appears to be approximately 1. Indeed, the one-norm for this system is 1. The unit step response of the system in (3.2) is shown in Figure 3.2.
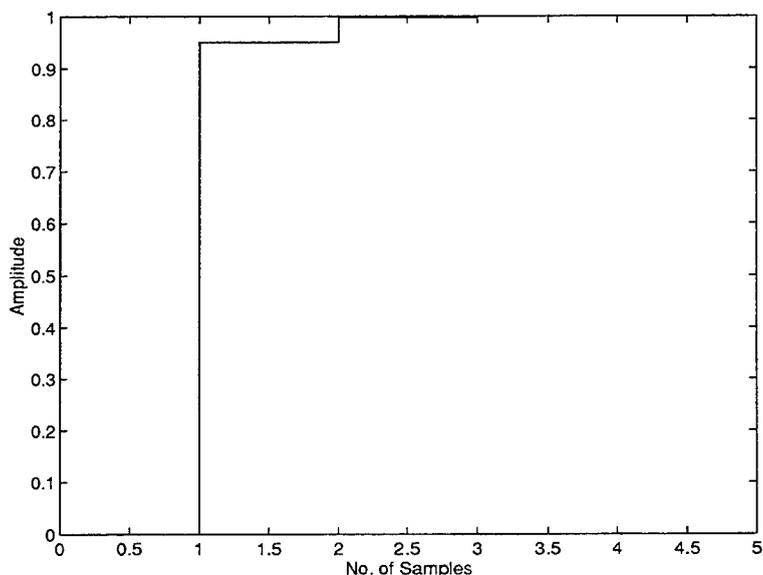


Figure 3.2. Step response of a discrete system

Notice the distinct relationship between the unit pulse response and the unit step response. If r is the sampled unit step response and h is the sampled unit pulse response, then

$$r(kT) = \sum_{j=1}^{k} h(j) \quad \text{for } k = 1, 2, \ldots, \tag{3.3}$$

where T is the sample period. This relationship implies that the faster the pulse response decays to zero, the quicker the step response reaches its steady-state value. Since $\ell_1$ optimization penalizes long pulse responses, it logically also penalizes slow unit step responses. This fact and the general relationship between the unit pulse and unit step responses are particularly important in using $\ell_1$ optimization for tracking problems.

## 3.4. Weighted and Unweighted Sensitivity Minimization

The goal of most tracking problems is to minimize the error between the commanded input and the system output. This type of problem can be posed as a sensitivity minimization problem, such as the one depicted in Figure 3.3.
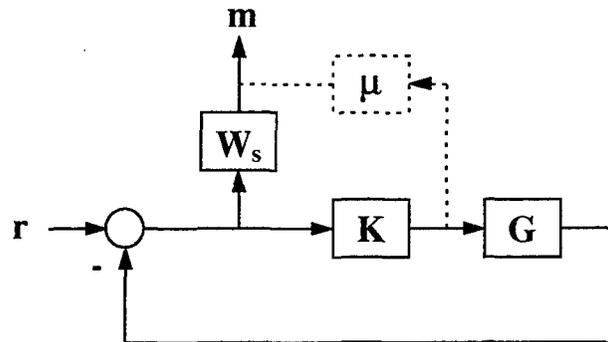


Figure 3.3. $\ell_1$ sensitivity block diagram

For the AFTI F-16 problem, the exogenous input, r, is an unknown commanded normal acceleration input with maximum magnitude less than or equal to one, and the controlled output, m, is the weighted error between the commanded acceleration and the actual aircraft acceleration. K is the unknown compensator, and a state-space description of the unweighted plant, G, is given in Appendix A.

For now, $W_s$ is set to 1 and $\ell_1$ optimization is performed on the above system. Since a weighted sensitivity problem places no penalty on control usage, a small penalty, $\mu = 1e - 5$, is added to $D_{mu}$ to ensure the left inverse of U exists. The optimal closed-loop system has a one-norm of 2.07 and the controller is 4th order. The system response to a 1g step in normal acceleration is shown in Figure 3.4. The "jags" in the response are a product of the sampled-data simulation. As the sample rate increases these "jags" become less apparent.

Notice that the step response is extremely fast. This is mainly due to the fact that there was only a small penalty placed on control usage. However, as discussed in the

previous section, unconstrained $\ell_1$ optimization tends to produce very quick step responses. Plots of control usage and rate of control usage are shown in Figures 3.5 and 3.6.
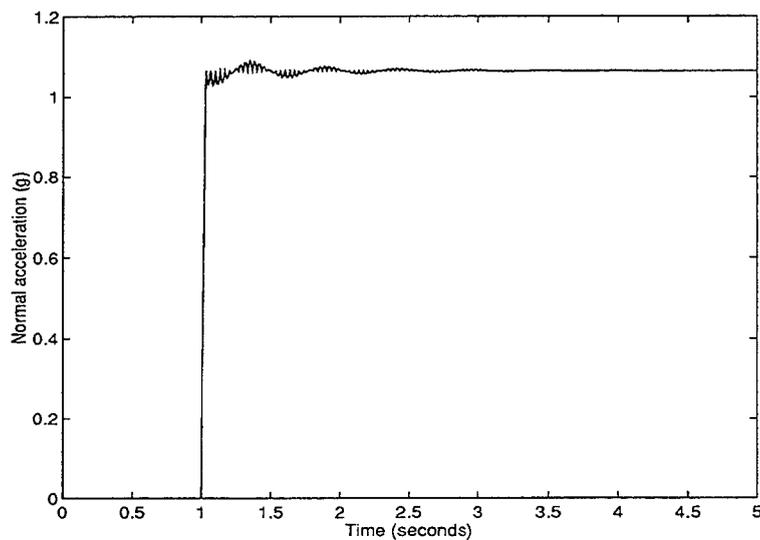


Figure 3.4. Unweighted sensitivity step response
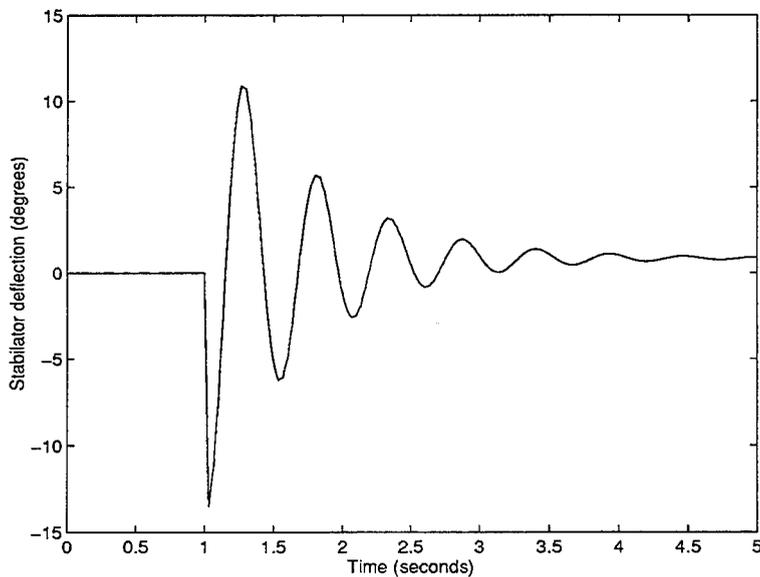


Figure 3.5. Unweighted sensitivity control usage
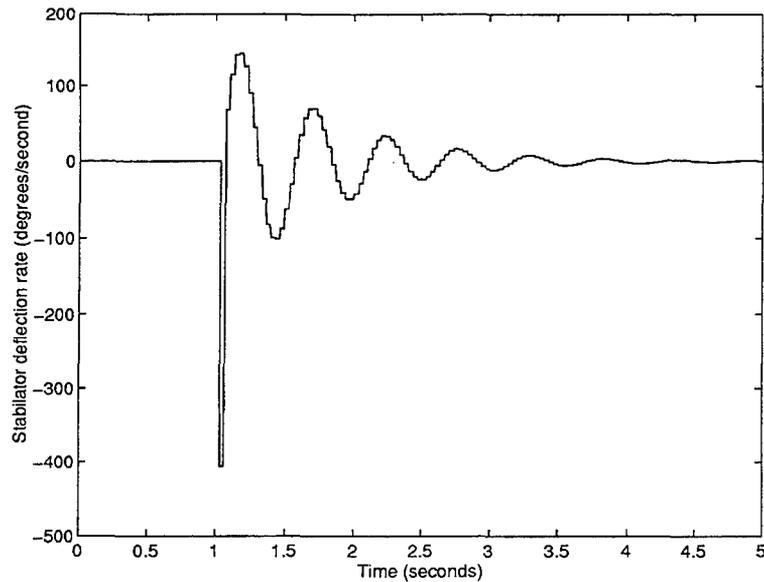
Figure 3.6. Unweighted sensitivity control rate

The control usage does not violate the maximum deflection limits, but it is quite large for only a 1g change in normal acceleration. Since the system is linear, it is easy to see that the maximum deflection limit would be violated for a commanded 2g change in normal acceleration. The control rate violates the maximum allowable rate limitation, even for a very small command. The controller found above would be undesirable for two reasons: first, the system tracks with a steady-state error; second, the level of performance shown in Figure 3.4 is unattainable by the AFTI F-16 due to limitations in the stabilator rate of deflection.

The first problem can be handled in two different ways. One option is to place a gain on the input to the system to ensure the closed-loop DC gain equals one. Another alternative is to weight the sensitivity more heavily at low frequency. This method is preferred since it can eliminate steady-state error problems to additional low frequency commands.

To demonstrate the second method, consider the following weighting:

$$W_s = \frac{s+10}{s+0.0001}.$$

(3.4)

3-8

This particular weighting represents the inverse of the desired sensitivity function and is very similar to the one used by Luke ([Luk93]) in an $H_\infty$ based design for the AFTI F-16. The optimal $\ell_1$ norm for the system using this weighting is 2.25 and the compensator is 5th order. A plot of the system response to a 1g step in normal acceleration is shown in Figure 3.7.
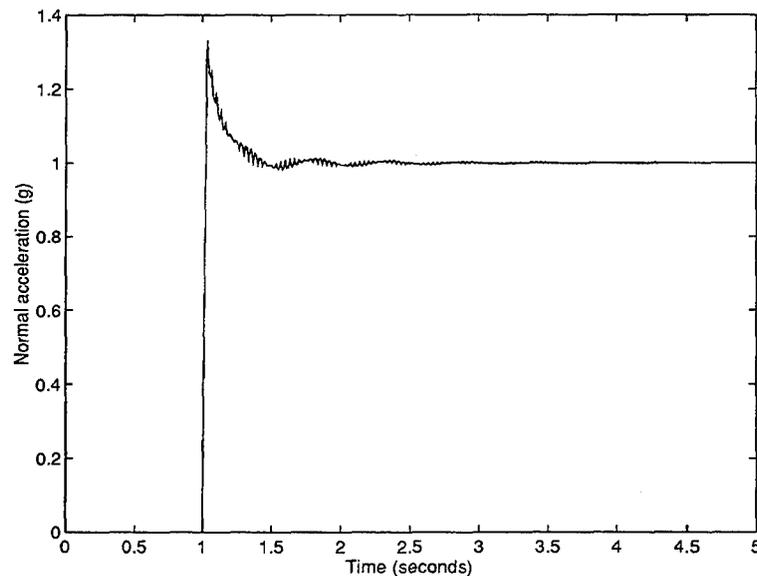


Figure 3.7. Weighted sensitivity step response

This weighting has clearly eliminated the problem of a steady-state error to a step input, but the system response is still extremely fast. Examination of the control usage and rate of control usage for this weighting, shown in Figures 3.8 and 3.9, reveals that the problem with the stabilator deflection rate still exists. In fact, this problem is worse than before.

Before discussing how to handle the problem of excessive control deflection and rates of deflection, it is important to discuss some objectives the tracking solution should achieve. The following list represents some typical factors which may be important:

i)   minimum error to low frequency commands

ii)  no violations of control defection and rate limitations
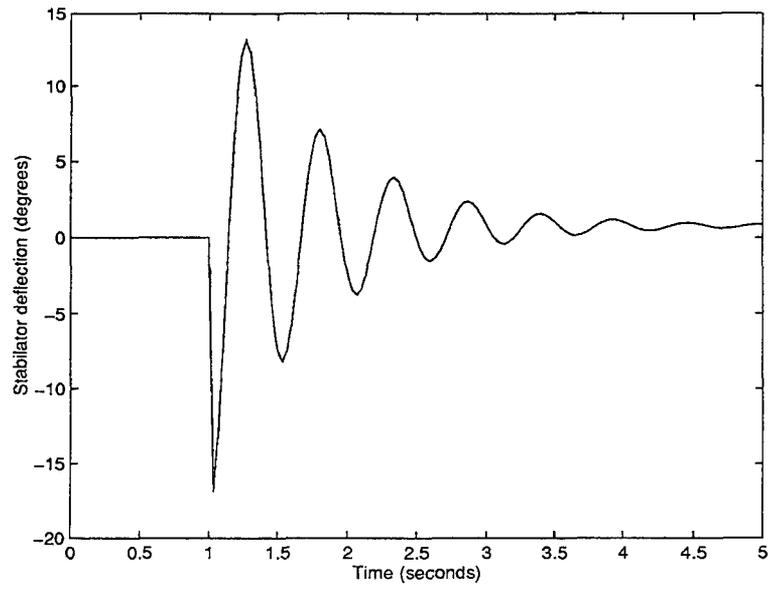
Figure 3.8. Weighted sensitivity control usage
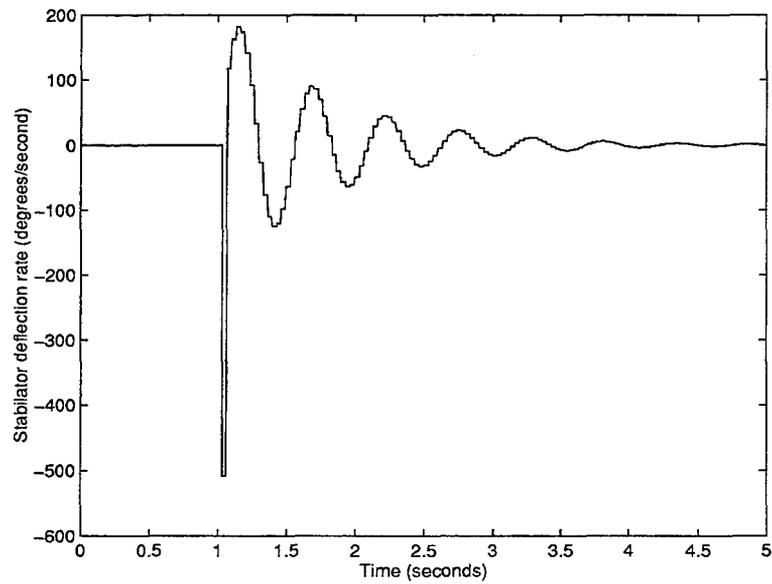


Figure 3.9. Weighted sensitivity control rate

iii) zero steady-state error to low frequency commands

iv) minimum overshoot and undershoot

v) the quickest response possible given the above.

Item i) indicates that sensitivity minimization is the proper objective function for $\ell_1$ optimization, but items ii)-iv) indicate that it must be done with certain constraints. Item v) is inherently built into $\ell_1$ optimization for most problems. To demonstrate how items i)-v) can be achieved without frequency weights on the sensitivity function, $W_s$ is set equal to one in the following sections.

The next section tackles item ii). It covers three different approaches for adding control deflection and rate constraints to the error minimization problem.

## 3.5. Control Deflection and Rate Limitations

The previous section was concerned with solving the one-block problem

$$\inf_{K \text{ stabilizing}} \|S\|_1 , \tag{3.5}$$

where S is the sensitivity function. This section is concerned with the general two-block problem

$$\inf_{K \text{ stabilizing}} \left\| \begin{array}{c} S \\ W_c KS \end{array} \right\|_1 , \tag{3.6}$$

where $W_c$ is a weighting on the control usage. The added block in (3.6) can be used to ensure that control deflection or rate limitations are not violated.

Since the control rate limitations were violated in the last section, only rate constraints are added in this section. It turns out that once the control rate is properly constrained for the AFTI F-16, the control deflection limitations are not a problem. The ideas presented below easily extend to penalizing control deflections alone or to both control rates and deflections.

3-11

In order to change the second block of (3.6) to a penalty on control rate instead of control usage, an appropriate weight must be chosen for $W_c$. Clearly the weighting must be chosen so that it effectively takes the derivative of the control signal. This problem is best handled directly in the z-domain with

$$W_c(z) = \frac{z-1}{Tz},\qquad (3.7)$$

where T is the sample period. This weighting function, known as the backward Euler transformation, calculates a finite different gradient between discrete pulses. Since the weighting is in the z domain, the continuous system must be discretized before this weight can be augmented to the problem.

The first approach to solving the rate-constrained tracking problem is to multiply each block in the two block problem by a desired level of performance. For example, if the one-norm of the first block is desired to be less than $\gamma$ and the maximum control deflection rate is $U_{rmax}$, then the problem becomes

$$\inf_{K \text{ stabilizing}} \left\| \begin{array}{c} \dfrac{1}{\gamma}S \\[2mm] \dfrac{1}{U_{rmax}}W_cKS \end{array} \right\|_1 .\qquad (3.8)$$

If the resulting one-norm of this system is less than 1, then both levels of performance are achieved. This follows from the previous assumption that the maximum magnitude of the exogenous input is less than or equal to one. If the goal is to find a solution which has the minimum achievable $\gamma$ without violating the maximum control rate, this is not the best approach because (3.8) would have to be solved iteratively for $\gamma$ until the resulting system one-norm is exactly one.

A better approach is to solve the following problem:

$$\inf_{K \text{ stabilizing}} \|S\|_1$$
$$\text{subject to } \|W_cKS\|_1 \leq U_{rmax}.\qquad (3.9)$$

Recall for multi-block problems that the one-norm of the system is the maximum absolute row sum. In (3.8), the maximum absolute row sum had to be less than one to ensure the one norm of the entire system was also less than one. In (3.9), the individual row sums are separated, with one being minimized while the other is constrained. The current version of the $l_1$ software developed by the author allows the user to specify different constraint levels for each controlled output in a multi-block column problem.

Problem (3.9) was solved for the AFTI F-16 with control rate limitations of 200 deg/sec, 100 deg/sec, and 60 deg/sec. A plot of the system unit step response for all three constraint levels is shown in Figure 3.10.
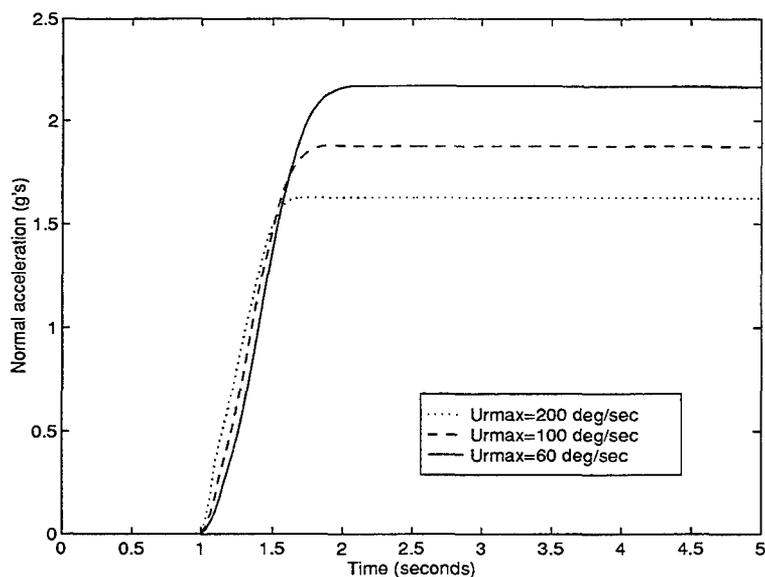


Figure 3.10. Unweighted sensitivity step responses with $l_1$ constraints on the control rate

The slowest response with the largest steady-state error corresponds to the actual stabilator deflection rate limitation of 60 deg/sec. The responses to the other two constraint levels are shown for comparison. This type of plot can also be used for design purposes since it is easy for the designer to see how much performance can be gained if

faster control actuators are obtained. Plots of the control deflections and rates are shown in Figures 3.11 and 3.12, respectively.
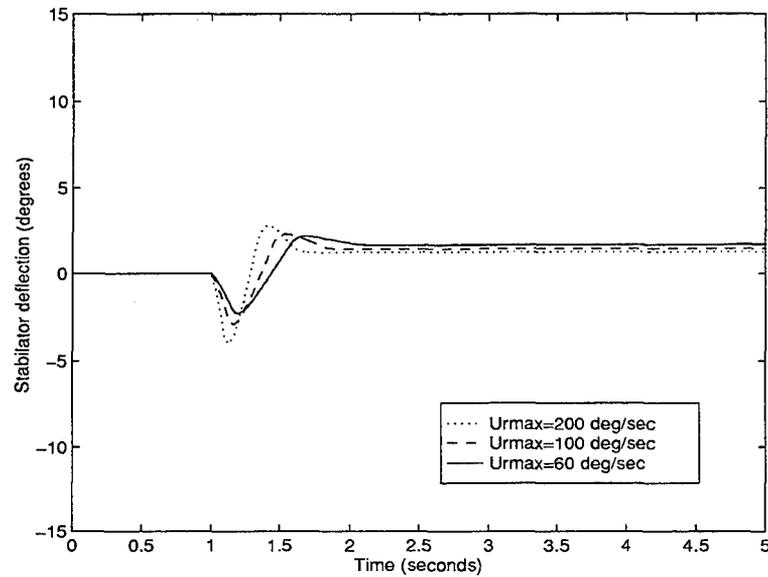


Figure 3.11. Unweighted sensitivity control usage with $\ell_1$ constraints on the control rate
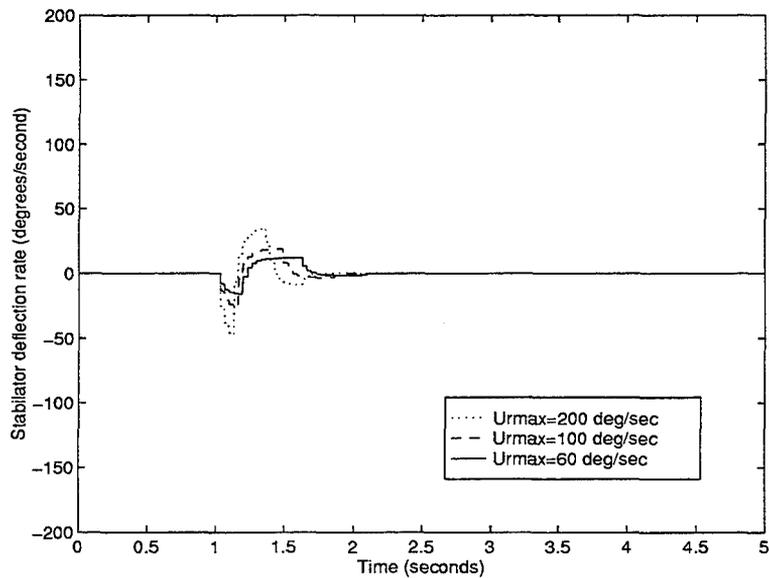


Figure 3.12. Unweighted sensitivity control rates with $\ell_1$ constraints on the control rate

Notice from Figure 3.12 that the control rates are well within their respective $\ell_1$ constraints for a unit step input. The $\ell_1$ constraints will actually ensure that the control rate limitations will not be violated for any input with a maximum magnitude less than or equal to one. In this particular example, the $\ell_1$ constraints are very conservative for a unit step input.

The one-norm of the objective and the compensator orders are shown in Table 3.1 for each constraint level.

| constraint | one-norm | controller order |
|---|---|---|
| 200 deg/sec | 2.63 | 35 |
| 100 deg/sec | 2.88 | 38 |
| 60 deg/sec | 3.17 | 43 |

Table 3.1. Comparison of different $\ell_1$ constraints on control rate

The order of the $\ell_1$ optimal compensators is directly related to the support length of the pulse response; i.e., the number of time steps it takes the pulse response to decay to zero. In fact, if the $\ell_1$ optimization software did not perform a balanced model reduction on the compensator, this relationship would be almost one-to-one. Recall that the support length of the pulse response is related to the time it takes the step response to reach steady-state. This explains why the controllers found using the above approach have such high orders.

The previous two approaches imposed $\ell_1$ constraints on the control rate. This means that the constraint limitation imposed will not be exceeded for any input into the system bounded in magnitude by one. Another possibility is to ensure that the constraint is not exceeded for a single class of input like the step command. Unlike the $\ell_1$ constraints, these $\ell_\infty$ types of constraints can only be used on a finite horizon. In other words, they can only be imposed over the support length of the solution. In many cases, however, imposing these constraints over the first few time steps is sufficient.

To understand how a constraint on the step response of the system can be imposed in $\ell_1$ optimization, recall the relationship in (3.3) between the unit pulse and unit step response. The step response at any particular time step is nothing more than the sum of the pulse response at that time step plus all previous time steps. Therefore, in terms of the pulse response at each time step, these constraints can be imposed with very simple Toeplitz matrices, with ones below the main diagonal and zeros above, such as

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \hat{\Phi}(0) \\ \hat{\Phi}(1) \\ \vdots \\ \hat{\Phi}(N) \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} U_{rmax}. \tag{3.10}$$

Notice that these constraints can easily be augmented to the interpolation conditions in (2.41).

The new problem with the augmented step input constraints becomes

$$\inf_{K \text{ stabilizing}} \|S\|_1 \tag{3.11}$$
$$\text{subject to } \|W_u K S w_f\|_\infty \leq U_{rmax},$$

where $w_f$ is a unit step. A plot of the AFTI F-16 normal acceleration step response for control rate constraint levels of 200 deg/sec, 100 deg/sec and 60 deg/sec is shown in Figure 3.13.

Again, the slowest response with the largest steady-state error corresponds to a constraint of 60 deg/sec. Notice that the step responses to this type of constraint are much quicker and have less steady-state error than the $\ell_1$ constraints. Control deflections are shown in Figure 3.14 and control rates are shown in Figure 3.15.

The one-norm of the objective and the compensator orders are shown in Table 3.2 for each constraint level. With this approach, quicker settling times also lead to lower order controllers.
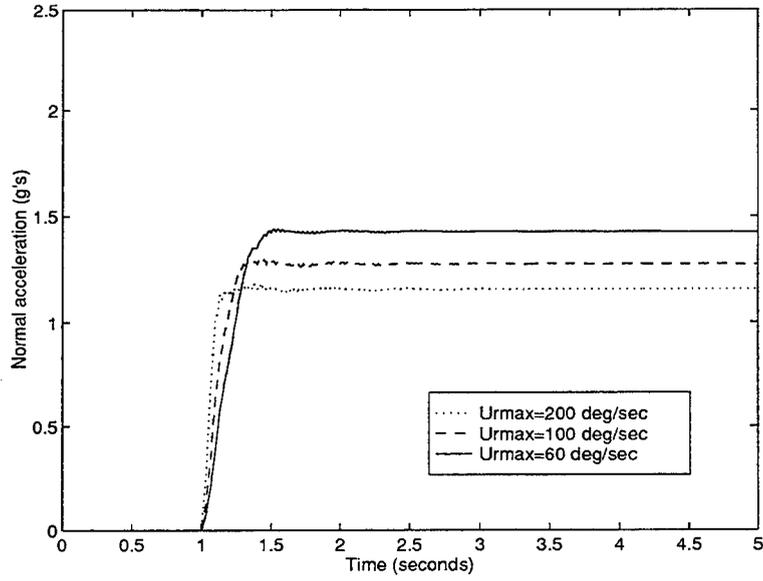
Figure 3.13. Unweighted sensitivity step responses with $\ell_\infty$ constraints on the control rate



Figure 3.14. Unweighted sensitivity control usage with $\ell_\infty$ constraints on the control rate

3-17
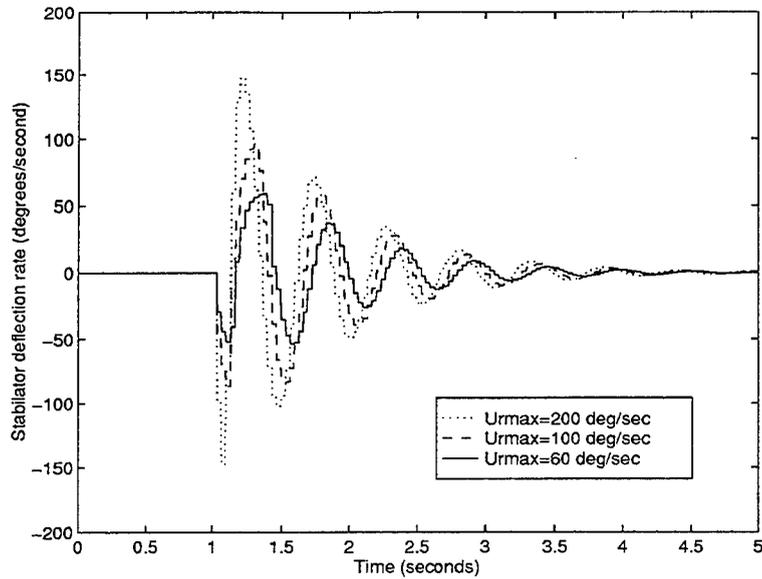
Figure 3.15. Unweighted sensitivity control rates with $l_\infty$ constraints on the control rate

| constraint | one-norm | controller order |
|---|---|---|
| 200 deg/sec | 2.16 | 9 |
| 100 deg/sec | 2.28 | 13 |
| 60 deg/sec | 2.43 | 19 |

Table 3.2. Comparison of different $l_\infty$ constraints on control rate

While all the step responses in this section meet some constraint level on the control rate, none of them has zero steady-state error. This issue is addressed in the next section.

## 3.6. Steady-State Error and Time-Varying Exponential Weights

Zero steady-state error to a step input can be enforced with an added equality constraint to the $l_1$ optimization problem. Recall from (3.3) that the final value of the step response is simply the summation of the unit pulse response over its entire support length.

3-18

Therefore, zero steady-state error to a unit step input is guaranteed if the sum of the sampled unit pulse response equals zero. This is not an absolute summation like the norm calculation; it is simply a summation of the pulse response at each time step. The added equality constraint takes the form

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \hat{\Phi}(0) \\ \hat{\Phi}(1) \\ \vdots \\ \hat{\Phi}(N) \end{bmatrix} = 0. \tag{3.12}$$

This constraint was added to the problem presented in (3.11), with the control rate constraint equal to 60 deg/sec. The resulting solution has an objective one-norm of 3.00 and the compensator is 44th order. The system response to a 1g normal acceleration step input is shown in Figure 3.16 and the control rate is shown in Figure 3.17.
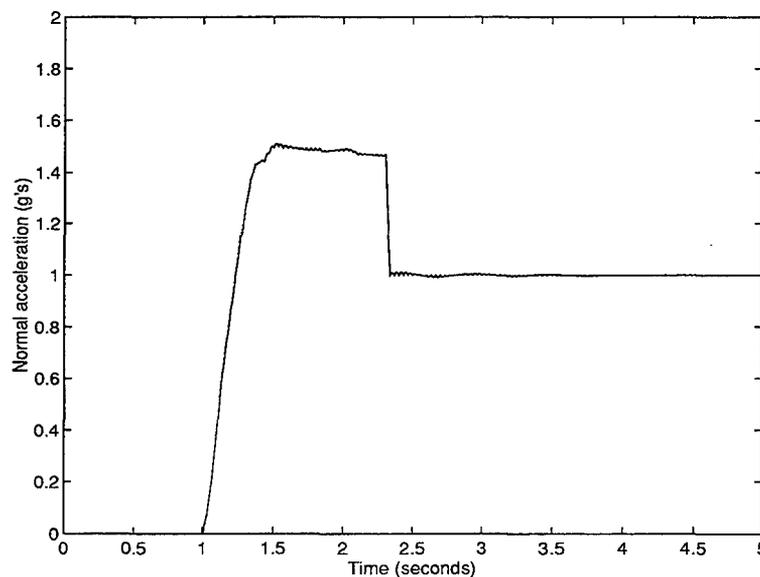


Figure 3.16. Unweighted sensitivity step response with steady-state error and $\ell_\infty$ control rate constraints
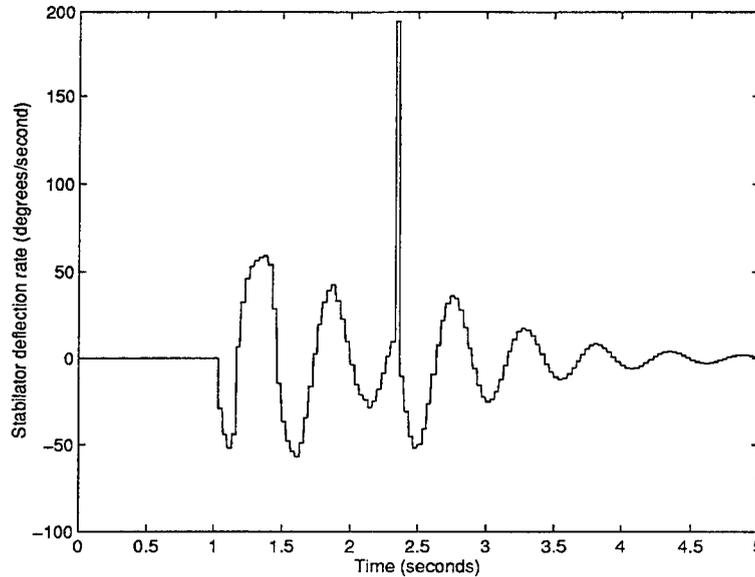
Figure 3.17. Unweighted sensitivity control rate with steady-state error and $\ell_\infty$ control rate constraints

Clearly, zero steady-state error is achieved, but the limit on control rate is not met at the nearly discontinuous jump just after two seconds. Notice that the response in Figure 3.16 is exactly the same as its counterpart in Figure 3.13 up until this jump. This happened because the steady-state error equality constraint was not imposed until the very last time step in the support length. In this system, imposing the constraint any earlier results in a larger one-norm, which the optimization rejects.

This problem can be overcome with time-varying exponential weights on the norm calculation. Consider multiplying each sampled pulse response by $a^{kT}$, where k is the sample index, T is the sample period and $a > 1$. Since this weight gets larger as k gets larger, it effectively penalizes late errors over early ones.

The system above was rerun with a time-varying exponential weight added to the norm calculation. With $a = 1.1$, the objective one-norm was 3.59 and the controller was 26th order. A plot of the system response to a step input is shown in Figure 3.18 and the control rate is shown in Figure 3.19.

3-20
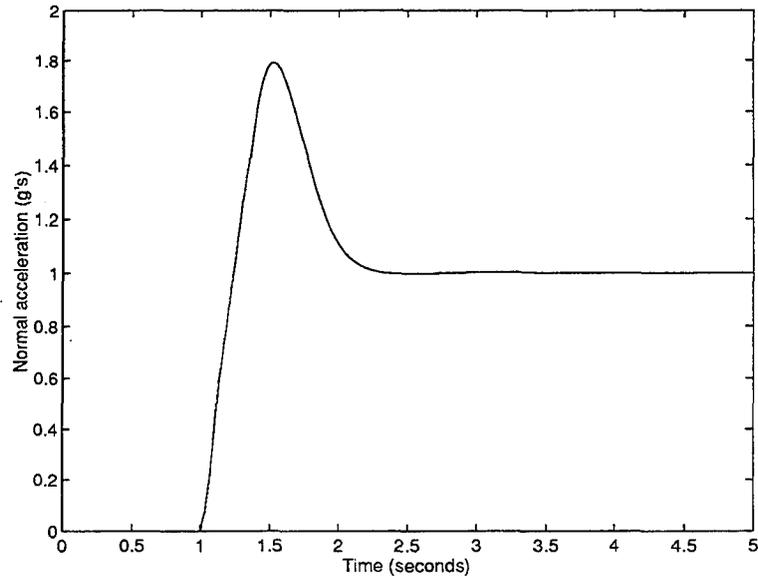
Figure 3.18. Unweighted sensitivity step response with steady-state error/$\ell_\infty$ control rate constraints and time-varying exponential weights
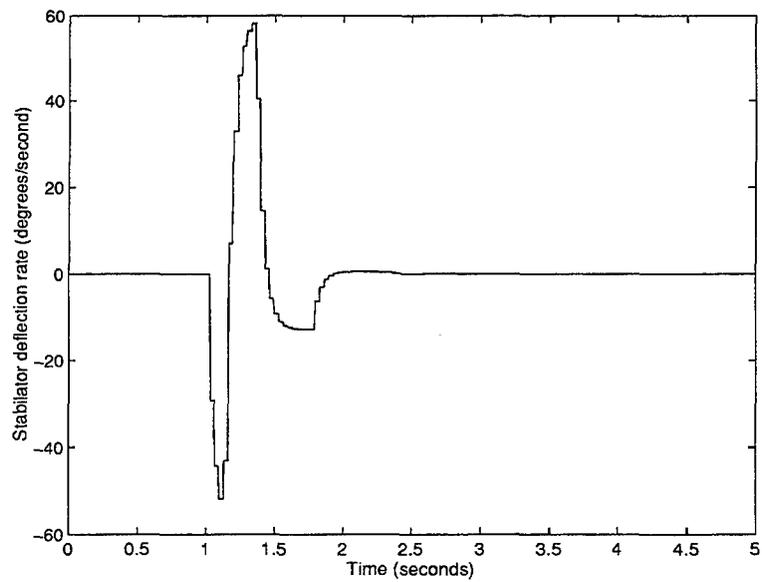


Figure 3.19. Unweighted sensitivity control rate with steady-state error/$\ell_\infty$ control rate constraints and time-varying exponential weights

Adding the exponential weights increased the one-norm as expected. The weighting also decreased the settling time and thus the controller order.

The step response at this point now meets all the criteria established in Section 3.4, except the overshoot issue. This problem is discussed in the next section.

### 3.7.    *Overshoot and Undershoot Limitations*

Problems with excessive overshoot and undershoot in the step response can be handled in exactly the same manner as excessive control deflections and rate violations. To demonstrate this concept, a very small reduction is done on the overshoot for the system response shown in Figure 3.18. Theoretically, the overshoot can be reduced to any desired level at the expense of a slower response. However, it was extremely difficult to find a solution for the problem presented with the current $\ell_1$ optimization software. The multi-block problem contains so many constraints and delays that calculation attempts alone are extremely expensive in terms of computer time. Further, the linear programming routine in the software has difficulty solving very large systems of equations, possibly due to scaling problems. Both of these issues are currently being addressed by researchers at MIT.

The system response in Figure 3.18 has an overshoot of about 80%. An $\ell_\infty$ type constraint on the overshoot was added to the problem to ensure that the overshoot would be less than 70% to a step input. The resulting solution had an objective one-norm of 3.40 and the controller was 28th order. A plot of the step response with the added constraint is shown in Figure 3.20.

This step response is still less than ideal; however, all the tools to shape and constrain the response are now available. As the $\ell_1$ optimization software becomes more reliable and efficient, a designer should be able to use all the techniques presented up to this point to find a compensator which meets all of his or her tracking requirements. Unfortunately, this compensator will probably be extremely high order and therefore

impractical to use. The next section on model matching offers one way to use $\ell_1$ optimization and still produce controllers at or about the order of the original discrete system.
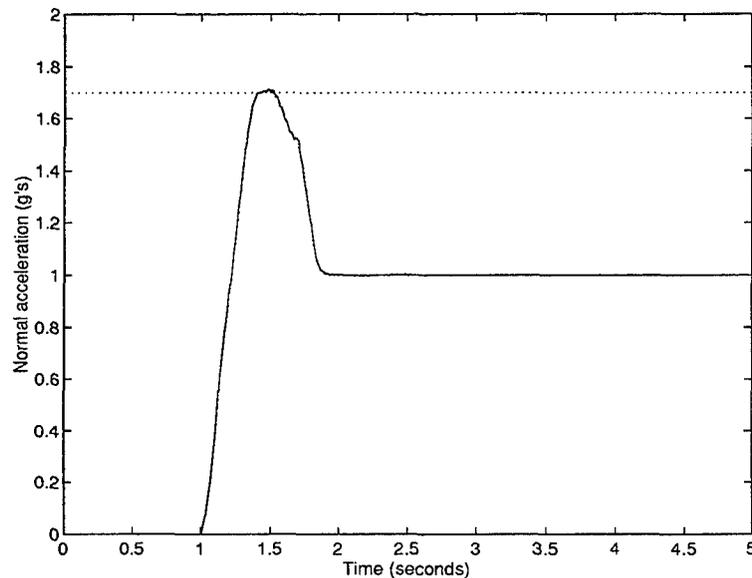


Figure 3.20. Unweighted sensitivity with control rate/steady-state error/overshoot constraints and time-varying exponential weights

## 3.8. Model Matching

Using $\ell_1$ optimization to solve multi-block systems generally produces high order controllers, especially when additional constraints are imposed on the system. One way to counter this problem is to solve a one-block system instead. Since these systems can be solved exactly, without delay augmentation, the resulting controllers tend to be much smaller (usually about the order of the unweighted plant). An added benefit to using one-block systems is that they can be solved much faster and more reliably than multi-block systems with the current $\ell_1$ optimization software.

In many cases, the constraints discussed in the previous sections can not be imposed with one-block systems. Therefore, a one-block system must be chosen that

incorporates as many of the design criteria required for good tracking as possible. One way to accomplish this objective is to model match the design problem to a system which has the desired tracking characteristics.

Two examples of model-matching designs are presented in this section. The first example involves the SISO AFTI F-16 problem that has been discussed throughout this chapter. The second example involves a MIMO missile problem. A small penalty on control usage, similar to the one discussed in Section 3.4, was added to each example to make the resulting $\ell_1$ problem nonsingular.

### 3.8.1. SISO Example

Consider the closed-loop model matching system shown in Figure 3.21, where H is the ideal closed-loop model given in continuous time by

$$H(s) = \frac{4}{s+4} \tag{3.13}$$
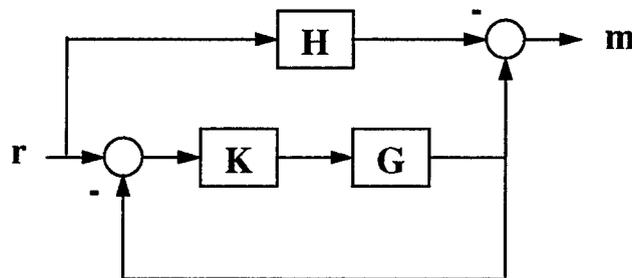


Figure 3.21. Closed-loop model matching diagram

This particular closed-loop model was chosen because its step response is relatively quick, has no overshoot, and no steady-state error. A plot of the step response of this model is shown in Figure 3.22.

The one-norm of the solution to this design problem is 0.37 and the compensator is 5th order.   A plot of the AFTI F-16 step response to a commanded 1g normal acceleration change is shown in Figure 3.23.
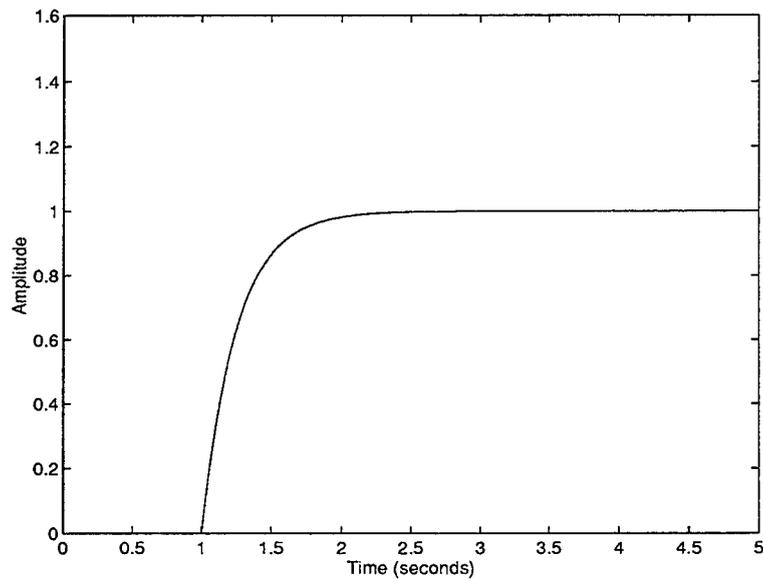


Figure 3.22.  Ideal model step response
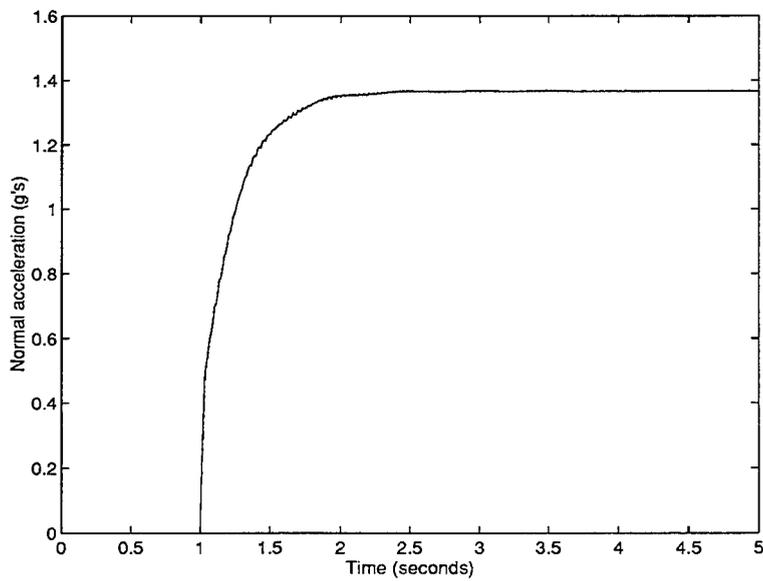


Figure 3.23.  SISO model matching, step response

The step response of the solution is approximately 0.37 g's larger than the step response of the ideal model at steady-state. To make the system response match the ideal model's response, the commanded normal acceleration has to be multiplied by a gain. This gain is the reciprocal of the DC gain of the discrete closed-loop transfer function,

$$T_{yr}(z) = \frac{K(z)G(z)}{1 + K(z)G(z)}.$$ (3.14)

For this problem, the gain equals 0.73.

Notice that with this particular approach there is no direct way to ensure the above solution will not violate control deflection and rate limitations. In this problem, the control rate limits were violated in the first few time steps. However, the closed-loop system still performs well if the step input is first passed through a continuous prefilter equal to

$$F(s) = \frac{10}{s + 10},$$ (3.15)

and a rate limiter is added to the control signal. With the added prefilter, the system actually sees an input which is similar to the response in Figure 3.22 (just slightly faster), instead of a discontinuous step input. The new input is actually a more realistic representation of a pilot command.

The system was tested with the prefilter, gain adjustments on the input, and a control rate limiter set at $\pm 60$ deg/sec. The response is shown in Figure 3.24. This response has no overshoot, no steady-state error and was achieved with a 5th order controller and a small gain on the input.

Plots of the sensitivity and complementary sensitivity for the resulting closed-loop system are shown in Figures 3.25 and 3.26. The vector gain margins (VGM) and phase margins (VPM) of this system are

$$-11.4 \text{ dB} \le \text{VGM} \le 10.4 \text{ dB} \quad \text{VPM} = \pm 42.9°.$$
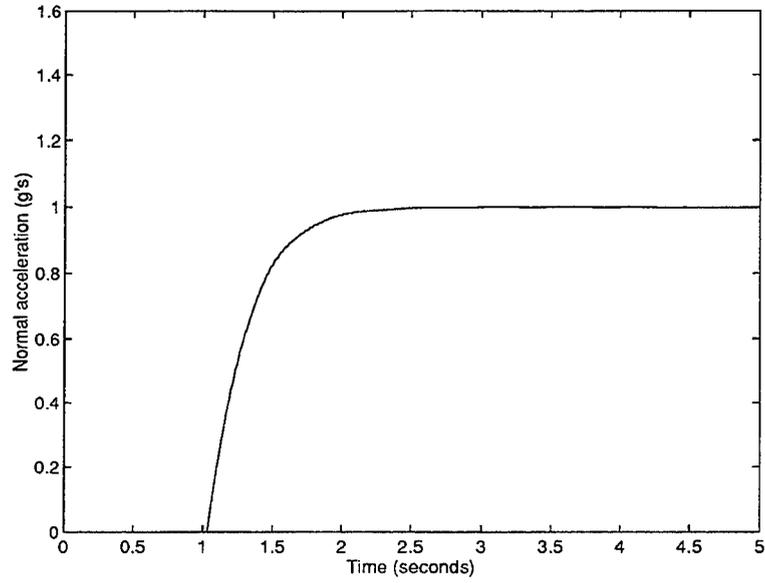
Figure 3.24. SISO model matching, step response with scaling factor, prefilter and control rate limiter
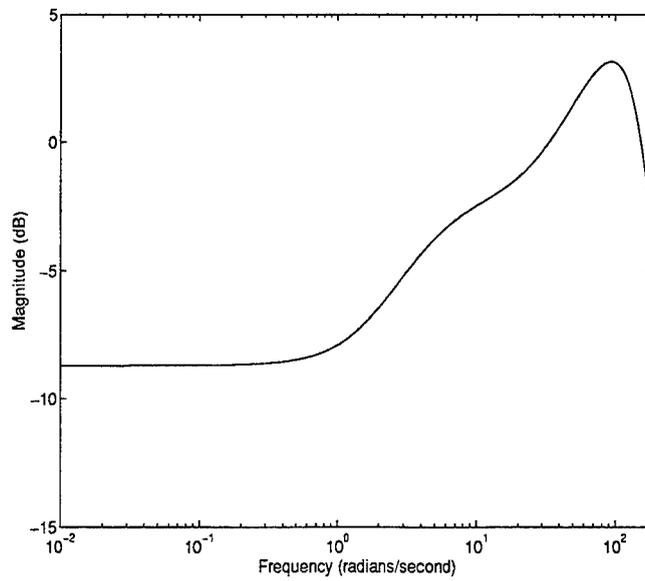


Figure 3.25. SISO model matching, sensitivity
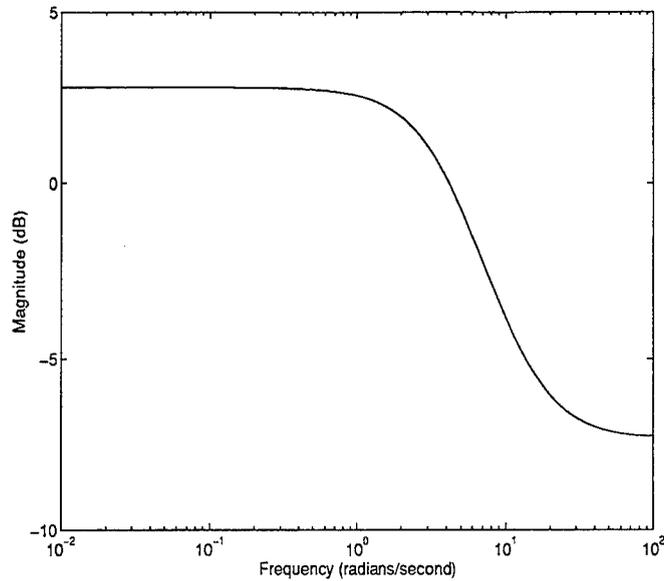
Figure 3.26. SISO model matching, complementary sensitivity

### 3.8.2. MIMO Example

This model matching example is done on the 3 input, 3 output missile system described in Appendix B. The goal of this design problem is to command normal or lateral acceleration while keeping the roll-rate of the missile as small as possible. When commanding normal acceleration, it is also desired to minimize the lateral acceleration and vice versa when commanding lateral acceleration.

An open-loop model matching design is used to solve this problem, as shown in Figure 3.27.
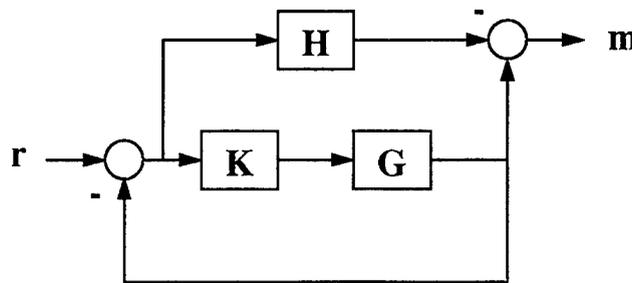


Figure 3.27. Open-loop model matching diagram

In this problem, the ideal open-loop model is given in continuous time by

$$H(s) = \begin{bmatrix} \dfrac{10}{s+0.001} & 0 & 0 \\ 0 & \dfrac{10}{s+0.001} & 0 \\ 0 & 0 & \dfrac{10}{s+0.001} \end{bmatrix}. \qquad (3.15)$$

This model was chosen with diagonal elements only in order to decouple the normal acceleration, lateral acceleration and roll-rate responses. The value 10 in the numerators was chosen to determine the speed of the responses. Poles were placed at -0.001 in each transfer function to provide near integral action, and ensure very little steady-state error.

The one-norm of the solution to this problem is 1.009 and the controller is 11th order. To avoid control rate violations, the system was tested with rate limiters set at the maximum allowable rate, 750 degrees/second. The control deflection limits (38 degrees) were not violated in any case. The system responses to a 10g normal acceleration step input (applied at 0.5 seconds) are shown in Figures 3.30 and 3.31. The system responses to a 5g lateral acceleration step input are shown in Figures 3.32 and 3.33. Closed-loop sensitivity and complementary sensitivity plots are shown in Figures 3.28 and 3.29, respectively. The vector gain margins (VGM) and phase margins (VPM) of this system are

$$-74.0 \text{ dB} \leq \text{VGM} \leq 11.2 \text{ dB} \quad \text{VPM} = \pm 60.0^\circ.$$

Notice that the open-loop model match in this example eliminates the need for the DC gain adjustment used in the SISO example. This approach, however, lead to an unstable system in the AFTI F-16 problem once rate limiters were added to the simulation model. While neither model matching method is guaranteed to produce an acceptable controller, both offer easy design alternatives to the methods presented in Sections 3.4-3.7.
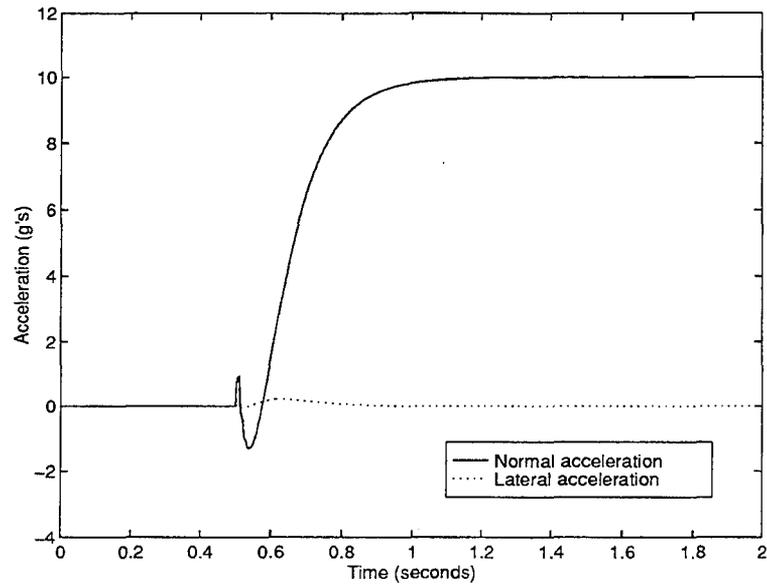
Figure 3.30. MIMO model matching, acceleration responses to a 10g step in normal acceleration
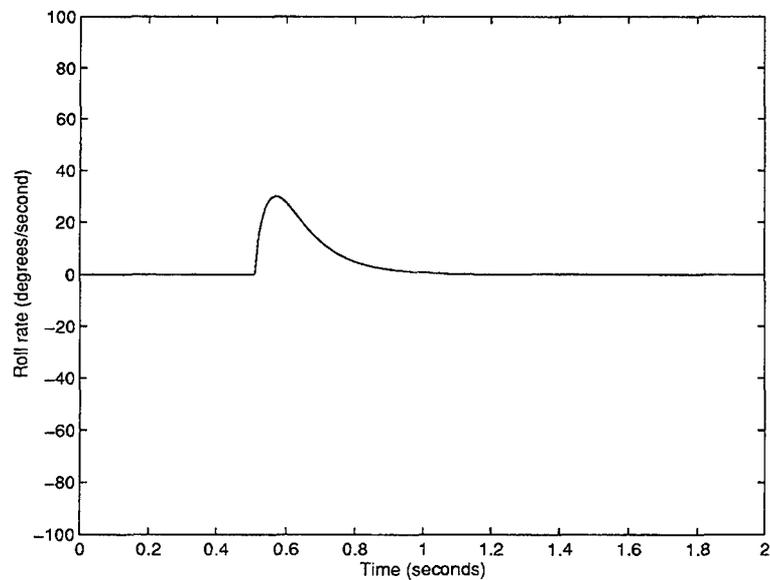


Figure 3.31. MIMO model matching, roll-rate response to a 10g step in normal acceleration
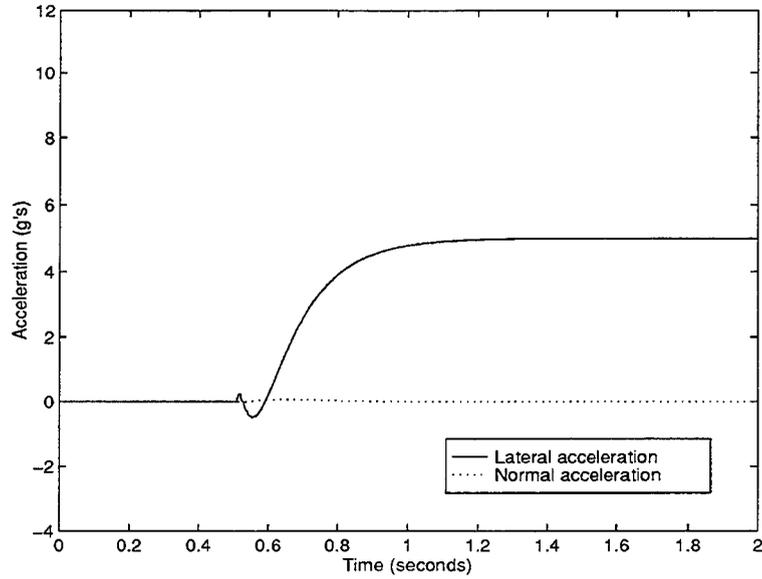
Figure 3.32. MIMO model matching, acceleration responses to a 5g step in lateral acceleration
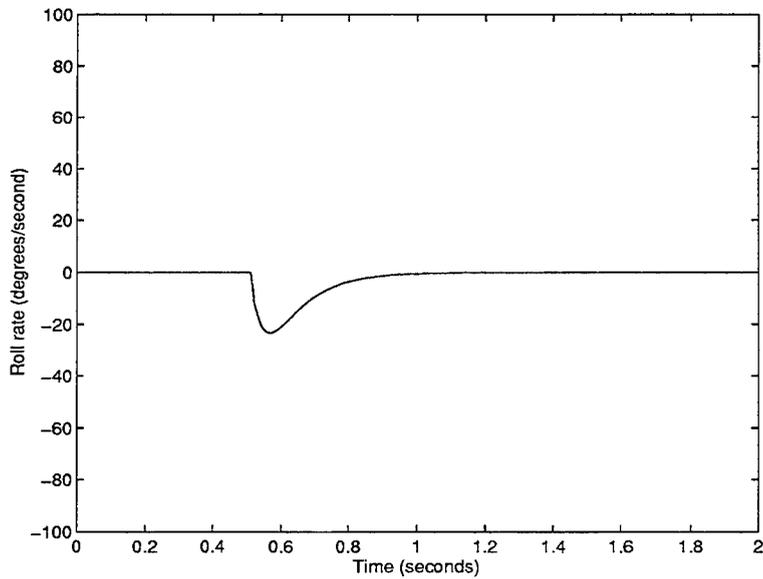


Figure 3.33. MIMO model matching, roll-rate response to 5g step in lateral acceleration
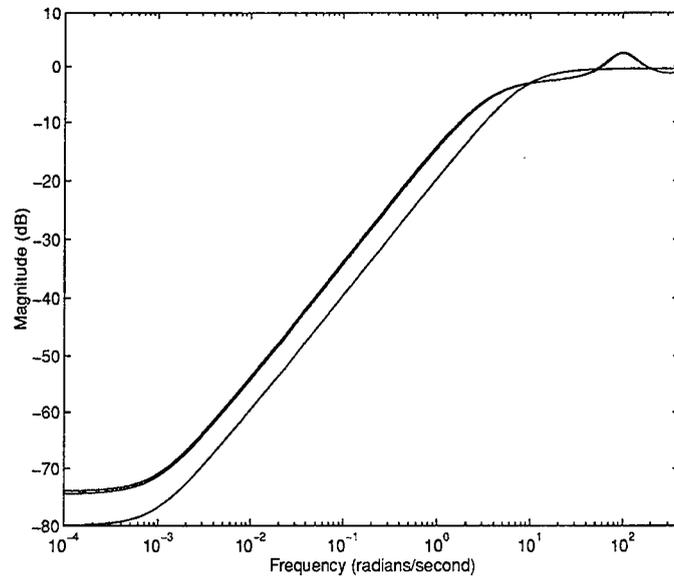
Figure 3.34.  MIMO model matching, sensitivity
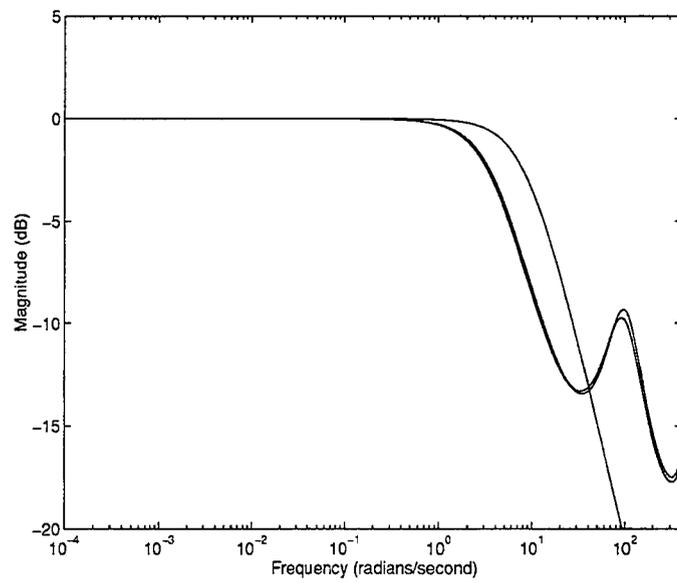


Figure 3.35.  MIMO model matching, complementary sensitivity

## 3.9. *Chapter Summary*

This chapter presented methods of using $\ell_1$ optimization to solve tracking problems. Several constraints that could be added to the standard $\ell_1$ problem to handle control deflection and rate limitations, zero steady-state error requirements, and overshoot limitations were discussed. Unfortunately, the constrained $\ell_1$ optimization problem often produced high order controllers. To counter this problem, two different model matching techniques were presented which can produce acceptable tracking results with lower order controllers.

The next design issue to be considered is system noise performance, which is best handled with $H_2$ optimization. To achieve both tracking and noise performance, one might suggest using a mix of $H_2$ and $\ell_1$ optimization techniques. Indeed, this is a viable approach. Mixed $H_2/L_1$ optimization for continuous time systems is discuss in Chapter 4 and mixed $H_2/\ell_1$ optimization for discrete time systems is discussed in Chapter 5. In both cases, the order of the resulting compensator is chosen by the designer which eliminates the problem of optimal solutions with excessively large compensator orders.

# IV. Mixed $H_2/L_1$ Optimization

## 4.1. Chapter Overview

This chapter develops a fixed-order, mixed $H_2/L_1$ optimization method to design controllers for continuous systems. Section 4.2 discusses the general mixed $H_2/L_1$ optimization problem. In Section 4.3, analytical derivatives for the $H_2$ portion of the problem are derived. Two methods of numerically approximating the $L_1$ norm of a transfer function are presented in Section 4.4. Analytical gradients for the $L_1$ portion of the problem and a stability constraint are discussed in Section 4.5. Section 4.6 covers the computer implementation of the $H_2/L_1$ problem. The last section in the chapter demonstrates $H_2/L_1$ optimization using a continuous version of the AFTI F-16 longitudinal model.

## 4.2. The Mixed $H_2/L_1$ Problem

The mixed $H_2/L_1$ design problem is depicted in Figure 4.1.
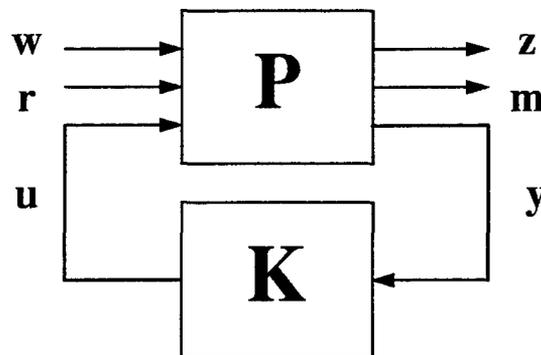


Figure 4.1. Mixed $H_2/L_1$ problem

In general, it is assumed that there is no relationship between w and r, or z and m.

The goal of mixed $H_2/L_1$ optimization is to find a stabilizing controller which satisfies

$$\inf_{K \text{ stabilizing}} \|T_{zw}\|_2 \text{ subject to } \|T_{mr}\|_1 \le \nu, \tag{4.1}$$

where

$$T_{zw} = \left[\begin{array}{c|c} A_2 & B_w \\ \hline C_z & D_{zw} \end{array}\right], \tag{4.2}$$

$$T_{mr} = \left[\begin{array}{c|c} A_1 & B_r \\ \hline C_m & D_{mr} \end{array}\right] \tag{4.3}$$

and $\nu$ is a user specified constraint level on the $L_1$ norm of $T_{mr}$.

The state-space of P is found by augmenting the stable weights of the $H_2$ problem and the $L_1$ problem to the original system. Typically, the orders of the individual $H_2$ and $L_1$ problems are less than the order of P. The state-space equations of the $H_2$ and $L_1$ problems can be written as

$$\dot{x}_2 = A_2 x_2 + B_w w + B_{u2} u$$
$$z = C_z x_2 + D_{zw} w + D_{zu} u$$
$$y = C_{y2} x_2 + D_{yw} w + D_{yu} u \tag{4.4}$$

$$\dot{x}_1 = A_1 x_1 + B_r r + B_{u1} u$$
$$m = C_m x_1 + D_{mr} r + D_{mu} u$$
$$y = C_{y1} x_1 + D_{yr} r + D_{yu} u, \tag{4.5}$$

where $x_2$ is the state vector for the underlying $H_2$ problem, and $x_1$ is the state vector of the underlying $L_1$ problem. The following assumptions are now made on the state-space elements in (4.4) and (4.5):

i) $D_{zw} = 0$

ii) $D_{yu} = 0$

iii) $(A_2, B_{u2})$ is stabilizable and $(C_{y2}, A_2)$ is detectable

iv) $D_{zu}^T D_{zu}$ and $D_{yw} D_{yw}^T$ have full rank

v) $\begin{bmatrix} A_2 - j\omega I & B_{u2} \\ C_z & D_{zu} \end{bmatrix}$ has full column rank for all $\omega$

4-2

vi) $\begin{bmatrix} A_2 - j\omega I & B_w \\ C_{y2} & D_{yw} \end{bmatrix}$ has full row rank for all $\omega$

vii) $(A_1, B_{u1})$ is stabilizable and $(C_{y1}, A_1)$ is detectable

Reasons for assumptions i)-vi) were given in Section 2.8. Assumption vii) is required to ensure the $L_1$ problem has a solution.

The controller state-space equations are

$$\dot{x}_k = A_k x_k + B_k y$$
$$u = C_k x_k + D_k y. \tag{4.6}$$

Using (4.4) and (4.6), the closed-loop state-space equations for $T_{zw}$ can be written as

$$\dot{x}_2 = (A_2 + B_{u2} D_k C_{y2}) x_2 + B_{u2} C_k x_c + (B_w + B_{u2} D_k D_{yw}) w$$
$$\dot{x}_k = B_k C_{y2} x_2 + A_k x_k + B_k D_{yw} w$$
$$z = (C_z + D_{zu} D_k C_{y2}) x_2 + D_{zu} C_k x_k + D_{zu} D_k D_{yw} w. \tag{4.7}$$

Notice that $D_{zu} D_k D_{yw}$ in (4.7) must be zero to ensure the resulting two-norm of $T_{zw}$ is finite. This fact and assumption iv) imply that $D_k$ must be zero. Therefore, a strictly proper controller, K, can be assumed without any loss of generality. With this additional assumption, the state-space equations for the mixed $H_2/L_1$ problem can be written as

$$\dot{x}_2 = A_2 x_2 + B_w w$$
$$z = C_z x_2$$
$$\dot{x}_1 = A_1 x_1 + B_r r$$
$$m = C_m x_1 + D_{mr} r, \tag{4.8}$$

where

$$x_2 = \begin{bmatrix} x_2 \\ x_k \end{bmatrix} \tag{4.9}$$

$$x_1 = \begin{bmatrix} x_1 \\ x_k \end{bmatrix} \tag{4.10}$$

$$A_2 = \begin{bmatrix} A_2 & B_{u2} C_k \\ B_k C_{y2} & A_k \end{bmatrix} \tag{4.11}$$

$$\mathbf{A_1} = \begin{bmatrix} A_1 & B_{u1}C_k \\ B_kC_{y1} & A_k \end{bmatrix} \tag{4.12}$$

$$\mathbf{B_w} = \begin{bmatrix} B_w \\ B_kD_{yw} \end{bmatrix} \tag{4.13}$$

$$\mathbf{B_r} = \begin{bmatrix} B_r \\ B_kD_{yr} \end{bmatrix} \tag{4.14}$$

$$\mathbf{C_z} = \begin{bmatrix} C_z & D_{zu}C_k \end{bmatrix} \tag{4.15}$$

$$\mathbf{C_m} = \begin{bmatrix} C_m & D_{mu}C_k \end{bmatrix} \tag{4.16}$$

$$\mathbf{D_{mr}} = D_{mr}. \tag{4.17}$$

The following definitions are used to discuss the solution to the mixed $H_2/L_1$ problem:

$$\underline{\alpha} \equiv \inf_{K \text{ admissible}} \|T_{zw}\|_2$$

$$\underline{\nu} \equiv \inf_{K \text{ admissible}} \|T_{mr}\|_1$$

$$K_{2\,opt} \equiv \text{the unique K that makes } \|T_{zw}\|_2 = \underline{\alpha}$$

$$K_{1\,opt} \equiv \text{a K that makes } \|T_{mr}\|_1 = \underline{\nu}$$

$$\overline{\alpha} \equiv \|T_{zw}\|_2 \text{ when } K = K_{1\,opt}$$

$$\overline{\nu} \equiv \|T_{mr}\|_1 \text{ when } K = K_{2\,opt}$$

$$K_{mix} \equiv \text{the global minimum solution to the } H_2/L_1 \text{ problem for some } \nu$$

$$\alpha^* \equiv \|T_{zw}\|_2 \text{ when } K = K_{mix}$$

$$\nu^* \equiv \|T_{mr}\|_1 \text{ when } K = K_{mix} \tag{4.18}$$

Admissible controllers must be stabilizing and have a fixed, user-specified order.

A solution to the $H_2/L_1$ problem must satisfy the Kuhn-Tucker necessary conditions:

i) $K_{mix}$ must be feasible, i.e. stabilize $T_{zw}$ and $T_{mr}$

ii) $\nabla\|T_{zw}\|_2 + \lambda_1\nabla\|T_{mr}\|_1 = 0, \quad \lambda_1 \geq 0$

iii) $\lambda_1(\|T_{mr}\|_1 - \nu) = 0, \lambda_1 \geq 0$

where $\lambda_1$ is a Lagrange multiplier associated with the one-norm constraint. Condition i) is simply a feasibility condition. Condition ii) states that the gradient of the objective must be balanced by the scaled gradient of the constraint. The last condition states that if the constraint is not satisfied exactly, then $\lambda_1$ must be zero. Conditions ii) and iii) together imply that an optimal solution (if it exists) must lie on the constraint boundary when $\underline{v} \le v \le \overline{v}$. If $v \ge \overline{v}$, then the unique solution is $K_{2\,opt}$. By definition, $v$ can not be chosen less than $\underline{v}$.

When $\underline{v} \le v \le \overline{v}$, $\alpha^*$ is a monotonically decreasing function of $v$. An illustration of a possible $H_2/L_1$ solution curve is shown in Figure 4.2.
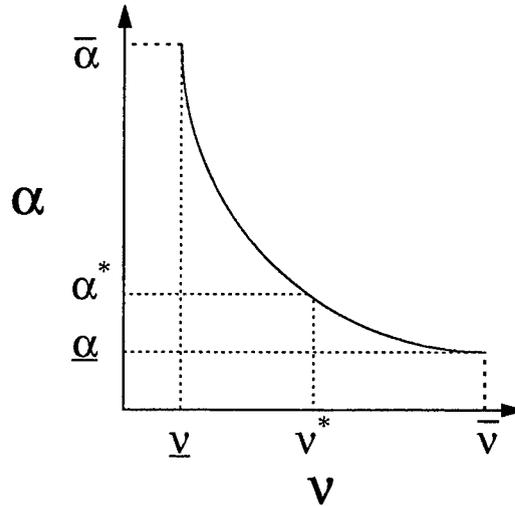


Figure 4.2. Mixed $H_2/L_1$ solution boundary

A sequential quadratic programming (SQP) algorithm is used to solve the $H_2/L_1$ problem numerically. The objective (f) and the constraints (g) are given by

$$f(c) = \varsigma_2 \|T_{zw}\|_2^2$$
$$g_1(c) = \varsigma_1 \left( \|T_{mr}\|_1 - v \right)$$
$$g_s(c) = \varsigma_s \left\{ \max_i (\text{Re}(\lambda_i (A_2))) \right\}, \tag{4.19}$$

4-5

where c is a vectorized compensator, $\varsigma$'s are scaling parameters, and $\lambda_i(\cdot)$ is the ith eigenvalue of $(\cdot)$. The stability constraint, $g_s$, is added to the problem to keep the SQP algorithm from getting lost in the unstable region. While this constraint should be posed as a strict inequality constraint, equality must also be allowed to incorporate it into the SQP algorithm. Modal form is assumed for the controller, K, to minimize the number of design variables in c. While this approach disallows repeated eigenvalues in the controller, it has been shown to be sufficient in practice.

The SQP algorithm requires gradients for the objective function and each of the constraints. Analytical gradients for the objective function are derived in the next section.

## 4.3. Gradients of the Two-Norm

Recall from Section 2.8 that the two-norm squared of $T_{zw}$ is given by

$$\|T_{zw}\|_2^2 = \text{tr}[QC_z^T C_z], \qquad (4.20)$$

where Q is the positive semidefinite solution to the Lyapunov equation

$$A_2 Q + Q A_2^T + B_w B_w^T = 0. \qquad (4.21)$$

If the constraint on Q in (4.21) is written in an equivalent form,

$$\text{tr}[(A_2 Q + Q A_2^T + B_w B_w^T)X] = 0 \text{ for all X}, \qquad (4.22)$$

then

$$\|T_{zw}\|_2^2 = \text{tr}[QC_z^T C_z] + \text{tr}[(A_2 Q + Q A_2^T + B_w B_w^T)X] \text{ for all X}. \qquad (4.23)$$

To simplify the notation, let J equal the two-norm squared of $T_{zw}$. Notice from (4.23) that J is an explicit function of the design variables and the matrices, X and Q. However, the partial derivative of J with respect to X is simply the left side of (4.21) and, therefore, it is always equal to zero. Further, it can be shown that

$$\frac{\partial J}{\partial Q} = A_2^T X + X A_2 + C_z^T C_z, \qquad (4.24)$$

4-6

which is a Lyapunov equation. Since X is arbitrary, let X be the positive semidefinite solution to the Lyapunov equation

$$A_2^T X + X A_2 + C_z^T C_z = 0. \tag{4.25}$$

This choice of X is guaranteed to exist since $A_2$ must be stable. With this choice, the partial of J with respect to Q is also zero, and the only remaining derivatives to be calculated are the partials of J with respect to the design variables (i.e. $A_k$, $B_k$ and $C_k$).

If Q is given by

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix} \tag{4.26}$$

and X is given by

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{12}^T & X_{22} \end{bmatrix}, \tag{4.27}$$

then these derivatives are

$$\frac{\partial J}{\partial A_k} = 2 \left[ X_{12}^T Q_{12} + X_{22} Q_{22} \right]$$

$$\frac{\partial J}{\partial B_k} = 2 \left[ X_{12}^T Q_{11} C_{y2}^T + X_{22} Q_{12}^T C_{y2}^T + X_{12}^T B_w D_{yw}^T + X_2 B_k D_{yw} D_{yw}^T \right]$$

$$\frac{\partial J}{\partial C_k} = 2 \left[ B_{u2}^T X_{11} Q_{12} + B_{u2}^T X_{12} Q_{22} + D_{zu}^T C_z Q_{12} + D_{zu}^T D_{zu} C_k Q_{22} \right]. \tag{4.28}$$

The complete method for finding the partial derivatives of J with respect to the vectorized compensator, c, is as follows:

i) Solve the Lyapunov equations in (4.21) and (4.25) for Q and X respectively

ii) Compute the partial derivatives in (4.28)

iii) Rearrange (4.28) to express the gradient with respect to the vector of design variables, c, as a vector,

$$\frac{\partial J}{\partial c} = \left[ \left( \frac{\partial J}{\partial A_k} \right)_1^T \cdots \left( \frac{\partial J}{\partial A_k} \right)_{n_k}^T \quad \left( \frac{\partial J}{\partial B_k} \right)_1^T \cdots \left( \frac{\partial J}{\partial B_k} \right)_{n_y}^T \quad \left( \frac{\partial J}{\partial C_k} \right)_1^T \cdots \left( \frac{\partial J}{\partial C_k} \right)_{n_k}^T \right]^T \quad (4.29)$$

where the individual vectors in parentheses are the columns of the partial derivative matrices, $n_k$ is the order of the compensator, and $n_y$ is the number of measurements. The next section discusses two numerical methods of approximating the $L_1$ norm of $T_{mr}$.

## 4.4. Calculating the $L_1$ norm

As mentioned in Section 2.7, there is no known way of analytically computing the $L_1$ norm of a continuous system. Many of the current approaches to the problem involve discretizing the continuous system with the forward Euler rule, and computing the $\ell_1$ norm of the resulting discrete system. These methods, however, prove to be impractical in many cases. This section presents two numerical approaches which attack the problem directly, and avoid the complications associated with forward Euler transformation.

The $L_1$ norm of a continuous, SISO system is given by

$$\|T_{mr}\|_1 = \int_0^\infty \left| C_m e^{A_1 t} B_r \right| dt + \left| D_{mr} \right|. \quad (4.30)$$

If $A_1$ is stable, then $e^{A_1 t}$ approaches zero as t approaches infinity, and (4.30) can be approximated by truncating the integral at some point in time, $t_N$, which can be computed from the eigenvalues of $A_1$. The remaining question is how best to evaluate the truncated integral.

The first approach to this problem is to eliminate the absolute value sign inside the integral by determining where the impulse response is positive and negative. This can be done by discretizing the continuous system with a ZOH, and finding the pulse response of the resulting discrete system. Since the ZOH transformation preserves the values of the continuous impulse response at the sample points, the discrete pulse response can be used

to find approximate locations where the continuous impulse response is zero. These approximate locations are then refined to any degree of accuracy by using a nonlinear root-solver on the continuous function, $C_m e^{A_1 t} B_r$. Once the zero locations of the impulse response are known, the absolute value sign can be removed by breaking the integral in (4.30) into a series of integrations. Further, if $A_1$ is invertible (i.e. $A_1$ has no zero eigenvalues), then

$$\int_{t_1}^{t_2} C_m e^{A_1 t} B_r \, dt = C_m A_1^{-1} [e^{A_1 t_2} - e^{A_1 t_1} - 2I] B_r . \tag{4.31}$$

Thus, each integral in the series can be determined exactly. When $A_1$ does have eigenvalues at the origin, the system is unstable and the one-norm is infinite.

The key to this method rests on determining the zero locations of the continuous impulse response. If zero locations are missed in the discretization step due to high frequency dynamics, then the one-norm will be inaccurate. In addition, most nonlinear root-solvers are only capable of finding the closest root to a given initial guess. This implies that the approximate root locations must be fairly precise, i.e. the discretization sample period must be fairly small. Using a small sample period to estimate the impulse response of systems with fast and slow dynamics can be expensive in terms of computer time. These issues make the above method impractical to use in a computer algorithm which must handle a wide range of problems. However, this method does offer an alternative way to calculate the one-norm for specific examples, and can be used to check another method.

The second approach to approximating the truncated integral is more robust but less accurate. In this method, the continuous system is discretized with a ZOH transformation using a small sample period. The truncated integral is then approximated with a trapezoidal integration of the discrete pulse response. As the sample period decreases, the approximation improves. If $H_2/L_1$ optimization is performed near $L_1$

optimal, where $t_N$ is small, this method works well at approximating the one-norm of a system without requiring an unreasonable number of samples. However, finding a starting point near $L_1$ optimal can be difficult. Currently, the most reliable method of finding an adequate starting point is to use a solution from fixed-order $H_2/H_\infty$ optimization ([Wal94]) near $H_\infty$ optimal (see [Rid92] for an excellent discussion of $H_\infty$ optimization).

The $L_1$ norm for MIMO systems is calculated from the maximum row sum of SISO transfer function norms. However, discontinuous gradients can occur when the maximum row sum occurs over more than one row. To counter this problem, each row sum is constrained as a separate Multiple Input Single Output (MISO) transfer function. This effectively adds more constraints to the $H_2/L_1$ optimization problem, but most of the additional constraints are not active at any specific design point. If an optimization algorithm is used which only calls for gradients of the active constraints, then these additional constraints have little impact on the overall performance of the $H_2/L_1$ algorithm. The next section discusses how to calculate the gradients associated with the $L_1$ norm.

### 4.5. Gradients of the One-Norm

This section derives the gradients of the one-norm with respect to the design variables in the matrices $A_k$, $B_k$, and $C_k$. Since all of these matrices appear in $A_1$ [see (4.12)], the first problem considered is how to compute the partial of $e^{A_1 t}$ with respect to any element of $A_1$. This is the most difficult issue in calculating the gradients of the one-norm with respect to the design variables. Contributions to the one-norm gradients from the other closed-loop state-space matrices, which are considerably simpler to find, are discussed thereafter. In solving the first problem, it will also become clear how to calculate the gradient of the added stability constraint in (4.19).

If $A_1$ is a non-defective matrix ($A_1$ can be diagonalized), then the partial of $e^{A_1 t}$ with respect to any element of $A_1$ is given by

$$\frac{\partial e^{A_1 t}}{\partial a_{ij}} = \frac{\partial}{\partial a_{ij}}\left[Re^{\Lambda t} R^{-1}\right] = \frac{\partial R}{\partial a_{ij}} e^{\Lambda t} R^{-1} + R \frac{\partial e^{\Lambda t}}{\partial a_{ij}} R^{-1} + Re^{\Lambda t} \frac{\partial R^{-1}}{\partial a_{ij}}, \qquad (4.32)$$

where R is the right eigenvector matrix of $A_1$ and $\Lambda$ is a diagonal matrix consisting of the eigenvalues of $A_1$. The partial of $e^{\Lambda t}$ with respect to $a_{ij}$ can be computed easily from

$$\frac{\partial e^{\Lambda t}}{\partial a_{ij}} = \sum_{i=1}^{n_1} \frac{\partial e^{\Lambda t}}{\partial \lambda_i} \frac{\partial \lambda_i}{\partial a_{ij}}, \qquad (4.33)$$

provided that the partial derivative of each eigenvalue is known with respect to $a_{ij}$. Likewise, the partial derivative of $R^{-1}$ with respect to $a_{ij}$ can be easily computed from

$$\frac{\partial R^{-1}}{\partial a_{ij}} = -R^{-1} \frac{\partial R}{\partial a_{ij}} R^{-1}, \qquad (4.34)$$

provided that the partial derivative of the right eigenvector matrix with respect to $a_{ij}$ is known.

The partials of an eigenvalue and eigenvector with respect to $a_{ij}$ can be computed from the standard eigenvalue problem if the $n_1$ eigenvalues of $A_1$ are distinct. The derivations of these two derivatives involve both the left and right eigenvectors of $A_1$, and are described in detail in [Nel76]. Letting

$$(\cdot)' = \frac{\partial(\cdot)}{\partial a_{ij}},$$

the solutions are given by

$$\lambda_i' = L_i^T A_1' R_i \qquad (4.35)$$

$$R_i' = \sum_{k=1}^{n_1} c_k R_k + c_i R_i = V_i + c_i R_i, \qquad (4.36)$$

where

$$c_k = \frac{L_k^T \left[R_i \lambda_i' - A_1' R_i\right]}{\lambda_k - \lambda_i}, \qquad i \neq k \qquad (4.37)$$

$$c_i = -\text{Re}\left(R_i^H M V_i\right) - \frac{1}{2} R_i^H M' R_i \qquad (4.38)$$

and

$$R_i^H M R_i = 1. \qquad (4.39)$$

The symbol, $(\cdot)^H$, denotes the complex conjugate transpose of $(\cdot)$, and L is the left eigenvector matrix. $R_i$ and $L_i$ refer to the $i^{th}$ eigenvector of R and L, respectively. Note that $A_1'$ is simply a $n_1 \times n_1$ matrix of zeros with a one in the (i,j) element. The real part of (4.35) provides a method of computing the gradients of the stability constraint in (4.19) if $\lambda_i$ is the maximum eigenvalue of $A_1$.

The partial of $e^{A_1 t}$ with respect to $a_{ij}$ can be completely determined from (4.33)-(4.39). Using this information, the partial derivative of the one-norm with respect to $A_1$ can be found element-by-element from

$$\frac{\partial \|T_{mr}\|_1}{\partial A_{1_{i,j}}} = \int_0^\infty \text{sgn}(C_m e^{A_1 t} B_r) C_m \frac{\partial e^{A_1 t}}{\partial a_{ij}} B_r dt, \qquad (4.40)$$

where $\text{sgn}(\cdot)$ is 1, -1, or 0 depending on the sign of $(\cdot)$. The partial derivatives with respect to the other closed-loop matrices are given by

$$\frac{\partial \|T_{mr}\|_1}{\partial B_r} = \int_0^\infty \text{sgn}(C_m e^{A_1 t} B_r)\left[\left(e^{A_1 t}\right)^T C_m^T\right] dt \qquad (4.41)$$

$$\frac{\partial \|T_{mr}\|_1}{\partial C_m} = \int_0^\infty \text{sgn}(C_m e^{A_1 t} B_r)\left[B_r^T \left(e^{A_1 t}\right)^T\right] dt \qquad (4.42)$$

$$\frac{\partial \|T_{mr}\|_1}{\partial D_{mr}} = \text{sgn}(D_{mr}). \qquad (4.43)$$

From these expressions, the partials of the one-norm with respect to $A_k$, $B_k$, and $C_k$ can be expressed as

$$\frac{\partial \|T_{mr}\|_1}{\partial A_{c_{i,j}}} = \left[\frac{\partial \|T_{mr}\|_1}{\partial A_1}\right]_{n_1+i,n_1+j} \tag{4.44}$$

$$\frac{\partial \|T_{mr}\|_1}{\partial B_{c_{i,j}}} = \sum_{p=1}^{n_k}\sum_{q=1}^{n_1} C_{yl_{j,q}}\left[\frac{\partial \|T_{mr}\|_1}{\partial A_1}\right]_{n_1+i,n_1+j} + \sum_{p=1}^{n_k}\sum_{q=1}^{n_r} \delta_{p,i} D_{yr_{j,q}}\left[\frac{\partial \|T_{mr}\|_1}{\partial B_r}\right]_{n_1+p,q} \tag{4.45}$$

$$\frac{\partial \|T_{mr}\|_1}{\partial C_{c_{i,j}}} = \sum_{p=1}^{n_1}\sum_{q=1}^{n_k} \delta_{j,q} B_{ul_{p,i}}\left[\frac{\partial \|T_{mr}\|_1}{\partial A_1}\right]_{p,n_1+q} + \sum_{p=1}^{n_m}\sum_{q=1}^{n_k} \delta_{j,q} D_{mu_{p,i}}\left[\frac{\partial \|T_{mr}\|_1}{\partial C_m}\right]_{p,n_1+q} . \tag{4.46}$$

The partial of the one-norm of $T_{mr}$ with respect to c can be formed from the columns of matrices in (4.44)-(4.46) as described in Section 4.3.

The integrals in (4.40)-(4.43) are approximated in the same manner as the integral in the one-norm calculation. The upper limit of integration is changed to $t_N$, and trapezoidal integration is performed over discrete points from the continuous function. The same discretization period is used for the norm calculation and the gradients, which allows the sign factor required in (4.40)-(4.43) to be computed from the data gathered in evaluating the one-norm.

As mentioned in the previous section, separate MISO gradients are calculated for each row of a MIMO system. This ensures that continuous gradient information is available regardless of where the maximum row sum occurs.

Recall that development of $R'$ and $\lambda'$ require that the eigenvalues of $A_1$ be distinct. Clearly this condition can be violated while performing $H_2/L_1$ optimization. The current algorithm switches to finite difference calculations for gradient information if this occurs. Research is still being conducted to develop ways to handle an $A_1$ matrix with repeated eigenvalues.

## 4.6. Computer Implementation

The $H_2/L_1$ optimization problem is currently implemented using the MATLAB™ SQP routine, *constr.m*. Separate subroutines for each norm, constraint and gradient calculation are called from this routine. Unfortunately, this routine currently requires all gradient information regardless of whether or not a constraint is active. FORTRAN™ shells can be written to allow the MATLAB™ subroutines to interface with IMSL™ optimization routines, which eliminate this problem. Section 4.7 demonstrates the $H_2/L_1$ optimization algorithm on a design problem for the AFTI F-16.

## 4.7. $H_2/L_1$ Design Example

A longitudinal controller design for the AFTI F-16, shown in detail in Appendix A, is used to demonstrate the $H_2/L_1$ optimization method. The primary objective of the $H_2$ portion of the problem is to minimize the effect of wind disturbances and measurement noise. The $L_1$ portion of the problem is designed to improve tracking performance.

## 4.7.1. The $H_2$ Problem

The $H_2$ problem for the AFTI F-16, taken from [Luk93], is to find an internally stabilizing controller which minimizes the response of the normal acceleration and weighted control to the wind disturbances and measurement noise. A block diagram of the $H_2$ problem is shown in Figure 4.3.
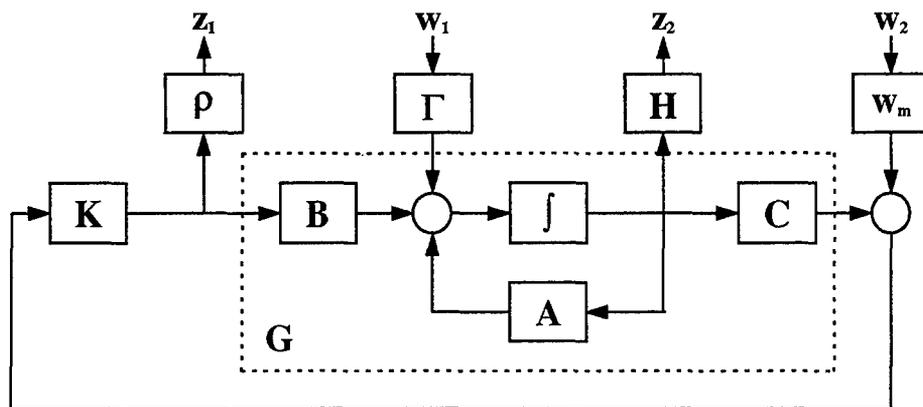
Figure 4.3. F-16 $H_2$ diagram

The control weight, $\rho$, equals 10 and the state weighting matrix, H, equals the system C matrix. The wind disturbance, $w_1$, is modeled as a white Gaussian noise (WGN) with $5.0 \times 10^{-4}$ $rad^2$-sec intensity, and $\Gamma$ is the column of A corresponding to the angle-of-attack state, $\alpha$. The measurement noise, $w_2$, is modeled as a WGN with $1.6 \times 10^{-5}$ $g^2$-sec intensity and $w_m$ equals 1. Weighted control power, $z_1$, and normal acceleration, $z_2$, are the controlled outputs. The plant state-space matrices are given in Appendix A and the resulting $H_2$ matrices are given in Appendix C.

### 4.7.2. The $L_1$ Problem

The $L_1$ problem, depicted in Figure 4.4, is a weighted sensitivity minimization design. $W_s$ is the inverse of the desired sensitivity, and is given by

$$W_s = \frac{s+10}{s+0.0001} .$$
(4.47)

A plot of the desired sensitivity is shown in Figure 4.5. The matrices for the $L_1$ problem are also given in Appendix C.
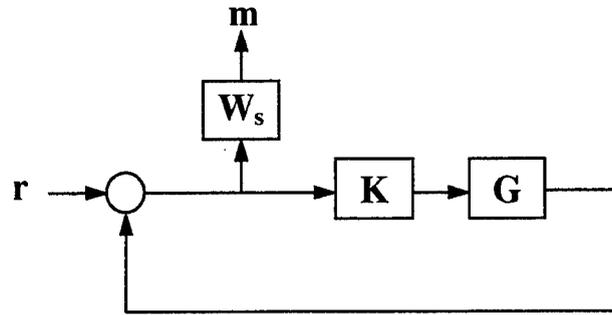
Figure 4.4. F-16 $L_1$ diagram
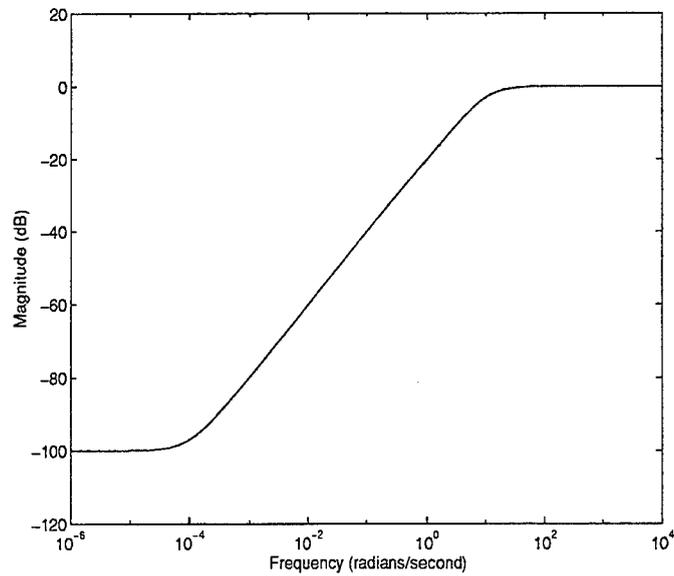


Figure 4.5. F-16 desired sensitivity

### 4.7.3. $H_2/L_1$ Results

A plot of the $H_2/L_1$ solution curve for a 6th order compensator is shown in Figure 4.6.
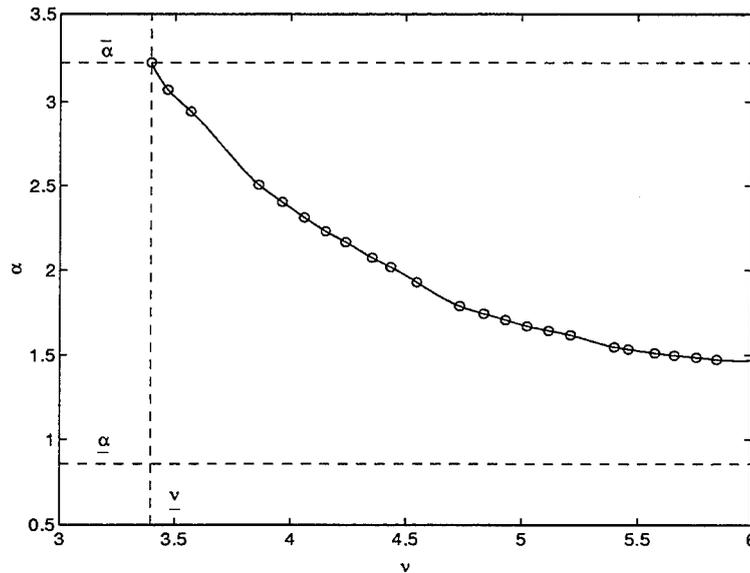


Figure 4.6. F-16 $H_2/L_1$ solution curve

Notice that the plot only depicts mixed solutions near $L_1$ optimal. A complete solution curve would extend much further along the horizontal axis, and asymptotically approach the dashed line, labeled $\underline{\alpha}$. Values for several of the solution points, depicted as circles in Figure 4.6, are given in Table 4.1.

The points are numbered from left to right in Figure 4.6. Point number 24 represents the $H_2$ optimal solution, which is not depicted in Figure 4.6. The points in Table 4.1 are referred to as the design points for the remainder of this section.

Plots of the sensitivity and complementary sensitivity for the design points are shown in Figures 4.7 and 4.8.

| point # | α | ν |
|---------|------|--------|
| 1 | 3.23 | 3.40 |
| 5 | 2.40 | 3.96 |
| 9 | 2.07 | 4.36 |
| 15 | 1.67 | 5.02 |
| 21 | 1.50 | 5.66 |
| 24 | 0.86 | 97,190 |

Table 4.1.  F-16 solution points
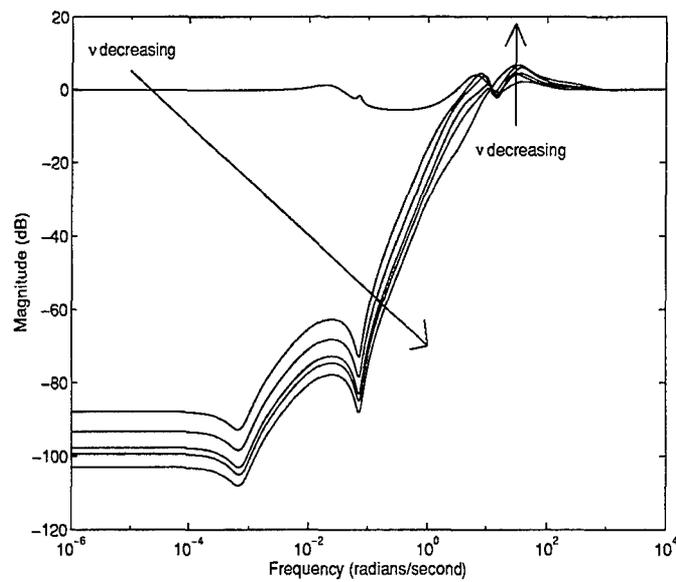


Figure 4.7.  F-16 sensitivity
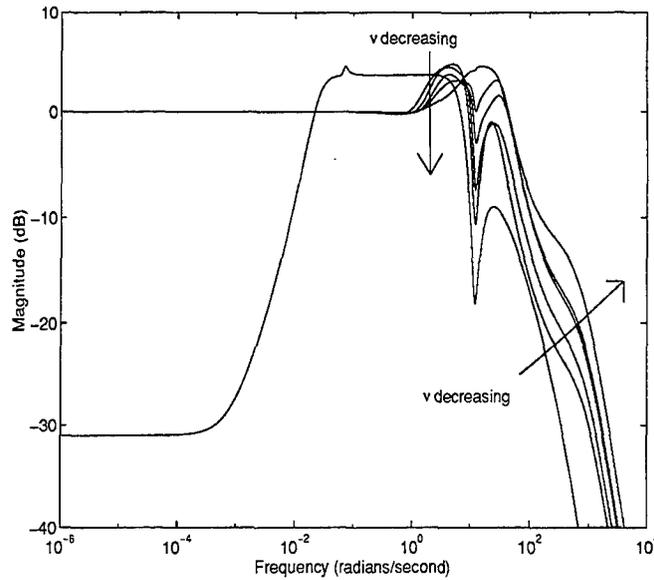
Figure 4.8. F-16 complementary sensitivity

In each plot, one curve is dramatically different than the others. This curve corresponds to design point 24, the $H_2$ solution.

The vector gain and phase margins for the design points are shown in Table 4.2.

| point # | lower VGM (dB) | upper VGM (dB) | VPM (deg) |
|---------|----------------|----------------|-----------|
| 1 | -7.77 | 5.42 | 34.4 |
| 5 | -10.3 | 5.42 | 40.7 |
| 9 | -9.12 | 6.10 | 37.9 |
| 15 | -7.88 | 8.02 | 35.1 |
| 21 | -7.50 | 8.11 | 35.3 |
| 24 | -7.82 | 8.80 | 37.1 |

Table 4.2. F-16 vector gain and phase margins

Notice that the stability margins do not consistently improve as $\nu$ is decreased. However, the margins are acceptable at all of the design points.

The open-loop GK plot is shown in Figure 4.9.
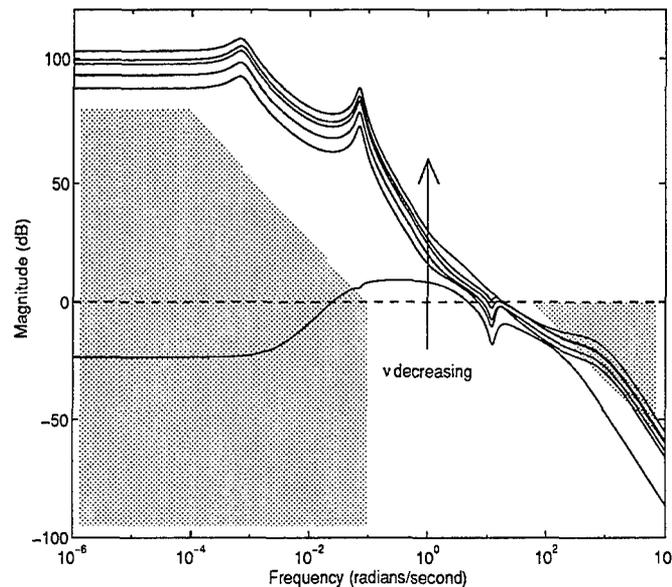


Figure 4.9.  F-16 open-loop GK

The shaded area on the left side of Figure 4.9 represents a performance and disturbance rejection barrier. The shaded area on the right side of Figure 4.9 represents a sensor noise and unmodeled dynamics barrier. Descriptions of both barriers were taken from [RB86], which also contains an excellent discussion of desired GK shapes.

Notice that the mixed design points, all relatively near $L_1$ optimal, miss the barrier on the left, but pass within the barrier on the right. A design which meets both barriers could be found by a mixed solution closer to $H_2$ optimal. Alternatively, the mixed $H_2/L_1$ optimization design could be redone with the weighting

$$W_s = \frac{s+4}{s+0.0001}$$

on the $L_1$ problem, instead of the weighting in (4.47). This new weighting should provide acceptable GK loop shapes for mixed $H_2/L_1$ solutions near $L_1$ optimal, without decreasing the bandwidth of the system below the frequency of average pilot commands.

The loop shapes from the design points imply that high frequency noise will be more prevalent than low frequency noise in system responses. Systems obtained from these design points should also track low frequency commands well. Both of these facts can be seen in the step responses presented below.

Figure 4.10 shows the F-16 responses, without noise, to a commanded 1g step input for the different designs. Notice that the $H_2$ solution tracks with a steady-state error while the mixed designs do not. Step responses with noise for design points 1, 9, 15 and 24 are shown in Figures 4.11-4.14, respectively. As expected, the systems with lower $\nu$ values contain more high frequency noise than those closer to the $H_2$ optimal design. The most important item to note from Figures 4.11-4.14 is that the tracking performance of the AFTI F-16 can be greatly improved using mixed $H_2/L_1$ designs, with very little increase on the amount of noise in the system response.
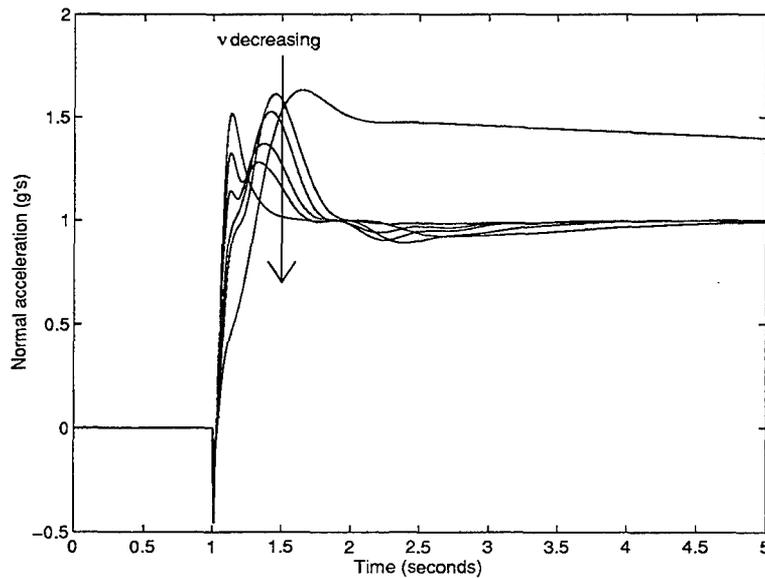


Figure 4.10. F-16 step responses without noise
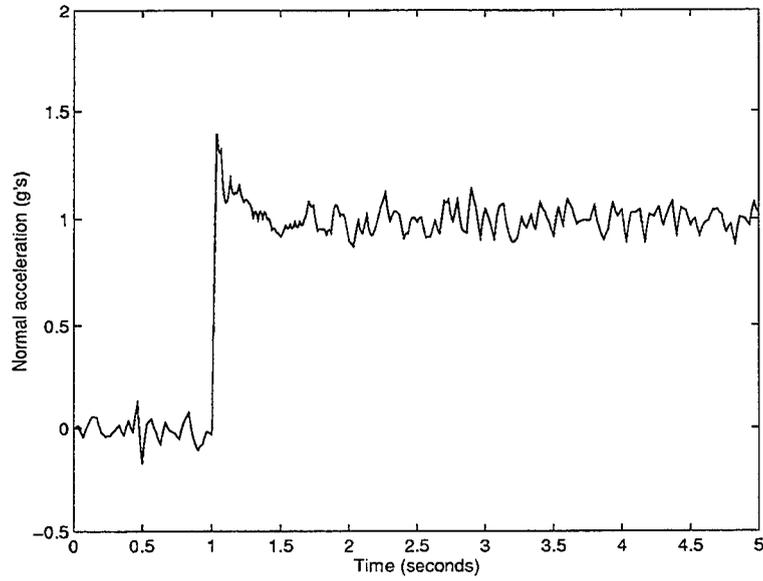
Figure 4.11.  F-16 step response with noise, design point 1
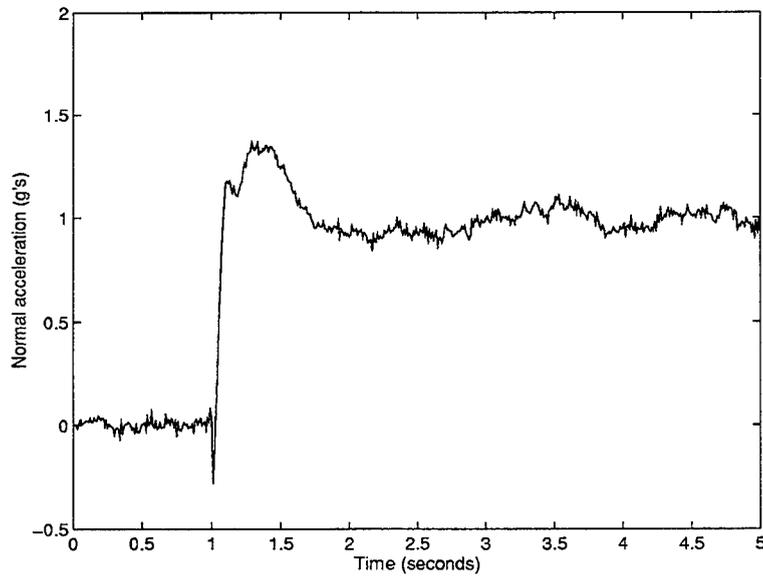


Figure 4.12.  F-16 step response with noise, design point 9
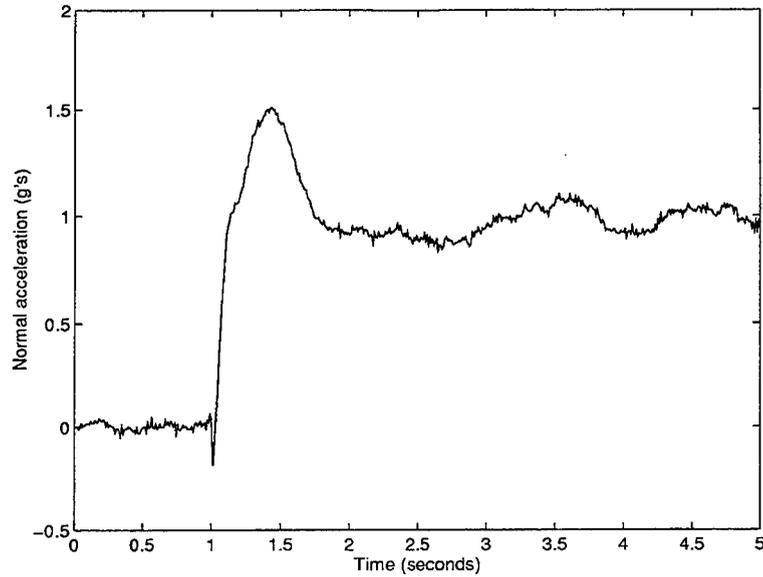
Figure 4.13. F-16 step response with noise, design point 15
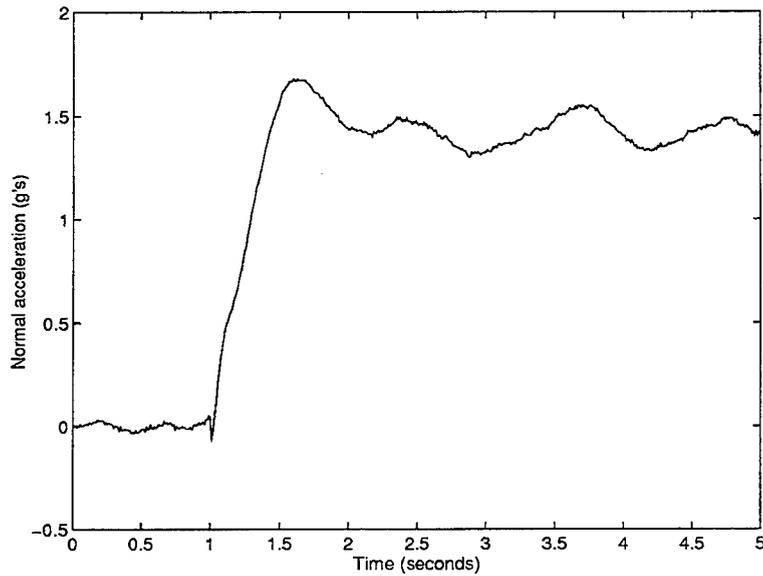


Figure 4.14. F-16 step response with noise, design point 24

Design point 15 was chosen as the best mixed $H_2/L_1$ design based on stability margins, loop shape and step responses. This particular design will be compared to the best discrete $H_2/l_1$ design in the next chapter. The control usage for this particular design point is shown in Figure 4.15.
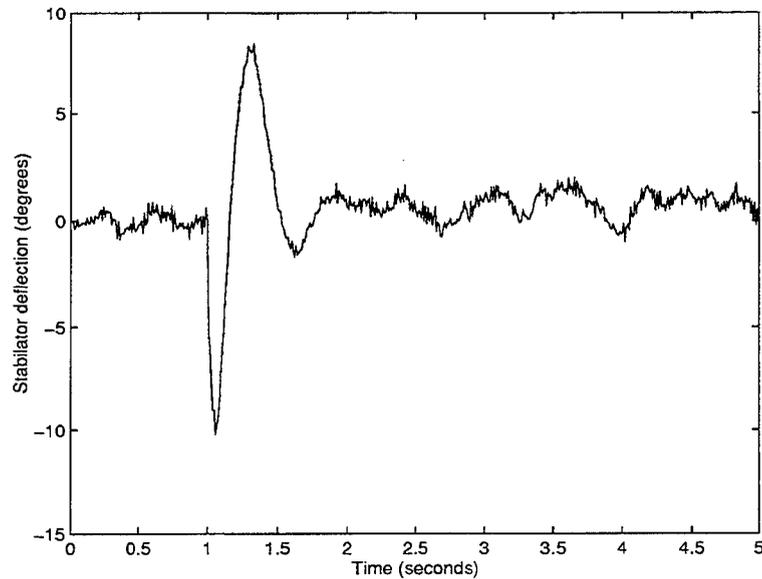
Figure 4.15. F-16 control usage with noise, design point 15

The same control deflection and rate problems noted in Section 3.4 (for the same $W_s$) are also evident in Figure 4.15. The current $H_2/L_1$ methodology could, however, be extended to include $L_1$ control deflection and rate constraints. This should be a topic of further research.

### 4.8. Chapter Summary

This chapter presented a newly developed numerical approach to $H_2/L_1$ optimization for continuous systems. This method was tested on a realistic example, and the results show the benefit of mixing $H_2$ and $L_1$ design problems. The controllers obtained in this mixed design approach, however, must still be discretized before they can be used on an actual aircraft. Unfortunately, the performance of the aircraft with the discretized controller can be significantly different than the results obtained with the continuous controller. An alternative, and potentially better design approach is to use mixed $H_2/l_1$ optimization to directly obtain a discrete controller. This method is presented in Chapter 5.

# V.  Mixed $H_2/\ell_1$ Optimization

## 5.1.  Chapter Overview

This chapter develops a fixed-order, mixed $H_2/\ell_1$ optimization method to design controllers for discrete systems.  The chapter is organized in the same manner as Chapter 4 to highlight the similarities and differences between mixed $H_2/\ell_1$ optimization for discrete systems and mixed $H_2/L_1$ optimization for continuous systems.  Section 5.2 covers the mixed $H_2/\ell_1$ optimization problem.  Section 5.3 discusses analytical gradients for the $H_2$ portion of the problem.  Methods of numerically approximating the $\ell_1$ norm and its gradient are presented in Sections 5.4 and 5.5, respectively.  Computer implementation issues associated with $H_2/\ell_1$ optimization are discussed in Section 5.6.  In Section 5.7, the $H_2/\ell_1$ optimization method is demonstrated on the discrete version of the AFTI  F-16 longitudinal model, and the results are compared to the $H_2/L_1$ results obtained in Chapter 4.

## 5.2.  The Mixed $H_2/\ell_1$ Problem

Much of the mixed $H_2/L_1$ optimization problem development presented in Section 4.2 also applies to the mixed $H_2/\ell_1$ optimization problem.  The purpose of this section is to simply point out the differences between the two methods.

The system in Figure 4.1, where the inputs and outputs are now sequences, represents the standard mixed $H_2/\ell_1$ problem.  The goal of $H_2/\ell_1$ optimization is stated in (4.1)-(4.3) and the individual $H_2$ and $\ell_1$ problems can be expressed in terms of the state-space systems given in (4.4) and (4.5).

The following assumptions are made on the state-space elements in (4.4) and (4.5) using the definitions given in (2.61) and (2.62):

i) $(A_2, B_{u2})$ is stabilizable and $(C_{y2}, A)$ is detectable

ii) $R_f, R_c > 0$

iii) $D_{zu}^T C_z = 0$ and $B_w D_{yw}^T = 0$

iv) $(A_1, B_{u1})$ is stabilizable and $(C_{y1}, A_1)$ is detectable

See Section 2.9 for the reasons for assumptions i)-iii). Assumption iv) is required to ensure the $\ell_1$ problem has a solution.

The controller state-space is given by (4.6), and the closed-loop state-space equations for $T_{zw}$ are given by (4.7). Unlike the continuous $H_2$ problem, a finite 2-norm for $T_{zw}$ can be attained when $\mathbf{D}_{zw}$ is non-zero in the discrete $H_2$ problem. Thus, a strictly proper controller cannot be assumed in the mixed $H_2/\ell_1$ optimization problem. Given this fact, the state-space equations for the mixed $H_2/\ell_1$ problem are:

$$\dot{\mathbf{x}}_2 = \mathbf{A}_2\mathbf{x}_2 + \mathbf{B}_w w$$
$$z = \mathbf{C}_z\mathbf{x}_2 + \mathbf{D}_{zw}w$$
$$\dot{\mathbf{x}}_1 = \mathbf{A}_1\mathbf{x}_1 + \mathbf{B}_r r$$
$$m = \mathbf{C}_m\mathbf{x}_1 + \mathbf{D}_{mr}r, \tag{5.1}$$

where

$$\mathbf{x}_2 = \begin{bmatrix} x_2 \\ x_k \end{bmatrix} \tag{5.2}$$

$$\mathbf{x}_1 = \begin{bmatrix} x_1 \\ x_k \end{bmatrix} \tag{5.3}$$

$$\mathbf{A}_2 = \begin{bmatrix} A_2 + B_{u2}D_kC_{y2} & B_{u2}C_k \\ B_kC_{y2} & A_k \end{bmatrix} \tag{5.4}$$

$$\mathbf{A}_1 = \begin{bmatrix} A_1 + B_{u1}D_kC_{y1} & B_{u1}C_k \\ B_kC_{y1} & A_k \end{bmatrix} \tag{5.5}$$

$$\mathbf{B}_w = \begin{bmatrix} B_w + B_{u2}D_kD_{yw} \\ B_kD_{yw} \end{bmatrix} \tag{5.6}$$

$$\mathbf{B}_r = \begin{bmatrix} B_r + B_{u1}D_kD_{yr} \\ B_kD_{yr} \end{bmatrix} \tag{5.7}$$

$$\mathbf{C}_z = \begin{bmatrix} C_z + D_{zu}D_kC_{y2} & D_{zu}C_k \end{bmatrix} \tag{5.8}$$

$$\mathbf{C}_m = \begin{bmatrix} C_m + D_{mu}D_kC_{y1} & D_{mu}C_k \end{bmatrix} \tag{5.9}$$

$$\mathbf{D}_{zw} = \begin{bmatrix} D_{zu}D_kD_{yw} + D_{zw} \end{bmatrix} \tag{5.10}$$

$$\mathbf{D}_{mr} = \begin{bmatrix} D_{mu}D_kD_{yr} + D_{mr} \end{bmatrix} \tag{5.11}$$

The definitions in (4.18) are used to discuss solutions to the mixed $H_2/\ell_1$ problem, which must satisfy the Kuhn-Tucker necessary conditions given in Section 4.2. An illustration of a possible $H_2/\ell_1$ solution curve is shown in Figure 4.2.

The $H_2/\ell_1$ problem, like the $H_2/L_1$ problem, is solved numerically using a SQP algorithm. The objective (f) and the constraints (g) are given by

$$f(c) = \varsigma_2 \|T_{zw}\|_2^2$$
$$g_1(c) = \varsigma_1 \left( \|T_{mr}\|_1 - \nu \right)$$
$$g_s(c) = \varsigma_s \left\{ \max_i \left( |\lambda_i(A_2)|^2 \right) - 1 \right\}. \tag{5.12}$$

The only difference between the equations in (5.12) and their continuous equivalents in (4.19) is in the stability constraint. Recall that a discrete system in the z-domain is stable if and only if its closed-loop eigenvalues lie inside the open unit disk.

Modal form is also assumed for the controller, K, in the $H_2/\ell_1$ optimization problem, in order to minimize the number of design variables in c. A block-Jordan form or fully populated state-space could be used to allow for repeated eigenvalues in the controller.

In order to use the SQP algorithm to numerically solve the $H_2/\ell_1$ problem, gradients must be found for the objective function and each of the constraints. Analytical gradients for the stability constraint can be easily derived from (4.35). Gradients for one- and two-norms are presented in Sections 5.5 and 5.3, respectively.

## 5.3. Gradients of the Two-Norm

Recall from Section 2.9 that the two-norm squared of $T_{zw}$ is given by

$$\|T_{zw}\|_2^2 = \text{tr}\left[D_{zw}D_{zw}^T + C_z Q C_z^T\right], \tag{5.13}$$

where Q is the positive semidefinite solution to the Lyapunov equation

$$A_2 Q A_2^T + B_w B_w^T = Q. \tag{5.14}$$

If the constraint on Q in (5.14) is written in an equivalent form,

$$\text{tr}[(A_2 Q A_2^T + B_w B_w^T - Q)X] = 0 \text{ for all } X, \tag{5.15}$$

then

$$\|T_{zw}\|_2^2 = \text{tr}[D_{zw}^T D_{zw} + C_z Q C_z^T] + \text{tr}[(A_2 Q A_2^T + B_w B_w^T - Q)X] \text{ for all } X. \tag{5.16}$$

To simplify the notation, let J equal the two-norm squared of $T_{zw}$. Notice from (5.16) that J is a function of the design variables and the matrices X and Q. However,

$$\frac{\partial J}{\partial X} = A_2 Q A_2^T + B_w B_w^T - Q = 0. \tag{5.17}$$

Further, it can be shown that

$$\frac{\partial J}{\partial Q} = A_2^T X A_2 + C_z^T C_z - X, \tag{5.18}$$

which is a Lyapunov equation. Since X is arbitrary, let X be the positive semidefinite solution to the Lyapunov equation

$$A_2^T X A_2 + C_z^T C_z - X = 0. \tag{5.19}$$

With this choice of X, (5.18) is equal to zero. This choice of X will always exist since $A_2$ must be stable. The only remaining derivatives to be calculated are the partials of J with respect to the design variables. If Q and X are given by (4.26) and (4.27), respectively, then these derivatives are

$$\frac{\partial J}{\partial A_k} = 2\left[X_{12}^T A_2 Q_{12} + X_{12}^T B_{u2} C_k Q_{22} + X_{22} B_k C_{y2} Q_{12} + X_{22} A_k Q_{22} \right.$$
$$\left. + X_{12}^T B_{u2} D_k C_{y2} Q_{12}\right] \tag{5.20}$$

$$\frac{\partial J}{\partial B_k} = 2\Big[ X_{12}^T A_2 Q_{11} C_{y2}^T + X_{12}^T B_{u2} Q_{12}^T C_{y2}^T + X_{22} B_k C_{y2} Q_{11} C_{y2}^T$$
$$+ X_{22} A_k Q_{12}^T C_{y2}^T + X_{12}^T B_w D_{yw}^T + X_{22} B_k D_{yw} D_{yw}^T$$
$$+ X_{12}^T B_{u2} D_k C_{y2} Q_{11} C_{y2}^T + X_{12}^T B_{u2} D_k D_{yw} D_{yw}^T \Big] \tag{5.21}$$

$$\frac{\partial J}{\partial C_k} = 2\Big[ D_{zu}^T C_z Q_{12} + D_{zu}^T D_{zu} C_k Q_{22} + B_{u2}^T X_{11} A_2 Q_{12}$$
$$+ B_{u2}^T X_{11} B_{u2} C_k Q_{22} + B_{u2}^T X_{12} B_k C_{y2} Q_{12} + B_{u2}^T X_{12} A_k Q_{22}$$
$$+ D_{zu}^T D_{zu} D_c C_{y2} Q_{12} + B_{u2}^T X_{11} B_{u2} D_k C_{y2} Q_{12} \Big] \tag{5.22}$$

$$\frac{\partial J}{\partial D_k} = 2\Big[ D_{zu}^T D_{zw} D_{yw}^T + D_{zu}^T D_{zu} D_k D_{yw} D_{yw}^T + D_{zu}^T C_z Q_{11} C_{y2}^T$$
$$+ D_{zu}^T D_{zu} D_k C_{y2} Q_{11} C_{y2}^T + D_{zu}^T D_{zu} C_k Q_{12}^T C_{y2}^T + B_{u2}^T X_{11} A_2 Q_{11} C_{y2}^T$$
$$+ B_{u2}^T X_{11} B_{u2} D_k C_{y2} Q_{11} C_{y2}^T + B_{u2}^T X_{11} B_{u2} C_k C_{y2} Q_{12}^T C_{y2}^T$$
$$+ B_{u2}^T X_{12} B_k C_{y2} Q_{11} C_{y2}^T + B_{u2}^T X_{12} A_k Q_{12}^T C_{y2}^T + B_{u2}^T X_{11} B_w D_{yw}^T$$
$$+ B_{u2}^T X_{11} B_{u2} D_k D_{yw} D_{yw}^T + B_{u2}^T X_{12} B_k D_{yw} D_{yw}^T \Big] \tag{5.23}$$

These equations are much more complicated than the derivatives for the continuous time case, because $D_k \neq 0$.

The complete method for finding the partial derivatives of J with respect to the vectorized compensator, c, is as follows:

i) Solve the Lyapunov equations in (5.14) and (5.19) for Q and X, respectively

ii) Compute the partial derivatives in (5.20)-(5.23)

iii) Rearrange (5.20)-(5.23) to form the gradient with respect to the vector of design variables, c, as a vector

$$\frac{\partial J}{\partial c} = \Bigg[ \left(\frac{\partial J}{\partial A_k}\right)_1^T \cdots \left(\frac{\partial J}{\partial A_k}\right)_{n_k}^T \quad \left(\frac{\partial J}{\partial B_k}\right)_1^T \cdots \left(\frac{\partial J}{\partial B_k}\right)_{n_y}^T \quad \left(\frac{\partial J}{\partial C_k}\right)_1^T \cdots \left(\frac{\partial J}{\partial C_k}\right)_{n_k}^T$$
$$\left(\frac{\partial J}{\partial D_k}\right)_1^T \cdots \left(\frac{\partial J}{\partial D_k}\right)_{n_y}^T \Bigg]^T \tag{5.24}$$

The next section discusses a new numerical method of approximating the $\ell_1$ norm of $T_{mr}$.

## 5.4. Calculating the $l_1$ norm

The $l_1$ norm of a discrete SISO system is given by

$$\left\| T_{mr} \right\|_1 = \sum_{k=0}^{\infty} \left| C_m A_1^k B_r \right| + \left| D_{mr} \right|. \tag{5.25}$$

If $A_1$ is stable, then $A_1^k$ approaches zero as k approaches infinity, and (5.25) can be approximated by truncating the infinite series at some index, N, which can be computed from the eigenvalues of $A_1$. The truncated series can then be evaluated on a computer with a simple loop, where k is the loop index. With each pass through the loop, the absolute value of $C_m A_1^k B_r$ is calculated and added to the cumulative sum of the previous passes. The final sum is added to the absolute value of $D_{mr}$ to produce the one-norm of $T_{mr}$. This approach is used in the MATLAB™ one-norm calculation routine created by Jacques for fixed-order, mixed-norm optimization ([JR94]). The method is extremely reliable, but can be slow if N is large.

The one-norm calculation can be computed considerably faster, however, by taking advantage of the built-in MATLAB™ function, *ltitr.m*. This function rapidly calculates the vector, $(A_1^k B_r)^T$, at each k. The resulting $N \times n_r$ matrix can be multiplied by $C_m^T$ to produce an $N \times 1$ vector, where the $k^{th}$ element equals $C_m A_1^k B_r$. The absolute sum of this vector can easily be calculated and added to the absolute value of $D_{mr}$ to produce the one-norm of $T_{mr}$.

The $l_1$ norm for MIMO systems is calculated from the maximum row sum of SISO transfer functions. To perform this calculation using this author's method, a loop is wrapped around the *ltitr.m* function to cycle through each column of $B_r$.

Algorithms based on both methods were tested on a SUN™ SPARC 20 computer to compare efficiencies. The benchmark problem was a SISO $12^{th}$ order closed-loop system with 2 exogenous inputs, 2 controlled outputs, and 20 design variables in the compensator ($5^{th}$ order compensator). N was set to 5,000. The first method took 3.5 seconds to evaluate the one-norm and the second method took 0.95 seconds (all

performance times refer to cpu time). The second method also calculated the sign of $C_m A_1^k B_r$ for each column of $B_r$ and index k, which can be used to dramatically improve the speed of the $\ell_1$ gradient calculations. In some parts of the $H_2/\ell_1$ algorithm this computation can be removed, and the resulting norm calculation routine is even faster. For example, in Section 5.7, an N of 2 million is required to produce an accurate estimate of the one-norm for the AFTI F-16 problem at $H_2$ optimal. This computation was completed in under 4 minutes on a SUN™ SPARC 20 computer.

## 5.5. Gradients of the One-Norm

The analytical gradients of the one-norm with respect to the design variables and an improved method of calculating these gradients are discussed in this section. The partial derivatives of the one-norm of $T_{mr}$ with respect to the closed-loop state-space can be expressed as

$$\frac{\partial \|T_{mr}\|_1}{\partial A_1} = \sum_{k=0}^{\infty}\left[ \text{sgn}\left(C_m A_1^k B_r\right) \sum_{j=0}^{k-1}\left[\left(A_1^T\right)^j C_m^T B_r^T \left(A_1^T\right)^{k-j-1}\right]\right] \qquad (5.26)$$

$$\frac{\partial \|T_{mr}\|_1}{\partial B_r} = \sum_{k=0}^{\infty} \text{sgn}\left(C_m A_1^k B_r\right)\left[\left(A_1^T\right)^k C_m^T\right] \qquad (5.27)$$

$$\frac{\partial \|T_{mr}\|_1}{\partial C_m} = \sum_{k=0}^{\infty} \text{sgn}\left(C_m A_1^k B_r\right)\left[B_r^T \left(A_1^T\right)^k\right] \qquad (5.28)$$

$$\frac{\partial \|T_{mr}\|_1}{\partial D_{mr}} = \text{sgn}\left(D_{mr}\right). \qquad (5.29)$$

From these expressions, the gradients with respect to the compensator state-space matrices can be expressed as

$$\frac{\partial \|T_{mr}\|_1}{\partial A_{c_{i,j}}} = \left[\frac{\partial \|T_{mr}\|_1}{\partial A_1}\right]_{n_1+i,n_1+j} \qquad (5.30)$$

5-7

$$\frac{\partial \|T_{mr}\|_1}{\partial B_{c_{i,j}}} = \sum_{p=1}^{n_k} \sum_{q=1}^{n_1} C_{yl_{j,q}} \left[ \frac{\partial \|T_{mr}\|_1}{\partial A_1} \right]_{n_1+i,n_1+j} + \sum_{p=1}^{n_k} \sum_{q=1}^{n_r} \delta_{p,i} D_{yr_{j,q}} \left[ \frac{\partial \|T_{mr}\|_1}{\partial B_r} \right]_{n_1+p,q} \qquad (5.31)$$

$$\frac{\partial \|T_{mr}\|_1}{\partial C_{c_{i,j}}} = \sum_{p=1}^{n_1} \sum_{q=1}^{n_k} \delta_{j,q} B_{ul_{p,i}} \left[ \frac{\partial \|T_{mr}\|_1}{\partial A_1} \right]_{p,n_1+q} + \sum_{p=1}^{n_m} \sum_{q=1}^{n_k} \delta_{j,q} D_{mu_{p,i}} \left[ \frac{\partial \|T_{mr}\|_1}{\partial C_m} \right]_{p,n_1+q} \qquad (5.32)$$

$$\frac{\partial \|T_{mr}\|_1}{\partial D_{c_{i,j}}} = \sum_{p=1}^{n_1} \sum_{q=1}^{n_1} B_{ul_{p,i}} C_{yl_{j,q}} \left[ \frac{\partial \|T_{mr}\|_1}{\partial A_1} \right]_{p,q} + \sum_{p=1}^{n_1} \sum_{q=1}^{n_r} B_{ul_{p,i}} D_{yr_{j,q}} \left[ \frac{\partial \|T_{mr}\|_1}{\partial B_r} \right]_{p,q}$$
$$+ \sum_{p=1}^{n_m} \sum_{q=1}^{n_1} D_{mu_{p,i}} C_{yl_{j,q}} \left[ \frac{\partial \|T_{mr}\|_1}{\partial C_m} \right]_{p,q} + \sum_{p=1}^{n_m} \sum_{q=1}^{n_r} D_{mu_{p,i}} D_{yr_{j,q}} \left[ \frac{\partial \|T_{mr}\|_1}{\partial D_{mr}} \right]_{p,q}. \qquad (5.33)$$

The partial of the one-norm of $T_{mr}$ with respect to c can be formed from the columns of the matrices in (5.30)-(5.33) as described in Section 5.3.

Like the one-norm calculation, the infinite summations in (5.26)-(5.29) are truncated at N, and can be calculated with a series of loops. This is the method used in the one-norm gradient routines created by Jacques for fixed-order, mixed-norm optimization ([JR94]). However, large N values in this gradient routine are more costly in terms of computer time than in Jacques' one-norm calculation routine.

Computation time can be minimized by using some of the techniques presented in Section 4.5 to compute the $L_1$ norm for continuous systems. If $A_1$ is a non-defective matrix, then partial of $A_1^k$ with respect to any element of $A_1$ is given by

$$\frac{\partial A_1^k}{\partial a_{ij}} = \frac{\partial}{\partial a_{ij}} \left[ R\Lambda^k R^{-1} \right] = \frac{\partial R}{\partial a_{ij}} \Lambda^k R^{-1} + R \frac{\partial \Lambda^k}{\partial a_{ij}} R^{-1} + R\Lambda^k \frac{\partial R^{-1}}{\partial a_{ij}}. \qquad (5.34)$$

Expressions for most of the partial derivatives on the right side of (5.34) were given in Section 4.5. The only new derivative to be calculated is the partial of $\Lambda^k$ with respect to $a_{ij}$, which is given by

$$\frac{\partial \Lambda^k}{\partial a_{ij}} = \sum_{i=1}^{n_1} \frac{\partial \Lambda^k}{\partial \lambda_i} \frac{\partial \lambda_i}{\partial a_{ij}}. \qquad (5.35)$$

Thus, the partial derivative of the one-norm with respect to $\mathbf{A}_1$ can be found element-by-element from

$$\frac{\partial \|\mathbf{T}_{mr}\|_1}{\partial \mathbf{A}_{1_{i,j}}} = \sum_{k=0}^{N} \operatorname{sgn}(\mathbf{C}_m \mathbf{A}_1^k \mathbf{B}_r) \mathbf{C}_m \frac{\partial \mathbf{A}_1^k}{\partial a_{ij}} \mathbf{B}_r . \qquad (5.36)$$

Recall that the sign of $\mathbf{C}_m \mathbf{A}_1^k \mathbf{B}_r$ for all k is known from the new method of doing the one-norm calculation. Therefore, the only information required to evaluate (5.36) is the $n_1 \times n_1$ diagonal matrix, $\Lambda^k$, at each index k. However, since only the main diagonal of this matrix changes, a $1 \times n_1$ vector of the diagonal elements at each k provides sufficient information to reconstruct $\Lambda^k$. These vectors can be computed rapidly using *ltitr.m*. The resulting $N \times n_1$ matrix only needs to be calculated once in the gradient routine regardless of the number of inputs and outputs in the system. Further, once this matrix is calculated, the remaining one-norm derivatives in (5.27)-(5.33) can be quickly evaluated.

One-norm gradients for MIMO systems are calculated for each output by summing the gradients of the individual SISO transfer functions between the output and the separate inputs. As mention in Section 4.5, this approach ensures that continuous gradient information is available regardless of where the maximum row sum occurs.

Both methods of calculating the one-norm gradients were tested on the benchmark problem described in the last section. The first method took 28.2 hours to complete one $\ell_1$ gradient calculation on a SUN™ SPARC 20 computer. The second method took 5.03 seconds. It should be noted that the first method works relatively fast for systems where N is small. However, larger N values are required for systems with both fast and slow dynamics. For example, the AFTI F-16 design problem presented in Section 5.7 required N values of at least 5,000.

The new approach to the one-norm gradient calculation requires that the eigenvalues of $\mathbf{A}_1$ be distinct. If this condition is violated, the gradient routine switches to

finite difference calculations based on the new approach of computing the $\ell_1$ norm. While less accurate than the old method of computing the gradients of the one-norm, the finite difference calculations are still considerably faster.

## 5.6. Computer Implementation

The author's $\ell_1$ norm and gradient routines were added to the fixed-order, mixed-norm optimization algorithm created by Jacques ([JR94]). This algorithm is capable of running several different types of discrete optimization problems with different SQP routines. Section 5.7 demonstrates the $H_2/\ell_1$ portion of the algorithm using the MATLAB™ SQP routine, *constr.m*.

## 5.7. $H_2/\ell_1$ Design Example

The $H_2/\ell_1$ optimization method is demonstrated on a discrete version of the $H_2/L_1$ design problem presented in Section 4.7. The state-space matrices for the discrete F-16 plant are given in Appendix A. The discrete state-space matrices for the $H_2$ and $\ell_1$ subproblems are shown in Appendix C.

## 5.7.1. $H_2/\ell_1$ Results

A plot of the $H_2/\ell_1$ solution curve for a 5th order compensator is shown in Figure 5.1. Values for several of the solution points are shown in Table 5.1. Point number 17 represents the discrete $H_2$ optimal solution, which is not depicted in Figure 5.1. The points in Table 5.1 are referred to as the discrete design points for the remainder of this section.
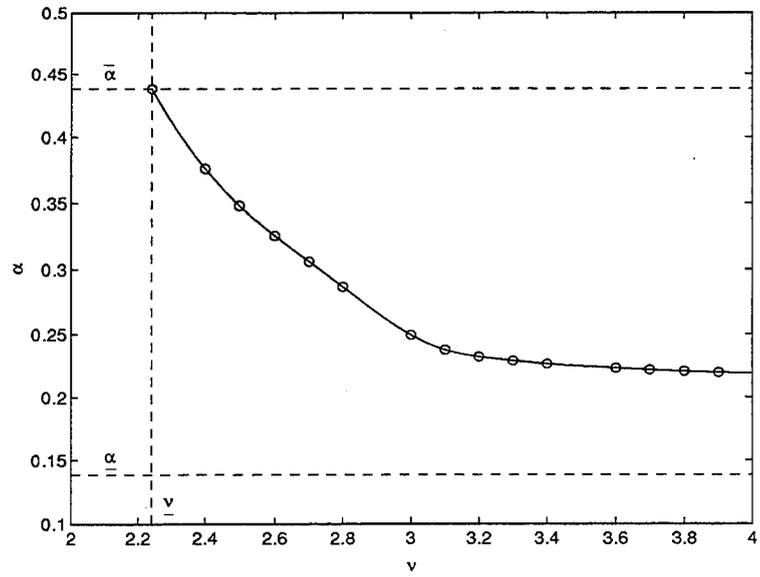
Figure 5.1. F-16 $H_2/\ell_1$ solution curve

| point # | $\alpha$ | $\nu$ |
|---------|----------|--------|
| 1 | 0.44 | 2.24 |
| 4 | 0.33 | 2.60 |
| 7 | 0.25 | 3.00 |
| 11 | 0.23 | 3.40 |
| 14 | 0.22 | 3.80 |
| 17 | 0.14 | 46,672 |

Table 5.1. F-16 solution points for $H_2/\ell_1$ design

Plots of the sensitivity and complementary sensitivity for the discrete design points are shown in Figures 5.2 and 5.3.
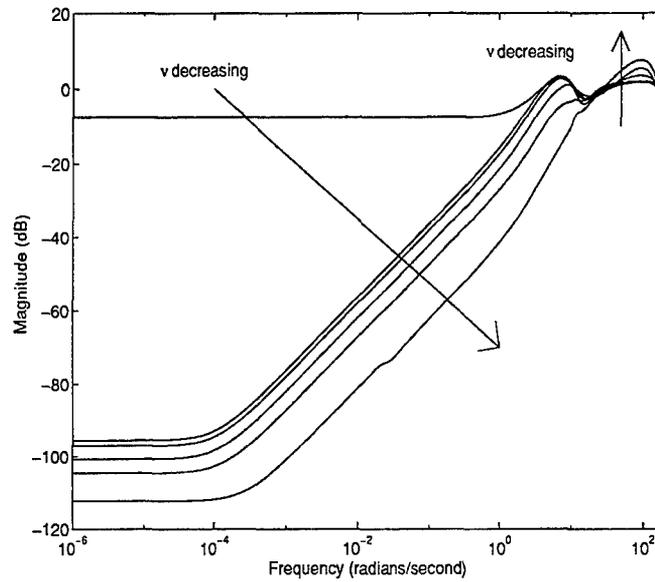


Figure 5.2. F-16 sensitivity from discrete design
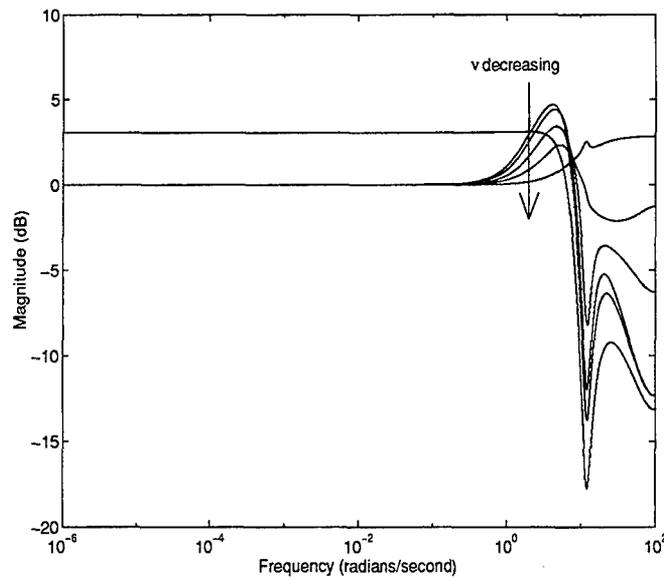


Figure 5.3. F-16 complementary sensitivity from discrete design

The distinctively different curve in each plot corresponds to the $H_2$ solution.

The vector gain and phase margins for the discrete design points are shown in Table 5.2.

| point # | lower VGM (dB) | upper VGM (dB) | VPM (deg) |
|---------|----------------|----------------|-----------|
| 1 | -11.2 | 4.73 | 42.4 |
| 4 | -12.6 | 6.70 | 45.0 |
| 7 | -9.80 | 9.73 | 39.5 |
| 11 | -8.01 | 10.8 | 41.7 |
| 14 | -7.61 | 9.97 | 39.9 |
| 17 | -10.5 | 11.4 | 42.9 |

Table 5.2. F-16 vector gain and phase margins from discrete design

Notice that the stability margins do not consistently improve as v decreases, but do remain acceptable for all the discrete design points.

The open-loop GK plot is shown in Figure 5.4
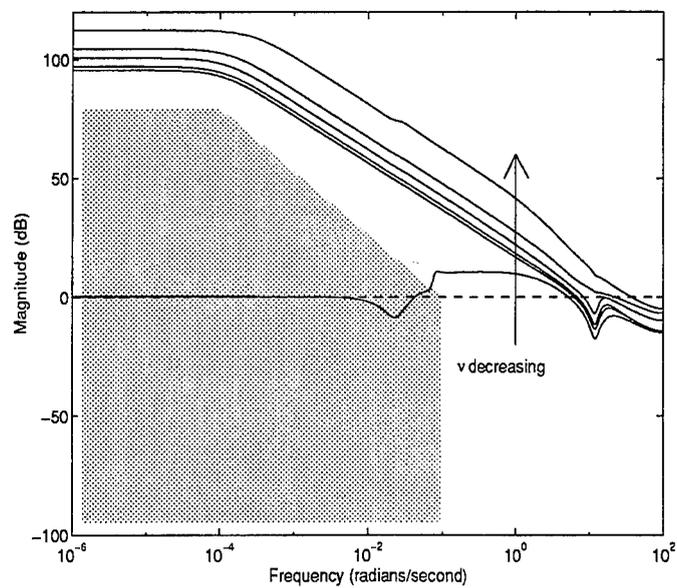


Figure 5.4. F-16 open-loop GK from discrete design

Notice that the mixed designs miss the performance and disturbance rejection barrier. The sensor noise and unmodeled dynamics barrier is not depicted since digital aliasing would cause it to be violated. An anti-aliasing filter (basically a low-pass filter) can be added to the system to produce any amount of high frequency roll-off.

Figure 5.5 shows the F-16 responses, without noise, to a commanded 1g step input for the different designs. Notice that the mixed designs track with no steady-state error. Step responses with noise for discrete design points 1, 7, 14 and 17 are shown in Figures 5.6-5.9. These plots demonstrate that tracking performance can be greatly improved using mixed $H_2/l_1$ designs with very little increase of noise in the system response. High frequency noise is not as prevalent in these plots as it was in the continuous simulations due to the sampling of the discrete controller.
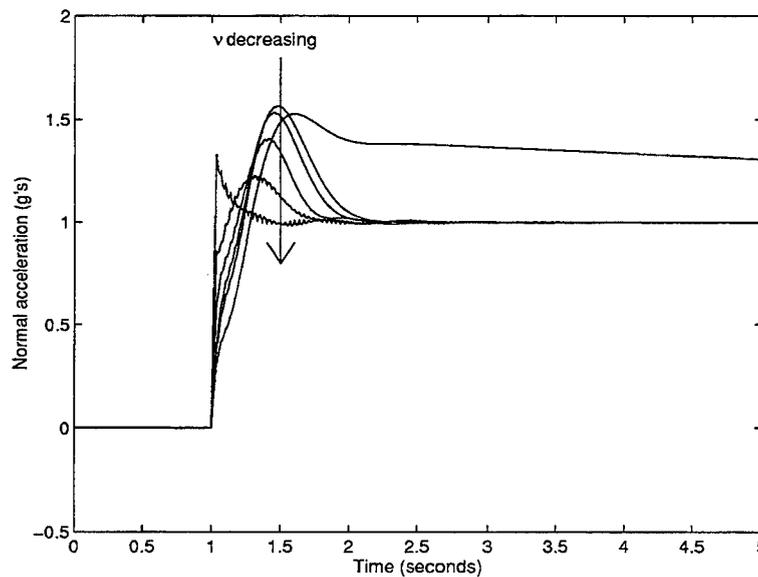


Figure 5.5. F-16 step responses without noise, from discrete design
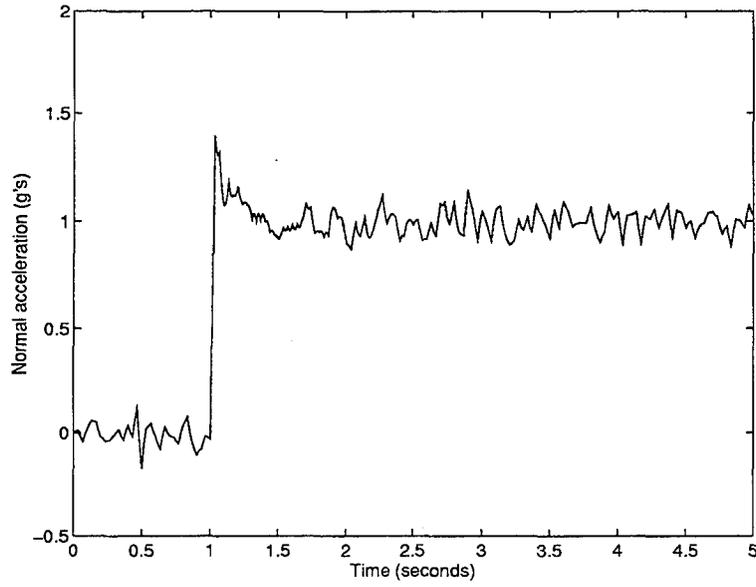
Figure 5.6. F-16 step response with noise, discrete design point 1
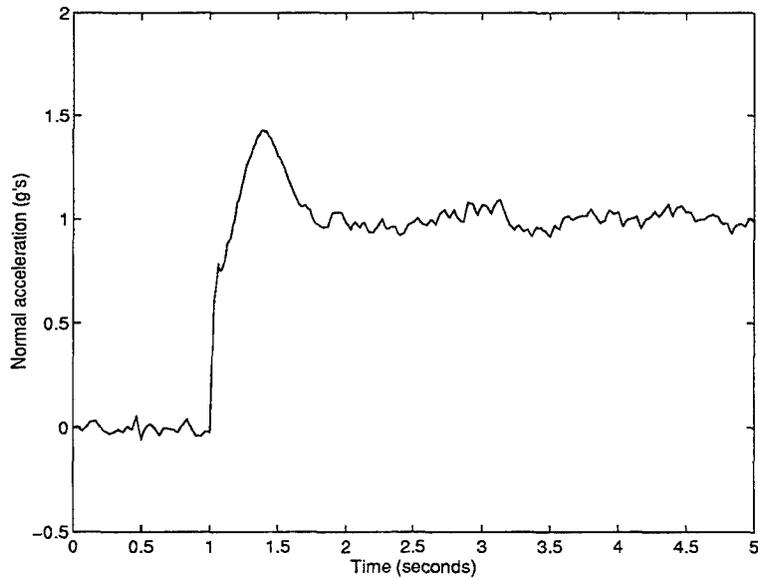


Figure 5.7. F-16 step response with noise, discrete design point 7
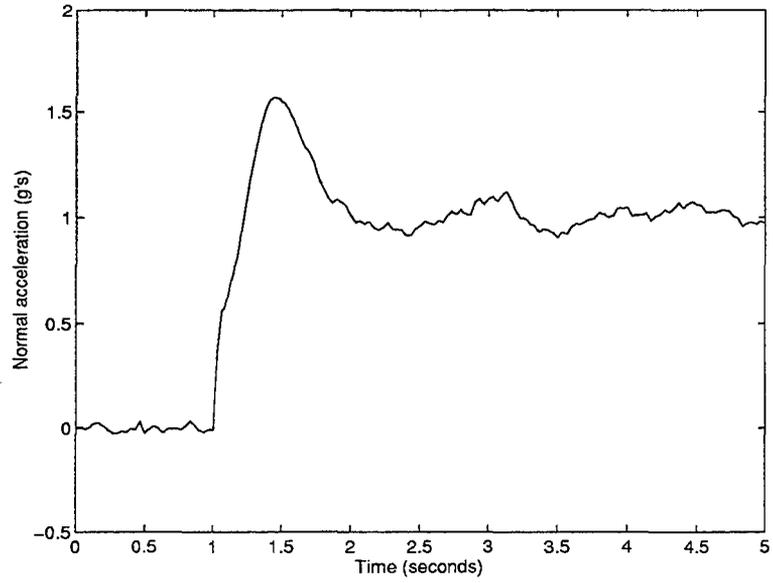
Figure 5.8. F-16 step response with noise, discrete design point 14
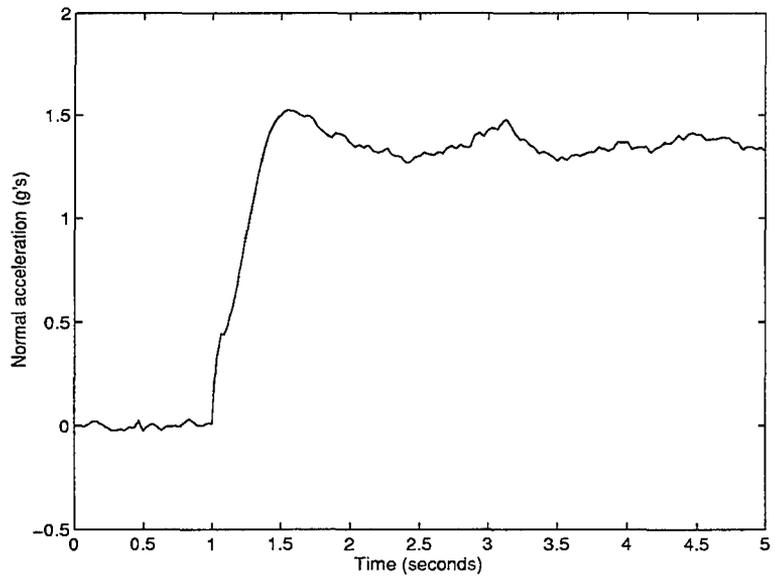


Figure 5.9. F-16 step response with noise, discrete design point 17

Discrete design point 7 is chosen as the best mixed $H_2/l_1$ design based on stability margins, loop shape and step responses. The control usage for this particular discrete design point is shown in Figure 5.10. Discrete design point 7 is compared to the best $H_2/L_1$ results from Chapter 4 in the next section.
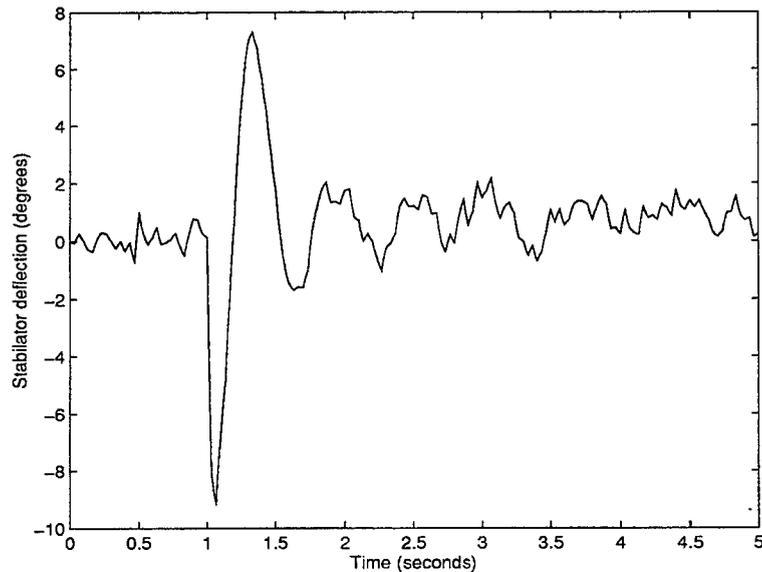


Figure 5.10. F-16 control usage with noise, discrete design point 7

## 5.7.2. Comparing $H_2/L_1$ to $H_2/l_1$

It is impossible to pick an equivalent point from the mixed $H_2/L_1$ solutions and the mixed $H_2/l_1$ solutions for direct comparison. However, some general conclusions can be drawn about the two methods. To illustrate the points in this section, the AFTI F-16 is tested with the best $H_2/L_1$ controller discretized at 30Hz using a ZOH. The step response and control usage are shown in Figures 5.11 and 5.12, respectively.
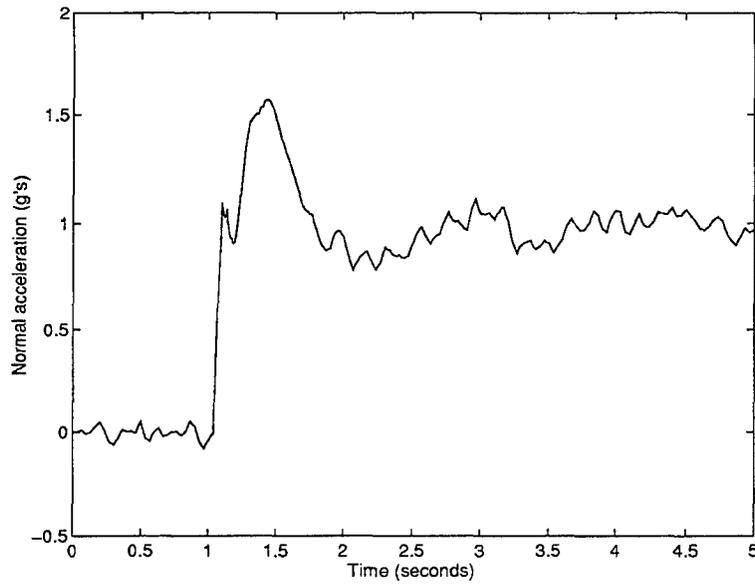
Figure 5.11. F-16 step response with noise, discretized controller from $H_2/L_1$ design point 15
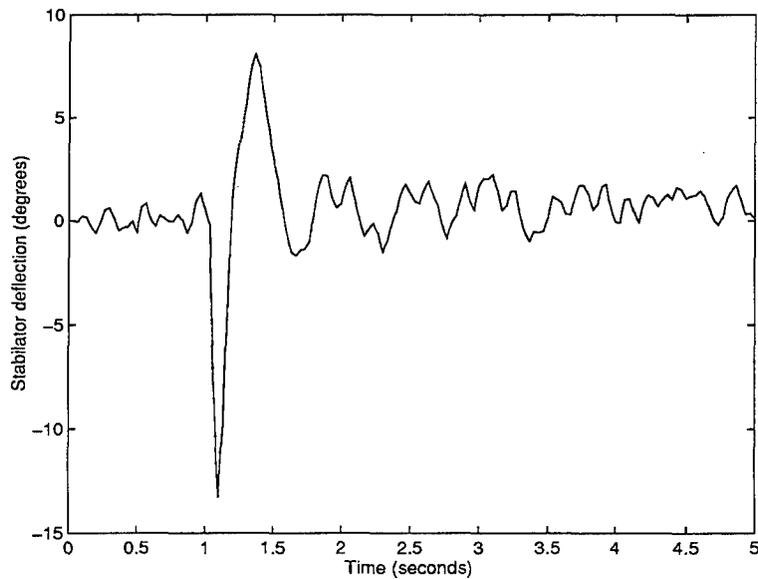


Figure 5.12. F-16 control usage with noise, discretized controller from $H_2/L_1$ design point15

Notice that the discretized controller does not produce the same level of performance as it did prior to discretization (see Figures 4.13 and 4.15). The system with the discretized controller has larger responses to noise.

The responses in Figures 5.11 and 5.12 are similar to the responses for the best $H_2/\ell_1$ controller (Figures 5.7 and 5.10) in terms of peak values, peak times and settling times. This suggests that comparable designs could be done with either method. However, the $H_2/L_1$ controller is 6th order and the $H_2/\ell_1$ controller is 5th order. If compensator orders are chosen to be the size of the $H_2$ plant in either design, then $H_2/L_1$ optimization will always produce a higher order compensator. This is due to the additional dynamics that must added to the continuous design to represent the time delay that will occur when the controller is discretized (see Appendix A).

The Padé approximations used to represent the time delay in the $H_2/L_1$ design can cause additional problems. First, they make the continuous system nonminimum phase. This can be seen in the $H_2/L_1$ step responses in Figure 4.10, where all initially respond in the opposite direction of the commanded normal acceleration. Since the actual aircraft model is minimum phase without the Padé approximations, these plots can be misleading. Second, large order Padé approximations are required to accurately represent a time delay, which can lead to higher order controllers in an $H_2/L_1$ design. If a lower order Padé approximation is used, care must taken to discretize the continuous controller at a slightly faster period than the delay time represented by the approximation. Discretizing the controller at a period equal to the delay time represented by the approximation can lead to an unstable closed-loop system. In light of the issues covered above, it can be concluded that digital controller designs are better accomplished with $H_2/\ell_1$ optimization.

## 5.8. Chapter Summary

This chapter presented a numerical approach to $H_2/\ell_1$ optimization for discrete systems. New and efficient methods of approximating the one-norm and its gradients

were also presented. The $H_2/l_1$ algorithm was tested on a realistic problem which showed the benefits of this mixed-norm design approach. Comparisons of systems with discrete $H_2/l_1$ controllers and discretized $H_2/L_1$ controllers showed that adequate digital controllers are better obtained with $H_2/l_1$ optimization. Chapter 6 summarizes the results in this thesis and presents recommendations for further research.

# VI. Conclusions and Recommendations

## 6.1. Summary

This thesis explored the use of $\ell_1$ optimization for flight controller designs. Methods of using $\ell_1$ optimization were developed to handle "hard" magnitude constraints, such as control deflection limitations, control rate limitations and overshoot restrictions which are difficult to incorporate into other optimization methods. These constraints were added to the standard free-order $\ell_1$ optimization problem and demonstrated on a realistic flight control design with mixed success. The $\ell_1$ optimization algorithm worked well for small systems with one or two constraints, but had difficulty handling very large systems of equations with multiple constraints. The constrained $\ell_1$ optimization problem also had the tendency to produce compensators with unreasonably large orders. To counter this problem, two model matching techniques were presented based on one block systems. These one block problems were solved quickly and accurately with the current $\ell_1$ optimization software, and acceptable tracking results were produced for SISO and MIMO systems with controller orders less than or equal to the weighted plant.

$H_2$ optimization was mixed with $\ell_1$ optimization to produce systems with good noise performance and tracking characteristics. This method was done first in continuous time with fixed-order $H_2/L_1$ optimization. New numerical approaches of calculating the $L_1$ norm and its gradient were developed and incorporated into the $H_2/L_1$ algorithm which avoid many of the complications associated with previous methods dependent on the forward Euler transformation. The $H_2/L_1$ algorithm was tested on a SISO system and the results showed the benefits of a mixed design over a pure $H_2$ or $L_1$ design. This method had the added benefit that the compensator order is completely chosen by the designer.

Fixed-order $H_2/\ell_1$ optimization was then used to design discrete systems. New numerical approaches of calculating the $\ell_1$ norm and its gradient were presented which offered dramatic improvements in terms of computational efficiency, and essentially

extended the utility of the existing $H_2/\ell_1$ algorithm to a wider class of problems. The $H_2/\ell_1$ algorithm was tested on a realistic SISO system that could not have been solved prior to the improvement in the algorithm. The results showed that a good blend of tracking and noise performance was easy to obtain with mixed $H_2/\ell_1$ optimization. Like the $H_2/L_1$ method, the size of the controller in $H_2/\ell_1$ optimization is chosen by the designer.

Two different methods of digital controller designs were examined in the context of $H_2/L_1$ optimization and $H_2/\ell_1$ optimization. The first method involved completing a satisfactory $H_2/L_1$ design and discretizing the resulting controller. In the second method, the discrete controller was obtained directly with $H_2/\ell_1$ optimization. Comparisons of the two approaches showed that comparable designs could be done with either method, but the $H_2/\ell_1$ approach would generally produce a smaller order controller due to the additional dynamics required in the $H_2/L_1$ problem to approximate a time delay. Since the $H_2/\ell_1$ algorithm is also quicker and more accurate than the $H_2/L_1$ algorithm, it can be concluded that fixed-order digital controller design is best done with $H_2/\ell_1$ optimization.

Of all the methods presented in this thesis, fixed-order $H_2/\ell_1$ optimization offers the most promise for successful designs of small order digital flight controllers that produce systems with good noise and tracking performance. However, research still needs to be done on free-order $H_2/\ell_1$ optimization to determine the amount of performance lost in fixed-order $H_2/\ell_1$ optimization by setting the size of the controller a priori. Additional research topics are discussed in the next section.

### 6.2. Recommendations for Future Research

While this thesis has contributed to a better understanding of how $\ell_1$ and mixed $H_2/\ell_1$ optimization can be applied, more research needs to be done in these two areas. First, the $\ell_1$ optimization algorithm needs to be improved to allow it to handle large systems with multiple constraints. This could probably be accomplished with an improved linear programming algorithm and appropriate scaling of the design variables. Additional

improvements are required to increase the speed of the $l_1$ algorithm, and to simplify the way constraints are augmented to a standard $l_1$ design problem. Further research in $l_1$ frequency constraints (not discussed in this thesis) should also be conducted.

Second, many of the "hard" constraints incorporated into $l_1$ optimization should be incorporated into $H_2/l_1$ optimization. This should not be difficult to do given the present structure of the $H_2/l_1$ algorithm. The added capabilities, however, will greatly improve the utility of the $l_1$ portion of the $H_2/l_1$ algorithm.

Third, the $H_2/l_1$ algorithm requires several modifications before it can be reliably used on wide range of problems. Research needs to be done to allow the analytical $l_1$ gradients to handle systems with repeated eigenvalues. The current version of the software relies on finite difference calculations in this case. Repeated eigenvalues are also not permitted in the resulting $H_2/l_1$ controller. This problem could be fixed by allowing a Jordan form for the controller instead of restricting it to modal form. Further, scaling problems in first and second order derivatives in the MATLAB™ SQP routine, *constr.m*, restrict the ability of the $H_2/l_1$ algorithm to quickly generate a complete design curve (such as the one depicted in Figure 5.1). While the IMSL™ SQP routine alleviates some of these problems, it is commercial software and thus can not be modified to suit specific requirements.

Finally, further research is needed on mixed $H_2/l_1/H_\infty$ optimization for discrete systems. The $H_\infty$ portion of the problem could be used to improve stability margins with sensitivity minimization. This capability was not possible with $l_1$ sensitivity minimization. The $l_1$ portion of the problem could then be used to handle the "hard" magnitude constraints of the system. Work is currently underway in this area.

## Appendix A.  SISO AFTI F-16 Model

The AFTI F-16 model for normal acceleration command following is shown in Figure A.1. The continuous system consists of a 4 state model of the aircraft's longitudinal dynamics ($W_p$), a first order actuator ($W_a$) and a first order Padé approximation of a 0.05 second time delay ($W_t$). The time delay model is not required in the discrete version of the system.
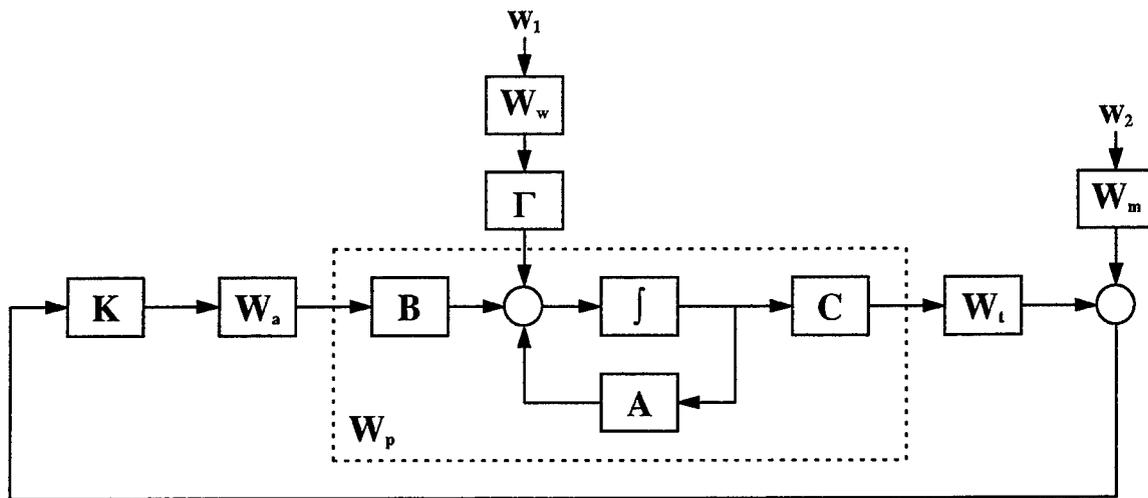


Figure A.1.  F-16 model block diagram

The four states in the longitudinal model are forward speed (u in ft/sec), angle of attack ($\alpha$ in radians), pitch angle ($\theta$ in radians), and pitch rate (q in radians/sec). The input to $W_p$ is the stabilator deflection ($\delta_e$ in radians) and the output is the normal acceleration ($n_z$ in g's). $W_p$ is given by

$$\begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -1.485e-2 & 3.738e+1 & -3.220e+1 & -1.794e+1 \\ -8.000e-5 & -1.491e+0 & -1.300e-3 & 9.960e-1 \\ 0.000e+0 & 0.000e+0 & 0.000e+0 & 1.000e+0 \\ -3.600e-4 & 9.753e+0 & 2.900e-4 & -9.600e-1 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ \theta \\ q \end{bmatrix} + \begin{bmatrix} 2.140e-3 \\ -1.880e-1 \\ 0.000e+0 \\ -1.904e+1 \end{bmatrix} \delta_e$$

$$n_z = \begin{bmatrix} 1.500e-3 & 3.5264e+1 & 2.720e-2 & -3.340e-1 \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ \theta \\ q \end{bmatrix} + \begin{bmatrix} -4.366e+0 \end{bmatrix} \delta_e.$$

The input to $W_a$ is the commanded stabilator deflection ($\delta_{e_c}$ in radians) and the output is the stabilator deflection. $W_a$ is given by

$$\dot{x}_a = \begin{bmatrix} -2.000e+1 \end{bmatrix} x_a + \begin{bmatrix} 2.000e+1 \end{bmatrix} \delta_{e_c}$$
$$\delta_e = \begin{bmatrix} 1.000e+0 \end{bmatrix} x_a + \begin{bmatrix} 0.000e+0 \end{bmatrix} \delta_{e_c}.$$

The input to $W_t$ is normal acceleration and the output is the delayed normal acceleration ($n_{z_d}$ in g's). $W_t$ is given by

$$\dot{x}_t = \begin{bmatrix} -4.000e+1 \end{bmatrix} x_t + \begin{bmatrix} 1.000e+0 \end{bmatrix} n_z$$
$$n_{z_d} = \begin{bmatrix} 8.000e+1 \end{bmatrix} x_t + \begin{bmatrix} -1.000e+0 \end{bmatrix} n_z.$$

The continuous aircraft plant, $G_c$, equals $W_a W_p W_t$. $G_c$ is given by

$$G_c = \begin{bmatrix} A_c & B_c \\ \hline C_c & D_c \end{bmatrix}$$

where

$$A_c = \begin{bmatrix} -1.485e-2 & 3.738e+1 & -3.220e+1 & -1.794e+1 & 2.140e-3 & 0.000e+0 \\ -8.000e-5 & -1.491e+0 & -1.300e-3 & 9.960e-1 & -1.880e-1 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 & 0.000e+0 & 1.000e+0 & 0.000e+0 & 0.000e+0 \\ -3.600e-4 & 9.753e+0 & 2.900e-4 & -9.600e-1 & -1.904e+1 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 & 0.000e+0 & 0.000e+0 & -2.000e+1 & 0.000e+0 \\ 1.500e-3 & 3.526e+1 & 2.720e-2 & -3.340e-1 & -4.366e+0 & -4.000e+0 \end{bmatrix}$$

$$B_c = \begin{bmatrix} 0.000e+0 \\ 0.000e+0 \\ 0.000e+0 \\ 0.000e+0 \\ 2.000e+1 \\ 0.000e+0 \end{bmatrix}$$

$$C_c = \begin{bmatrix} -1.500e-3 & -3.526e+1 & -2.720e-2 & 3.340e-1 & 4.366e+0 & 8.000e+1 \end{bmatrix}$$

$$D_c = \begin{bmatrix} 0.000e+0 \end{bmatrix}.$$

The discrete aircraft plant, $G_d$, equals $W_a W_p$ discretized at 30Hz using a ZOH. $G_d$ is given by

$$G_d = \left[ \begin{array}{c|c} A_d & B_d \\ \hline C_d & D_d \end{array} \right]$$

where

$$A_d = \begin{bmatrix} 9.995e-1 & 1.121e+0 & -1.073e+0 & -5.870e-1 & 1.489e-1 \\ -2.800e-6 & 9.567e-1 & -4.072e-5 & 3.193e-2 & -1.276e-2 \\ -2.028e-7 & 5.278e-3 & 1.000e+0 & 3.286e-2 & -8.494e-3 \\ -1.225e-5 & 3.127e-1 & 9.198e-6 & 9.737e-1 & -4.568e-1 \\ 0.000e+0 & 0.000e+0 & 0.000e+0 & 0.000e+0 & 5.134e-1 \end{bmatrix}$$

$$B_d = \begin{bmatrix} 3.494e-2 \\ -3.623e-3 \\ -1.992e-3 \\ -1.700e-1 \\ 4.866e-1 \end{bmatrix}$$

$$C_d = \begin{bmatrix} 1.500e-3 & 3.526e+1 & 2.720e-2 & -3.340e-1 & -4.366e+0 \end{bmatrix}$$

$$D_d = \begin{bmatrix} 0.000e+0 \end{bmatrix}.$$

The truth model used in simulations includes a Von Karman wind model ($W_w$) for the wind noise, and a high-pass filter ($W_m$) for the measurement noise. $W_w$ is given by

$$\dot{x}_w = [-6.700e + 0]x_w + [1.870e - 3]w_1$$
$$\xi_1 = [1.000e + 0]x_w.$$

where $w_1$ is a unit strength white Gaussian noise, and $\xi_1$ is the wind noise. $W_m$ is given by

$$\dot{x}_m = [-1.000e + 1]x_m + [4.000e - 4]w_2$$
$$\xi_2 = [1.000e + 0]x_m + [4.000e - 4]w_2.$$

where $w_2$ is a unit strength white Gaussian noise, and $\xi_2$ is the measurement noise. The wind noise enters the aircraft plant as an angle of attack perturbation. $\Gamma$, in Figure A.1, equals the second column of the A matrix in the longitudinal model.

## Appendix B.  MIMO Missile Model

The 3 input, 3 output missile model (taken from [Bro91] and [Luk93]) for normal and lateral acceleration command following is shown in Figure B.1.  The system consists of a 5 state model of the missile's lateral and longitudinal dynamics ($W_p$), and a 3 state model of the actuator dynamics ($W_a$).
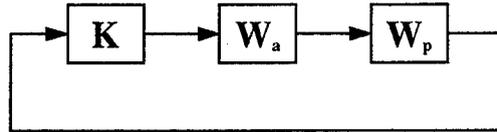


Figure B.1.  Missile model block diagram

The 5 states in $W_p$ are angle of attack ($\alpha$ in radians), sideslip angle ($\beta$ in radians), roll rate (p radians/sec), pitch rate (q radians/sec), and yaw rate (r radians/sec).  The inputs to $W_p$ are the control fin deflections ($\delta_p$, $\delta_q$, $\delta_r$ in radians) and the outputs are the roll rate, lateral acceleration ($n_y$ in ft/sec$^2$), and normal acceleration ($n_z$ in ft/sec$^2$).  $W_p$ is given by

$$
\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -9.844e-1 & -9.234e-2 & 0.000e+0 & 1.000e+0 & 0.000e+0 \\ -9.234e-2 & -9.844e-1 & 0.000e+0 & 0.000e+0 & -1.000e+0 \\ -2.674e+2 & 2.674e+2 & -1.595e+0 & 0.000e+0 & 0.000e+0 \\ -1.946e+2 & -6.967e-1 & 0.000e+0 & -1.550e+0 & 0.000e+0 \\ 6.967e-1 & 1.946e+2 & 0.000e+0 & 0.000e+0 & -1.550e+0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ p \\ q \\ r \end{bmatrix}
$$

$$
+ \begin{bmatrix} -2.186e-3 & 3.261e-1 & -1.135e-3 \\ -6.189e-4 & 1.566e-3 & -3.188e-1 \\ 1.140e+4 & -6.550e+1 & -1.070e+1 \\ -2.736e+0 & 3.211e+2 & 1.451e-1 \\ -3.045e-1 & 1.410e+0 & 3.149e+2 \end{bmatrix} \begin{bmatrix} \delta_p \\ \delta_q \\ \delta_r \end{bmatrix}
$$

B-1

$$
\begin{bmatrix} p \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} 0.000e+0 & 0.000e+0 & 1.000e+0 & 0.000e+0 & 0.000e+0 \\ -2.062e+2 & -2.198e+3 & 0.000e+0 & 0.000e+0 & 0.000e+0 \\ -2.198e+3 & -2.062e+2 & 0.000e+0 & 0.000e+0 & 0.000e+0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ p \\ q \\ r \end{bmatrix}
$$

$$
+ \begin{bmatrix} 0.000e+0 & 0.000e+0 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 & 0.000e+0 \\ -1.368e+0 & 3.496e+0 & -7.119e+2 \\ -4.876e+0 & 7.282e+2 & -2.535e+0 \end{bmatrix} \begin{bmatrix} \delta_p \\ \delta_q \\ \delta_r \end{bmatrix}
$$

The inputs to $W_a$ are the commanded fin deflections ($\delta_{pc}$, $\delta_{qc}$, $\delta_{rc}$ in radians) and the outputs are the actual fin deflections. $W_a$ is given by

$$
\begin{bmatrix} \dot{x}_{a1} \\ \dot{x}_{a2} \\ \dot{x}_{a3} \end{bmatrix} = \begin{bmatrix} -4.000e+2 & 0.000e+0 & 0.000e+0 \\ 0.000e+0 & -4.000e+2 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 & -4.000e+2 \end{bmatrix} \begin{bmatrix} x_{a1} \\ x_{a2} \\ x_{a3} \end{bmatrix}
$$

$$
+ \begin{bmatrix} 1.000e+0 & 0.000e+0 & 0.000e+0 \\ 0.000e+0 & 1.000e+0 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 & 1.000e+0 \end{bmatrix} \begin{bmatrix} \delta_{pc} \\ \delta_{qc} \\ \delta_{rc} \end{bmatrix}
$$

$$
\begin{bmatrix} \delta_p \\ \delta_q \\ \delta_r \end{bmatrix} = \begin{bmatrix} 4.000e+2 & 0.000e+0 & 0.000e+0 \\ 0.000e+0 & 4.000e+2 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 & 4.000e+2 \end{bmatrix} \begin{bmatrix} x_{a1} \\ x_{a2} \\ x_{a3} \end{bmatrix}
$$

$$
+ \begin{bmatrix} 0.000e+0 & 0.000e+0 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 & 0.000e+0 \end{bmatrix} \begin{bmatrix} \delta_{pc} \\ \delta_{qc} \\ \delta_{rc} \end{bmatrix}
$$

The discrete aircraft plant, $G_d$, equals $W_a W_p$ discretized at 100Hz using a ZOH. $G_d$ is given by

$$
G_d = \begin{bmatrix} A_d & B_d \\ \hline C_d & D_d \end{bmatrix}
$$

where

$$A_d = \begin{bmatrix} 1.832e-2 & 0.000e+0 & 0.000e+0 & 0.000e+0 & \cdots \\ 0.000e+0 & 1.832e-2 & 0.000e+0 & 0.000e+0 & \cdots \\ 0.000e+0 & 0.000e+0 & 1.832e-2 & 0.000e+0 & \cdots \\ -2.250e-2 & 2.708e+0 & 1.121e-3 & 9.806e-1 & \cdots \\ 1.679e-3 & -1.015e-2 & -2.655e+0 & -9.428e-4 & \cdots \\ 1.105e+4 & -6.682e+1 & -1.360e+1 & -2.632e+0 & \cdots \\ -2.634e+0 & 3.091e+2 & 1.492e-1 & -1.916e+0 & \cdots \\ -2.945e-1 & 1.369e+0 & 3.032e+2 & 5.950e-3 & \cdots \end{bmatrix}$$

$$\begin{bmatrix} 0.000e+0 & 0.000e+0 & 0.000e+0 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 & 0.000e+0 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 & 0.000e+0 & 0.000e+0 \\ -9.428e-4 & 0.000e+0 & 9.842e-3 & 4.663e-6 \\ 9.806e-1 & 0.000e+0 & -4.663e-6 & -9.842e-3 \\ 2.632e+0 & 9.842e-1 & -1.317e-2 & -1.317e-2 \\ -5.950e+0 & 0.000e+0 & 9.750e-1 & 3.130e-5 \\ 1.916e+0 & 0.000e+0 & 3.130e-5 & 9.750e-1 \end{bmatrix}$$

$$B_d = \begin{bmatrix} 2.4542e-3 & 0.000e+0 & 0.000e+0 \\ 0.000e+0 & 2.4542e-3 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 & 2.4542e-3 \\ -1.009e-4 & 1.236e-2 & -5.034e-7 \\ 4.790e-6 & -3.539e-5 & -1.211e-2 \\ 8.541e+1 & -5.013e-1 & -9.025e-2 \\ -2.045e-2 & 2.400e+0 & 1.117e-3 \\ -2.282e-3 & 1.058e-2 & 2.354e+0 \end{bmatrix}$$

$$C_d = \begin{bmatrix} 0.000e+0 & 0.000e+0 & 0.000e+0 & 0.000e+0 & \cdots \\ -5.474e+2 & 1.398e+3 & -2.848e+5 & -2.062e+2 & \cdots \\ -1.950e+3 & 2.913e+5 & -1.014e+3 & -2.1979e+3 & \cdots \end{bmatrix}$$

$$\begin{bmatrix} 0.000e+0 & 1.000e+0 & 0.000e+0 & 0.000e+0 \\ -2.198e+3 & 0.000e+0 & 0.000e+0 & 0.000e+0 \\ -2.062e+2 & 0.000e+0 & 0.000e+0 & 0.000e+0 \end{bmatrix}$$

$$D_d = \begin{bmatrix} 0.000e+0 & 0.000e+0 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 & 0.000e+0 \end{bmatrix}$$

*Appendix C. Matrices for $H_2/L_1$ and $H_2/\ell_1$ Subproblems*

The matrices for the continuous $H_2$ problem are:

$$A_2 = \begin{bmatrix} -1.485e-2 & 3.738e+1 & -3.220e+1 & -1.794e+1 & 2.140e-3 & 0.000e+0 \\ -8.000e-5 & -1.491e+0 & -1.300e-3 & 9.960e-1 & -1.880e-1 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 & 0.000e+0 & 1.000e+0 & 0.000e+0 & 0.000e+0 \\ -3.600e-4 & 9.753e+0 & 2.900e-4 & -9.600e-1 & -1.904e+1 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 & 0.000e+0 & 0.000e+0 & -2.000e+1 & 0.000e+0 \\ 1.500e-3 & 3.526e+1 & 2.720e-2 & -3.340e-1 & -4.366e+0 & -4.000e+1 \end{bmatrix}$$

$$B_w = \begin{bmatrix} 8.359e-1 & 0.000e+0 \\ -3.334e-2 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 \\ 2.181e-1 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 \end{bmatrix} \qquad B_{u2} = \begin{bmatrix} 0.000e+0 \\ 0.000e+0 \\ 0.000e+0 \\ 0.000e+0 \\ 2.000e+1 \\ 0.000e+0 \end{bmatrix}$$

$$C_z = \begin{bmatrix} -1.500e-3 & -3.526e+1 & -2.720e-2 & 3.340e-1 & 4.366e+0 & 8.000e+1 \\ 0.000e+0 & 0.000e+0 & 0.000e+0 & 0.000e+0 & 0.000e+0 & 0.000e+0 \end{bmatrix}$$

$$C_{y2} = \begin{bmatrix} -1.500e-3 & -3.526e+1 & -2.720e-2 & 3.340e-1 & 4.366e+0 & 8.000e+1 \end{bmatrix}$$

$$D_{zw} = \begin{bmatrix} 0.000e+0 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 \end{bmatrix} \qquad D_{zu} = \begin{bmatrix} 0.000e+0 \\ 1.000e+1 \end{bmatrix}$$

$$D_{yw} = \begin{bmatrix} 0.000e+0 & 4.000e-3 \end{bmatrix} \qquad D_{yu} = \begin{bmatrix} 0.000e+0 \end{bmatrix}$$

The matrices for the $L_1$ problem are

$$A_1 = \begin{bmatrix} -1.485e-2 & 3.738e+1 & -3.220e+1 & \cdots \\ -8.000e-5 & -1.491e+0 & -1.300e-3 & \cdots \\ 0.000e+0 & 0.000e+0 & 0.000e+0 & \cdots \\ -3.600e-4 & 9.753e+0 & 2.900e-4 & \cdots \\ 0.000e+0 & 0.000e+0 & 0.000e+0 & \cdots \\ 1.500e-3 & 3.526e+1 & 2.720e-2 & \cdots \\ -1.500e-3 & -3.526e+1 & -2.720e-2 & \cdots \end{bmatrix}$$

$$\begin{bmatrix} -1.794e+1 & 2.140e-3 & 0.000e+0 & 0.000e+0 \\ 9.960e-1 & -1.880e-1 & 0.000e+0 & 0.000e+0 \\ 1.000e+0 & 0.000e+0 & 0.000e+0 & 0.000e+0 \\ -9.600e-1 & -1.904e+1 & 0.000e+0 & 0.000e+0 \\ 0.000e+0 & -2.000e+1 & 0.000e+0 & 0.000e+0 \\ -3.340e-1 & -4.366e+0 & -4.000e+1 & 0.000e+0 \\ 3.340e-1 & 4.366e+0 & 8.000e+1 & -1.000e-4 \end{bmatrix}$$

$$B_r = \begin{bmatrix} 0.000e+0 \\ 0.000e+0 \\ 0.000e+0 \\ 0.000e+0 \\ 0.000e+0 \\ 0.000e+0 \\ 1.000e+0 \end{bmatrix} \qquad B_{u1} = \begin{bmatrix} 0.000e+0 \\ 0.000e+0 \\ 0.000e+0 \\ 0.000e+0 \\ 2.000e+1 \\ 0.000e+0 \\ 0.000e+0 \end{bmatrix}$$

$$C_m = \begin{bmatrix} -1.500e-3 & -3.526e+1 & -2.720e-2 & \cdots \\ 3.340e-1 & 4.366e+0 & 8.000e+1 & 1.000e+1 \end{bmatrix}$$

$$C_{y1} = \begin{bmatrix} -1.500e-3 & -3.526e+1 & -2.720e-2 & \cdots \\ 3.340e-1 & 4.366e+0 & 8.000e+1 & 0.000e+0 \end{bmatrix}$$

$$D_{mr} = \begin{bmatrix} 1.000e+0 \end{bmatrix} \qquad D_{mu} = \begin{bmatrix} 0.000e+0 \end{bmatrix}$$

$$D_{yr} = \begin{bmatrix} 1.000e+0 \end{bmatrix} \qquad D_{yu} = \begin{bmatrix} 0.000e+0 \end{bmatrix}$$

The state-space matrices for the discrete $H_2$ problem are

$$A_2 = \begin{bmatrix} 9.9951e-1 & 1.121e+0 & -1.073e+0 & -5.870e-1 & 1.489e-1 \\ -2.800e-6 & 9.567e-1 & -4.072e-5 & 3.193e-2 & -1.276e-2 \\ -2.028e-7 & 5.278e-3 & 1.000e+0 & 3.286e-2 & -8.494e-3 \\ -1.225e-5 & 3.127e-1 & 9.198e-6 & 9.737e-1 & -4.568e-1 \\ 0.000e+0 & 0.000e+0 & 0.000e+0 & 0.000e+0 & 0.000e+0 \end{bmatrix}$$

$$B_w = \begin{bmatrix} 2.506e-2 & 0.000e+0 \\ -9.686e-4 & 0.000e+0 \\ 1.180e-4 & 0.000e+0 \\ 6.991e-3 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 \end{bmatrix} \qquad B_{u2} = \begin{bmatrix} 3.494e-2 \\ -3.623e-3 \\ -1.992e-3 \\ -1.699e-1 \\ 4.865e-1 \end{bmatrix}$$

$$C_z = \begin{bmatrix} -1.500e-3 & -3.564e+1 & -2.720e-2 & 3.340e-1 & 4.366e+0 \\ 0.000e+0 & 0.000e+0 & 0.000e+0 & 0.000e+0 & 0.000e+0 \end{bmatrix}$$

$$C_{y2} = \begin{bmatrix} -1.500e-3 & -3.564e+1 & -2.720e-2 & 3.340e-1 & 4.366e+0 \end{bmatrix}$$

$$D_{zw} = \begin{bmatrix} 0.000e+0 & 0.000e+0 \\ 0.000e+0 & 0.000e+0 \end{bmatrix} \qquad D_{zu} = \begin{bmatrix} 0.000e+0 \\ 1.000e+1 \end{bmatrix}$$

$$D_{yw} = \begin{bmatrix} 0.000e+0 & 4.000e-3 \end{bmatrix} \qquad D_{yu} = \begin{bmatrix} 0.000e+0 \end{bmatrix}$$

The matrices for the $\ell_1$ problem are

$$A_1 = \begin{bmatrix} 9.9951\text{e}-1 & 1.121\text{e}+0 & -1.073\text{e}+0 & -5.870\text{e}-1 & 1.489\text{e}-1 & 0.000\text{e}+0 \\ -2.800\text{e}-6 & 9.567\text{e}-1 & -4.072\text{e}-5 & 3.193\text{e}-2 & -1.276\text{e}-2 & 0.000\text{e}+0 \\ -2.028\text{e}-7 & 5.278\text{e}-3 & 1.000\text{e}+0 & 3.286\text{e}-2 & -8.494\text{e}-3 & 0.000\text{e}+0 \\ -1.225\text{e}-5 & 3.127\text{e}-1 & 9.198\text{e}-6 & 9.737\text{e}-1 & -4.568\text{e}-1 & 0.000\text{e}+0 \\ 0.000\text{e}+0 & 0.000\text{e}+0 & 0.000\text{e}+0 & 0.000\text{e}+0 & 5.134\text{e}-1 & 0.000\text{e}+0 \\ -4.805\text{e}-5 & -1.147\text{e}+0 & -8.555\text{e}-4 & -8.030\text{e}-3 & 1.098\text{e}-1 & 1.000\text{e}+0 \end{bmatrix}$$

$$B_r = \begin{bmatrix} 0.000\text{e}+0 \\ 0.000\text{e}+0 \\ 0.000\text{e}+0 \\ 0.000\text{e}+0 \\ 0.000\text{e}+0 \\ 3.333\text{e}-2 \end{bmatrix} \qquad B_{u1} = \begin{bmatrix} 3.494\text{e}-2 \\ -3.623\text{e}-3 \\ -1.992\text{e}-3 \\ -1.699\text{e}-1 \\ 4.865\text{e}-1 \\ 3.994\text{e}-2 \end{bmatrix}$$

$$C_m = \begin{bmatrix} -1.500\text{e}-3 & -3.564\text{e}+1 & -2.720\text{e}-2 & 3.340\text{e}-1 & 4.366\text{e}+0 & 1.000\text{e}+1 \end{bmatrix}$$
$$C_{y1} = \begin{bmatrix} -1.500\text{e}-3 & -3.564\text{e}+1 & -2.720\text{e}-2 & 3.340\text{e}-1 & 4.366\text{e}+0 & 0.000\text{e}+0 \end{bmatrix}$$

$$D_{mr} = \begin{bmatrix} 1.000\text{e}+0 \end{bmatrix} \qquad D_{mu} = \begin{bmatrix} 0.000\text{e}+0 \end{bmatrix}$$
$$D_{yr} = \begin{bmatrix} 1.000\text{e}+0 \end{bmatrix} \qquad D_{yu} = \begin{bmatrix} 0.000\text{e}+0 \end{bmatrix}$$

# References

[Bro91]     J. M. Brown II, "Design of an Air-to-Air Missile Autopilot Using Linear Quadratic Gaussian/Loop Transfer Recovery", MS Thesis, AF Inst. of Tech., Wright-Patterson AFB, OH, Dec 1991.

[DDB94]     M.A. Dahleh and I.J. Diaz-Bobillo, *Control of Uncertain Systems: A Linear Programming Approach*, Prentice Hall, Englewood Cliffs, NJ, 1994.

[DGKF89]    Doyle, J.C., Glover, K., Khargonekar, P. and Francis, B., "State-Space Solutions to Standard $H_2$ and $H_\infty$ Control Problems", *IEEE Trans. on Auto. Control*, Vol.34#8, Aug 1989.

[DP87]      M.A. Dahleh and J.B. Pearson, "$\ell_1$ Optimal Feedback Controllers for MIMO Discrete-time Systems", *IEEE Trans. on Auto. Control*, Vol.32#4, Apr 1987, pp.314-327.

[JR94]      D.R. Jacques and D.B. Ridgely, "A Fixed-Order, Mixed-Norm Control Synthesis Method for Discrete Linear Systems", Submitted to the 1995 American Control Conference (ACC), Seattle, WA, Jun 95.

[Luk93]     J.P. Luke, "Flight Controller Design Using Mixed $H_2/H_\infty$ Optimization with a Singular $H_\infty$ Constraint", MS Thesis, AF Inst. of Tech., Wright-Patterson AFB, OH, Dec 1993.

[Mac89]     J.M. Maciejowski, *Multivariable Feedback Design*, Addison-Wesley, Workingham, England, 1989.

[Nel76]     R.B. Nelson, "Simplified Calculation of Eigenvector Derivatives", *AIAA Journal*, Vol.14#9, Sep 1976, pp.1201-1205.

[RB86]      D.B. Ridgely and S.S. Banda, *Introduction to Robust Multivariable Control*, AFWAL-TR-85-3102, Flight Dynamics Lab, Wright-Patterson AFB, OH, Feb 1986.

[Rid92]     D.B. Ridgely, "A Non-Conservative Solution to the General Mixed $H_2/H_\infty$ Optimization Problem", PhD Dissertation, Mass. Inst. of Tech., Cambridge, MA, Feb 1992.

[Vid86]     M. Vidyasagar, "Optimal Rejection of Persistent Bounded Disturbances", *IEEE Trans. on Auto. Control*, Vol.31#6, Jun 1986, pp.527-534.

[Wal94]     D.E. Walker, "$H_2$ Optimal Control with $H_\infty$, $\mu$, and $L_1$ Constraints", PhD Dissertation, AF Inst. of Tech., Wright-Patterson AFB, OH, Jun 1994.

*Vita*

Captain Mark S. Spillman was born on 3 March 1967 in San Pedro, California. He graduated from Mariner High School, Everett, Washington in 1985 and subsequently attended the University of Washington on an AFROTC scholarship. In June 1989, he received a Bachelor of Science in Aeronautical and Astronautical Engineering and was commissioned in the USAF. Capt Spillman began active duty in October 1989 attending Undergraduate Controller Training at Tyndall AFB, Florida. Upon graduation, he was assigned to the 607 Air Control Squadron at Luke AFB, Arizona. He entered the Air Force Institute of Technology's Graduate School of Engineering in May 1994 and, upon graduation, will be assigned to the Flight Dynamics Directorate of Wright Laboratories, Wright-Patterson AFB, Ohio.

Permanent address:   13411 Meridian Place West

Everett, WA 98208

# REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

| 1. AGENCY USE ONLY (Leave blank) | 2. REPORT DATE<br>December 1994 | 3. REPORT TYPE AND DATES COVERED<br>Master's Thesis |
|---|---|---|

**4. TITLE AND SUBTITLE**
APPLICATIONS OF l1 AND MIXED H2/l1 OPTIMIZATION

**5. FUNDING NUMBERS**

**6. AUTHOR(S)**
Mark S. Spillman, Captain, USAF

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Air Force Institute of Technology, WPAFB OH 45433-6583

**8. PERFORMING ORGANIZATION REPORT NUMBER**

AFIT/GAE/ENY/94D-12

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
Dr Marc Jacobs
AFOSR/NM
110 Duncan Ave, Suite B115
Bolling AFB DC 20332-0001

**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**
Approved For Public Release
Distribution Unlimited

**12a. DISTRIBUTION / AVAILABILITY STATEMENT**

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** (Maximum 200 words)

This thesis explores the use of l1 and mixed H2/l1 optimization methods to design flight control systems. l1 optimization is used to handle tracking issues in the design of digital compensators. Control deflection and rate limitations, overshoot and undershoot limitations and steady-state error requirements are discussed. Model-matching techniques which produce acceptable tracking results with lower order controllers are also examined. New numerical methods for continuous H2/L1 and discrete H2/l1 optimization are presented. These methods are used to design an aircraft controller in continuous and discrete time and the results are compared.

**14. SUBJECT TERMS**
Control Theory, Discrete Control Theory, l1 Optimization, L1 Optimization, H2 Optimization, Mixed H2/l1 Optimization, Mixed H2/L1 Optimization

**15. NUMBER OF PAGES**
134

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT<br>UNCLASSIFIED | 18. SECURITY CLASSIFICATION OF THIS PAGE<br>UNCLASSIFIED | 19. SECURITY CLASSIFICATION OF ABSTRACT<br>UNCLASSIFIED | 20. LIMITATION OF ABSTRACT<br>UL |
|---|---|---|---|