

HIFOO 1.5: Structured control of linear systems with non-trivial feedthrough

by

Marc A. Millstone

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Master of Science
Department of Mathematics
Courant Institute of Mathematical Sciences
New York University
September 2006

Michael Overton

© Marc A. Millstone
All Rights Reserved, 2006

- *To my Grandfather, my main inspiration*
- *To Fitzgerald, my main relaxation*

ABSTRACT

Burke et al. [3], introduced the Matlab package HIFOO (H-Infinity Fixed Order Optimization) in 2006 with the main goal of providing a powerful, yet user-friendly tool for computing reduced-order controllers of linear systems. Built upon powerful methods for non-convex and non-smooth optimization, HIFOO attempts to compute controllers which not only stabilize the given plant, but also locally optimize one of several provided objective functions.

In this thesis, we will present two new extensions to HIFOO. The first allows the specified plant to contain a non-trivial feedthrough term, allowing a direct connection between the control input and the system output that is important in many realistic, physical processes. The second extension allows the user to require specific structure in the state-space representation of the controller matrices. Although well-known techniques exist to convert systems with a non-trivial feedthrough to systems with a feedthrough equal to zero, these techniques destroy any structure in the original controller.

Numerical experiments are provided not only for benchmark problems found in the COMPL_eib library, but also for realistic, physical systems such as the control of a flexible beam and structured, static-output-feedback control of an F-16.

ACKNOWLEDGMENTS

This work would have been impossible without the copious amounts of help, patience, and encouragement from both my advisor, Michael Overton, and his collaborator, Didier Henrion.

I am indebted to any number of people, but I definitely wish to thank my roommates Mike Daglian and Mike Damron for all of the midnight conversations, whimsical nonsense, and timely distractions.

To my family, thank you for letting me do what I want to do, supporting my decisions as crazy as they may be. You are always there for me.

Finally, I wish to thank Anne and Fitzgerald for always keeping me grounded, either with a hug, a smile or by a wet lick to the face. I could have never completed this work without your support, tolerance and love.

TABLE OF CONTENTS

Dedication	iii
Abstract	iv
Acknowledgments	v
List of Notations	viii
Introduction	1
1 Background: Linear Control Theory	3
1.1 Linear Systems	3
1.1.1 State-space Description	4
1.1.2 Transfer Function and Frequency Response	5
1.2 Feedback Control	6
1.2.1 The Linear Fractional Transformation	6
1.2.2 Internal Stability	7
1.3 \mathcal{H}_∞ Control	8
1.4 Reduced-order control	10
2 Problem statement and HIFOO Extensions	12
2.1 Problem Statement	12
2.2 Non-trivial Feedthrough	12
2.3 User-specified controller structure	14
3 HIFOO: Implementation and Usage	15
3.1 Non-trivial Feedthrough	15
3.2 Controller Structure	15

TABLE OF CONTENTS

4 Numerical Experiments with HIFOO	18
4.1 Complex examples	18
4.2 A Flexible Beam	19
4.3 Examples with Structure	21
4.4 Structured SOF control of an F-16 aircraft	26
Future Work	32
A Mathematics	33
A.1 Matrix Calculus	33
A.2 The Space \mathcal{H}_∞	35
B Closed-Loop System Calculations	36
Bibliography	39

LIST OF NOTATIONS

$\ \cdot\ $	\mathcal{H}_∞ norm
\mathcal{H}_∞	Hardy Space
$\mathcal{F}(G, K)$	The (lower) Linear Fractional Transformation
$\alpha(M)$	Spectral Abscissa of M
$\bar{\sigma}(M)$	Maximum singular value of M
$\left[\begin{array}{c c} A & B \\ \hline C & D \end{array} \right]$	State-space realization of system
$\left[\begin{array}{c c} A & B \\ \hline C & D \end{array} \right]$	Transfer function $D + C(sI - A)^{-1}B$

INTRODUCTION

Originally formulated in the 1980's as a framework for handling uncertainty or noise in a given plant model, \mathcal{H}_∞ control quickly provided powerful techniques in the area of controller-synthesis (an overview, with detailed references is provided in Zhou et al. [12]). These techniques quickly led to practical, yet powerful tools in controller design, however the domain in which they could be applied was limited. Specifically, most of these tools were focused on computing full-order controllers, that is ones whose order is the dimension of the plant itself. Recently, there has been a resurgence in new computational tools for finding controllers that have a lower order than that of the plant. In many physically relevant systems, it is often the case that the plant's order is too large to allow a full-order controller to be computationally feasible. Furthermore, even when controlling plants of low order, a reduced-order controller may be desirable when the available memory and computational power is limited, such as in embedded controllers. At the other extreme from full-order control is static-output feedback (SOF), where a controller is sought with order zero. SOF control is desirable, because the control law will be of a specific, simple form, as opposed to a complex system of ordinary-differential equations. Although no general mathematical techniques guarantee a solution to the SOF control problem, computational tools may still succeed in computing a controller with desirable characteristics.

This work provides extensions to the Matlab package HIFOO for \mathcal{H}_∞ Fixed-Order Optimization, created by Burke et al. [3]. Built upon the Matlab package HANSO (Hybrid Algorithm for Non-Smooth Optimization), a powerful tool for non-smooth, non-convex optimization, HIFOO follows a local optimization approach to low-order controller synthesis, utilizing a combination of random and user-provided initial controller matrices. The present work involves two seemingly disparate, but related extensions to HIFOO. The first extension allows a direct link between the plant's controlled input and its measurable output. Although well-known techniques exist which make this extension unnecessary, these methods destroy any specified structure in the controller, which may be important from an engineering standpoint. The second extension allows a simple means for the user to specify this controller structure. As

input, the user may now select which elements of the controller's system matrices to optimize over, fixing the other elements to zero. Together, these extensions provide a user-friendly tool for structured, low-order, control of linear-systems with a nontrivial feedthrough term.

BACKGROUND: LINEAR CONTROL THEORY

This chapter will provide a brief review of topics from Linear Systems and \mathcal{H}_∞ Control Theory. For a more detailed introduction to Linear Systems, Robust and Optimal Control, and \mathcal{H}_∞ control see Chen [4], Dullerud and Paganini [5], Zhou et al. [12], and Francis [6] respectively.

1.1 Linear Systems

When analyzing physical processes, it is often important to mathematically model both the internal dynamics of the process as well as its interaction with the external world. This mathematical model is called a *dynamical system* and we define its *state* at time t_0 as the internal information $x(t_0)$ that together with the external input $u(t)$ determines the model's output $y(t)$ uniquely, for $t \geq t_0$. This relationship is denoted via the compact notation $[(x(t_0), u(t)) \rightarrow y(t)]$. Let $[(x_1(t_0), u_1(t)) \rightarrow y_1(t)]$ and $[(x_2(t_0), u_2(t)) \rightarrow y_2(t)]$ correspond to two state-input-output pairs. A given system is *linear* if, for all $t \geq t_0$, it satisfies the law of superposition, ie.

$$[(\alpha x_1(t_0) + \beta x_2(t_0), \alpha u_1(t) + \beta u_2(t)) \rightarrow \alpha y_1(t) + \beta y_2(t)]. \quad (1.1)$$

Furthermore we will be interested in systems which are *time invariant*. Intuitively, this means that regardless of the starting time, if the initial state and control input are the same, the output will also be the same. Formally, let $[(x(t_0), u(t)) \rightarrow y(t)]$ for $t \geq t_0$. The system is time-invariant, if for any $T \in \mathbb{R}$,

$$[(x(t_0 + T), u(t - T)) \rightarrow y(t - T)] \quad \text{for all } t \geq t_0 + T. \quad (1.2)$$

If a system satisfies both Property 1.1 and Property 1.2, we say the system is *Linear Time-Invariant (LTI)*. Throughout this work, we will assume that all systems are linear time-invariant and furthermore that the state of the system has finite dimension.

1.1.1 State-space Description

Given the linearity condition in Equation 1.1, every LTI-system emits a *state-space realization* that is described by the set of differential equations:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t), \quad x(0) = 0 \\ y(t) &= Cx(t) + Du(t),\end{aligned}\tag{1.3}$$

where $x \in \mathbb{R}^n$ is the *state vector*, $u(t) \in \mathbb{R}^m$ is the system *input* or *control vector*, and $y(t) \in \mathbb{R}^p$ is the system output vector. Observe that the above plant is a system of first-order differential equations with a specified initial condition and therefore, must have a unique solution. A common, more compact, notation is frequently used by collecting x and u into a single vector and writing (1.3) as:

$$\begin{bmatrix} \dot{x} \\ y \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}.\tag{1.4}$$

In many situations, the controller may not be able to control a subset of the system input, u , nor measure a subset of the system output, y . We call these additional, non-controllable and non-measurable terms the *exogenous* input vector, $w(t)$, and *exogenous* output vector, $z(t)$, respectively. Often the exogenous input will represent system noise or information and the exogenous output a known quantity that we wish to control. We expand System 1.3 to include such terms as follows:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + [B_1 \ B_2] \begin{bmatrix} w(t) \\ u(t) \end{bmatrix} \\ \begin{bmatrix} z(t) \\ y(t) \end{bmatrix} &= \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} x(t) + \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \end{bmatrix}.\end{aligned}\tag{1.5}$$

Following the above compact notation, we again collect vectors x , w , and u into a single vector and write this system as

$$\begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix}.\tag{1.6}$$

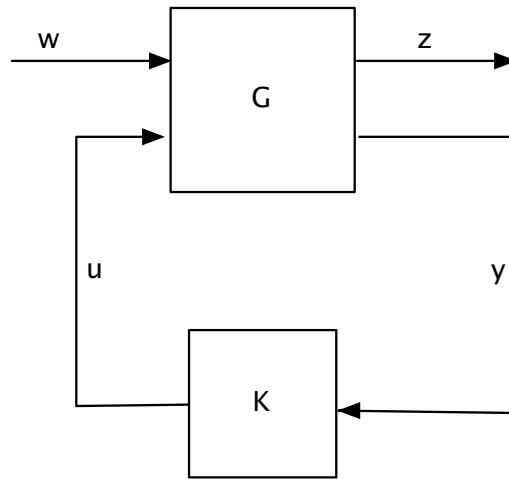


Figure 1.1: Closed Loop system.

1.1.2 Transfer Function and Frequency Response

In addition to the state-space representation given in the previous section, linear systems can be described via a *transfer function* that provides a relationship between the control input, u , and system output, y , in the frequency domain. As we will see, the transfer function representation will provide insight into various properties of the system. By applying the Laplace transform to System 1.3 and assuming that $x(0) = 0$, we obtain

$$\begin{aligned} sX(s) &= AX(s) + BU(s) \\ Y(s) &= CX(s) + DU(s) \end{aligned} \tag{1.7}$$

where the capitalized terms $X(s)$, $Y(s)$ and $U(s)$ denote the Laplace transforms of the system inputs $x(t)$, $y(t)$, and $u(t)$ respectively. Substituting the first equation into the second, we can calculate the system transfer function

$$Y(S) = \left(D + C(sI - A)^{-1}B \right) U(s). \tag{1.8}$$

Therefore, the state-space realization of the linear system given by Equation 1.3 provides a state-space realization of its transfer function via the equation

$$G(s) = D + C(sI - A)^{-1}B. \tag{1.9}$$

We will often refer to this transfer function via the matrix notation

$$G(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right]. \quad (1.10)$$

1.2 Feedback Control

Once the plant is modeled by Equation 1.5, the primary goal of *feedback control* is to design a controller that from the measurable system output y computes a new control vector u to then pass back into the plant. In this work, our goal will be to design a controller which, at the very least, *stabilizes* the given system. This process is depicted in Figure 1.1 where G denotes the plant and K the controller. There we connect the *open-loop* model of the plant with the *open-loop* model of the controller to form the *closed-loop* model of the entire system.

As in the above systems, the controller can be modeled by a set of state-space equations of the form:

$$\begin{bmatrix} \dot{d} \\ u \end{bmatrix} = \begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} \begin{bmatrix} d \\ y \end{bmatrix} \quad (1.11)$$

where $d \in \mathbb{R}^{\hat{n}}$ denotes the controller's state and \hat{n} denotes the *order* of the controller. As with the plant above, the controller has its own transfer function representation,

$$K(s) = \hat{D} + \hat{C}(sI - \hat{A})^{-1}\hat{B}. \quad (1.12)$$

1.2.1 The Linear Fractional Transformation

The *Linear Fractional Transformation (LFT)* provides a mathematical means of modeling the connection depicted in Figure 1.1. Let $G(s)$ and $K(s)$ denote the transfer functions of the plant and controller respectively. Furthermore, we partition the transfer function of the plant as follows:

$$G(s) = \left[\begin{array}{c|c} G_{11} & G_{12} \\ \hline G_{21} & G_{22} \end{array} \right] \quad (1.13)$$

where the dimensions of G_{11} , G_{12} , G_{21} , and G_{22} correspond the dimensions of the system matrices D_{11} , D_{12} , D_{21} , and D_{22} respectively. We can then model the closed-loop transfer

function from the exogenous input w to the exogenous output z via the LFT

$$\mathcal{F}(G, K) = G_{11} + G_{12}K(I - G_{22}K)^{-1}G_{21}. \quad (1.14)$$

One such state-space realization of this LFT is given by the closed-loop equations calculated in Appendix B.

In this work, the only theoretical use of this LFT will be in Section 1.3, where we use Equation 1.14 to show the equivalence of two different closed-loop systems.

1.2.2 Internal Stability

Once the controller is connected to the plant as in Figure 1.1, it is important to ensure that the new closed-loop system behaves properly regardless of the input provided to the system. One such desirable property is the *internal stability* of the closed-loop system.

We call the connection of a plant described by Equation 1.5 with a controller described by Equation 1.11, *well posed* if the matrix $(I - \hat{D}D_{22})$ is non-singular. Observe that if the plant has a trivial feedthrough term, ie. $D_{22} = 0$, then the connection is well-posed *a priori*. The connected system is *internally stable* if, in addition to being well posed, it is such that whenever $w = 0$,

$$\lim_{t \rightarrow \infty} (x(t), d(t)) = (0, 0) \quad (1.15)$$

for every pair of initial conditions $(x(t_0), d(t_0))$. Although precise, this is not a very computationally useful definition for testing whether a system is internally stable. Another equivalent definition of stability is given after the following definitions.

The *spectral abscissa* of a square matrix denotes the maximum real part of its eigenvalues. Symbolically, let $\Lambda(M)$ denote the set of eigenvalues for matrix M . The spectral abscissa, $\alpha(M)$, is given by the formula

$$\alpha(M) = \max_{\lambda_i \in \Lambda(M)} \operatorname{Re}(\lambda_i). \quad (1.16)$$

A matrix M is *stable* if the spectral abscissa satisfies $\alpha(M) < 0$, namely M is stable if its eigenvalues are located in the open left half-plane.

We now give an equivalent definition of internal stability, modified slightly from Dullerud and Paganini [5].

Proposition 1.1. *The closed-loop system described by Figure 1.1 is internally stable if and only if $(I - \hat{D}D_{22})$ is non-singular and the matrix*

$$A_k = \begin{bmatrix} A + B_2 F^{-1} \hat{D} C_2 & B_2 F^{-1} \hat{C} \\ \hat{B} C_2 + \hat{B} D_{22} F^{-1} \hat{D} C_2 & \hat{A} + \hat{B} D_{22} F^{-1} \hat{C} \end{bmatrix} \quad (1.17)$$

is stable, where $F = (I - \hat{D}D_{22})$.

A proof of this proposition is provided in the above reference. Observe that the matrix A_k is the partition of the closed-loop matrix calculated in Appendix B concerning the state variables x and d . The equivalence then follows from standard results of ordinary differential equations and the matrix exponential. See Dullerud and Paganini [5, Chapters 2 and 5] for more details.

1.3 \mathcal{H}_∞ Control

Again, given a plant, G , and a controller, K , we consider the system described by the block diagram depicted in Figure 1.1. Let (A_k, B_k, C_k, D_k) denote the state-space realization of this closed-loop system, derived in Appendix B. The transfer function between the input signal w and the output signal z is given by

$$T(s) = D_k + C_k(sI - A_k)^{-1}B_k. \quad (1.18)$$

We now define the \mathcal{H}_∞ norm of this system by the formula

$$\|T\| = \sup_{\omega \in \mathbb{R}} \bar{\sigma}(D_k + C_k(i\omega I - A_k)^{-1}B_k), \quad (1.19)$$

where $\bar{\sigma}(\cdot)$ denotes the maximum singular value. This quantity is shown graphically, via a *Bode magnitude plot*, which displays the logarithm of the singular values of a given system as a function of the logarithm of the frequency. In Matlab, the function `sigma(sys)` generates the Bode magnitude plot of the system specified by the `ss` object, `sys`. Note that Matlab actually plots the logarithm of the singular values scaled by a constant factor of 20.

The \mathcal{H}_∞ Control Problem can now be defined as: given a plant G , determine a controller K such that

1 BACKGROUND: LINEAR CONTROL THEORY

- The closed-loop system is internally stable, as in Proposition 1.1
- The \mathcal{H}_∞ -norm of the transfer function, given by Equation 1.19, is minimized.

Clearly, internal stability is a most basic requirement for the system to be physically feasible, since otherwise small amounts of noise or errors could lead to an unbounded system output of the closed-loop system. The second condition seeks to find a controller which for all possible input signals, w , minimizes the maximum energy of the output signal, z . Graphically, the goal will be to compute a controller which minimizes the peaks shown in the Bode magnitude plot of the given plant.

Often in \mathcal{H}_∞ control it is assumed that the given plant has a trivial feedthrough term, ie. $D_{22} = 0$, which greatly simplifies the state-equations given in Equation 1.17. We will now see how to remove this assumption via the following proposition, stated in Zhou et al. [12]. As no proof of this statement could be found, we present our own.

Proposition 1.2. *Let $G(s)$ be a given plant and let $\tilde{G}(s)$ denote an identical plant but with $D_{22} = 0$. Assume that \tilde{K} is a stabilizing controller for \tilde{G} and that $(I + D_{22}\tilde{K})$ is invertible. Then*

$$\mathcal{F}(G, K) = \mathcal{F}(\tilde{G}, \tilde{K}) \quad (1.20)$$

where $K = \tilde{K}(I + D_{22}\tilde{K})^{-1}$

Proof. Let the original plant $G(s)$ be denoted by the following equivalent representations.

$$\begin{aligned} G(s) &= \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array} \right] \\ &= \left[\begin{array}{c|cc} D_{11} & D_{12} \\ \hline D_{21} & D_{22} \end{array} \right] + \left[\begin{array}{c} C_1 \\ C_2 \end{array} \right] (sI - A)^{-1} [B_1 \ B_2] \\ &= \left[\begin{array}{c|cc} G_{11}(s) & G_{12}(s) \\ \hline G_{21}(s) & G_{22}(s) \end{array} \right]. \end{aligned}$$

From the second equation we observe that the simplified system $\tilde{G}(s)$ can be written as

$$\begin{aligned}\tilde{G}(s) &= G(s) - \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & D_{22} \end{array} \right] \\ &= \left[\begin{array}{c|c} G_{11}(s) & G_{12}(s) \\ \hline G_{21}(s) & G_{22}(s) - D_{22} \end{array} \right].\end{aligned}$$

We now expand $\mathcal{F}(G, K)$ via expression 1.14 to obtain

$$\mathcal{F}(G, K) = G_{11} + G_{12}K(I - G_{22}K)^{-1}G_{21}.$$

We achieve the desired result by substituting $K = \tilde{K}(I + D_{22}\tilde{K})^{-1}$ and performing a sequence of algebraic manipulations as follows:

$$\begin{aligned}\mathcal{F}(G, K) &= G_{11} + G_{12}\tilde{K}(I + D_{22}\tilde{K})^{-1}(I - G_{22}\tilde{K}(I + D_{22}\tilde{K})^{-1})^{-1}G_{21} \\ &= G_{11} + G_{12}\tilde{K} \left[(I - G_{22}\tilde{K}(I + D_{22}\tilde{K})^{-1})(I + D_{22}\tilde{K}) \right]^{-1}G_{21} \\ &= G_{11} + G_{12}\tilde{K} [(I + D_{22}\tilde{K}) - G_{22}\tilde{K}]^{-1}G_{21} \\ &= G_{11} + G_{12}\tilde{K} [I - (G_{22} - D_{22})\tilde{K}]^{-1}G_{21} \\ &= \mathcal{F}\left(G - \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & D_{22} \end{array} \right], \tilde{K}\right) \\ &= \mathcal{F}(\tilde{G}, \tilde{K})\end{aligned}$$

□

Therefore, since the two LFT's in Equation 1.20 are identical, for any stabilizing controller \tilde{K} of the simplified system \tilde{G} , we can obtain a stabilizing controller for the original system via the bijection $\tilde{K}(I + D_{22}\tilde{K})^{-1}$.

1.4 Reduced-order control

When the order of the plant is large, often it is desired to design a controller with smaller order. The extreme case of this is a *static-output feedback (SOF)* controller, where $\hat{n} = 0$. In this case, we see that System 1.11 has no internal dynamics, taking the form of the simple

1 BACKGROUND: LINEAR CONTROL THEORY

control law, $u = \hat{D}y$. The general problem is to compute a *reduced-order controller* with $\hat{n} < n$ which maintains similar \mathcal{H}_∞ performance and behavior to a full-order controller of the given plant.

2

PROBLEM STATEMENT AND HIFOO EXTENSIONS

This section presents the general problem of \mathcal{H}_∞ control as well as two extensions to the work of Burke et al. [3].

2.1 Problem Statement

We again consider the following state-space description of an open-loop, linear-system, G :

$$\begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_1 & B_2 \\ C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x \\ w \\ u \end{bmatrix} \quad (2.1)$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the controller input, $y \in \mathbb{R}^p$ is the measurable output. The vectors w and z correspond to the exogenous input vector and output vector respectively.

Our goal is to compute a controller K of the form:

$$\begin{bmatrix} \dot{d} \\ u \end{bmatrix} = \begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} \begin{bmatrix} d \\ y \end{bmatrix} \quad (2.2)$$

which stabilizes the closed-loop system shown in Figure 2.1.

2.2 Non-trivial Feedthrough

The first extension to Burke et al. [3] allows the given plant to have a non-trivial feedthrough connection, ie. D_{22} need not be zero. This allows a direct connection between the control input, u and the plants controllable output, y , leading to more complicated closed-loop equations when we connect systems G and K as in Figure 2.1.

Eliminating the internal connection u and y through algebraic manipulations (see Ap-

2 PROBLEM STATEMENT AND HIFOO EXTENSIONS

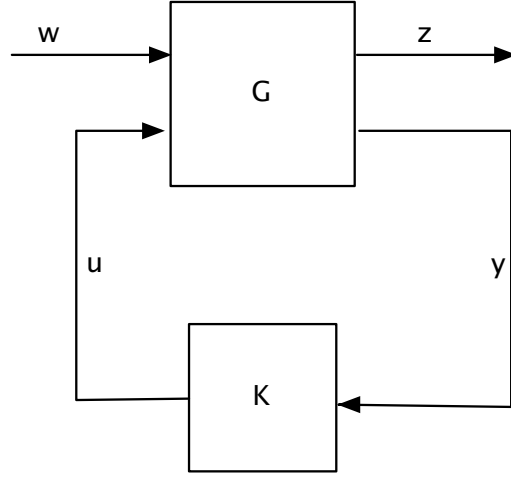


Figure 2.1: Closed Loop system.

pendix B), we obtain the closed-loop system

$$\begin{bmatrix} \dot{x} \\ \dot{d} \\ z \end{bmatrix} = \begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix} \begin{bmatrix} x \\ d \\ w \end{bmatrix} \quad (2.3)$$

where

$$\begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix} = \begin{bmatrix} A + B_2 F^{-1} \hat{D} C_2 & B_2 F^{-1} \hat{C} & B_1 + B_2 F^{-1} \hat{D} D_{21} \\ \hat{B} C_2 + \hat{B} D_{22} F^{-1} \hat{D} C_2 & \hat{A} + \hat{B} D_{22} F^{-1} \hat{C} & \hat{B} D_{21} + \hat{B} D_{22} F^{-1} \hat{D} D_{21} \\ C_1 + D_{12} F^{-1} \hat{D} C_2 & D_{12} F^{-1} \hat{C} & D_{11} + D_{12} F^{-1} \hat{D} D_{21} \end{bmatrix} \quad (2.4)$$

and $F = (I - \hat{D} D_{22})$. Observe that since the feedthrough term, D_{22} , is *not* assumed to be zero, this matrix is nonlinear in the controller variable, \hat{D} , also complicating required gradient calculations (given in Appendix B).

As mentioned in the previous chapter, we can represent the closed-loop system (A_k, B_k, C_k, D_k) as a transfer function between the exogenous input vector w and system output z which gives the input-output behavior of the closed loop, namely

$$\begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix} = D_k + C_k (sI - A_k)^{-1} B_k. \quad (2.5)$$

The originally specified control problems addressed by HIFOO in Burke et al. [3], which include

- Finding a reduced-order controller which stabilizes a given plant
- Finding a reduced-order controller which in addition to the above, locally minimizes the \mathcal{H}_∞ norm of the closed-loop system,

remain unchanged.

It should be noted that Proposition 1.2 provides an implicit means to construct a stabilizing controller for any general plant with nontrivial feedthrough. Simply apply HIFOO to the same plant, but with D_{22} set to zero. Then apply the given bijection to the calculated controller to obtain a stabilizing controller of the original plant. However, given the next extension, that of user-specified controller-structure, the ability to handle the feedthrough term directly is essential, as the aforementioned bijection destroys any structure in the calculated controller.

2.3 User-specified controller structure

The second extension to HIFOO involves the ability to specify structure to the controller state matrices \hat{A} , \hat{B} , \hat{C} and \hat{D} . The user may input which entries of each controller matrix to optimize over, fixing the other entries to zero. Desirable structure examples include fixing the \hat{D} matrix to all zeros, resulting in a strictly proper controller transfer function; specifying that each matrix is diagonal, resulting in a diagonal controller transfer function; or specifying that each controller matrix is block diagonal, so that the corresponding transfer function is itself block diagonal. The latter is useful in designing decoupled controllers for large-scale systems, where instead of designing one, large controller, the set of input and output variables is partitioned and several reduced-order controllers are found.

3

HIFOO: IMPLEMENTATION AND USAGE

This chapter provides a technical elaboration on the extensions provided to HIFOO as well as some basic examples of the new functionality. See the next chapter for more detailed examples.

3.1 Non-trivial Feedthrough

In Appendix B, we derive the closed-loop equations and gradients for a plant with $D_{22} \neq 0$. Observe that these equations are non-linear in the controller variable \hat{D} , but simplify when $D_{22} = 0$ exactly. Therefore, HIFOO tests whether $D_{22} = 0$ and if so, utilizes the original, linear equations and gradients.

3.2 Controller Structure

The ability to specify controller structure is implemented through Matlab's `logical` functionality. In Matlab, a `logical` matrix is a matrix of boolean values that is used as an index into another matrix of elements. The following provides an example of the usage of Matlab's `logical` indexing ability:

```
>> MATRIX = [1 2; 3 4];  
>> INDEX = logical([0 1; 1 0])  
>> OFFDIAG = MATRIX(INDEX)
```

```
OFFDIAG =
```

```
    3
```

```
    2
```

Observe that the elements are returned in a column-wise fashion and that the dimensions of the logical index must match the dimensions of the matrix of elements.

When calling HIFOO, the user, in addition to providing the order of the desired controller, may now specify a structure array of matrices corresponding to the controller variables \hat{A} , \hat{B} , \hat{C} and \hat{D} . Specifically, the user may pass a structure in the field options.structure, with subsequent fields Ahat, Bhat, Chat and Dhat that specifies which elements of the matrices \hat{A} , \hat{B} , \hat{C} , and \hat{D} HIFOO is free to optimize over and which are fixed to 0. The user may specify these matrices as either binary or logical matrices; however, HIFOO converts binary matrices to logical variables for internal use. For convenience, if a given matrix is not specified, HIFOO assumes that all entries of the matrix are free to be optimized over and sets the corresponding structure matrix to be a logical matrix consisting of all ones.

In addition to modifying the input format of HIFOO, it was necessary to update the required gradient computations. This new gradient information is implied in example A.2, simplified and rewritten here for convenience:

$$f(A + \Delta, B, C, D) = f(A, B, C, D) + \langle \nabla_A f, \Delta \rangle + O(\Delta^2).$$

If A has a specific structure, then an identical structure is imposed upon the perturbation Δ . Letting the logical matrix J denote the structure of A , the structured gradient becomes $\nabla_A f(J)$, a vector corresponding to the free elements of A .

The next example demonstrates a standard HIFOO session utilizing this new functionality. The goal is to find an order one controller for plant 'AC1' from the Compl_eib library, such that the controller's transfer function is diagonal. A sufficient condition for the transfer function to be diagonal is that the system realization of the controller has the following structure:

$$\hat{A} = (a) \quad \hat{B} = \begin{pmatrix} b_1 & 0 & 0 \end{pmatrix}$$

$$\hat{C} = \begin{pmatrix} c_1 \\ 0 \\ 0 \end{pmatrix} \quad \hat{D} = \begin{pmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{pmatrix}.$$

A typical session would then proceed as follows:

```
>> STRUCTURE.Ahat = logical([1]);
>> STRUCTURE.Bhat = logical([1 0 0]);
>> STRUCTURE.Chat = logical([1;0;0]);
```


3 HIFOO: IMPLEMENTATION AND USAGE

```
>> STRUCTURE.Dhat = logical(eye(3,3));  
>> options.structure = STRUCTURE;  
>> K = hifoo('AC1',1,options);
```

Observe that Ahat field as well as the logical conversions are not necessary, but are only included as an example of use. To verify that the desired structure is indeed obtained, we convert K to an ss object and calculate its transfer function as follows:

```
>> Kss = ss(K.a, K.b, K.c, K.d);  
>> tf(Kss)
```

Transfer function from input 1 to output...

```
      2.784 s + 3.213e-12  
#1:  -----  
           s + 3.775
```

```
#2:  0
```

```
#3:  0
```

Transfer function from input 2 to output...

```
#1:  0
```

```
#2: -0.1694
```

```
#3:  0
```

Transfer function from input 3 to output...

```
#1:  0
```

```
#2:  0
```

```
#3:  8.178
```

NUMERICAL EXPERIMENTS WITH HIFOO

This chapter provides numerical experiments of the new features of HIFOO along with comparisons, when applicable, to Matlab’s Robust Control Toolbox function `hinfsv`, a powerful tool for finding full-order controllers of a given plant.

4.1 `Compleib` examples

We first perform tests based on plants found in the `Compleib` library, described in Leibfritz [9]. However, since all the models in this library provide plants with a trivial feedthrough term, we set the D_{22} plant variable to a random matrix of the appropriate dimension. Although this modified plant will no longer correspond to the original physical system, it will allow for a means of comparison between the controllers generated by `hinfsv` and HIFOO.

We first apply both tools to the problem labeled ‘AC4’ in the `Compleib` library. Table 4.1 depicts the \mathcal{H}_∞ norms of the generated controllers for different orders by HIFOO (`hinfsv` only computes a full-order controller, which is the value listed). We immediately observe that the order-2 controller (marked by #) generated by HIFOO almost equals the \mathcal{H}_∞ performance of the full-order controller found by `hinfsv`.

order	HIFOO
0	0.935467
1	0.557814
2	0.557332 (#)
3	0.557332
4	0.557318
<code>hinfsv</code>	0.557371

Table 4.1: A comparison between HIFOO and `hinfsv` on `Compleib` ‘AC4’.

4.2 A Flexible Beam

Moheimani et al. [10] provides a realistic model of a flexible beam controlled by piezoelectric actuators, which naturally gives rise to a nonzero feedthrough term. We first look at the Bode magnitude plot (described in Section 1.3) of the plant which gives the oscillatory behavior of the singular values of the open-loop system at varying frequencies. The goal is to find a controller that mitigates these peaks and oscillations.

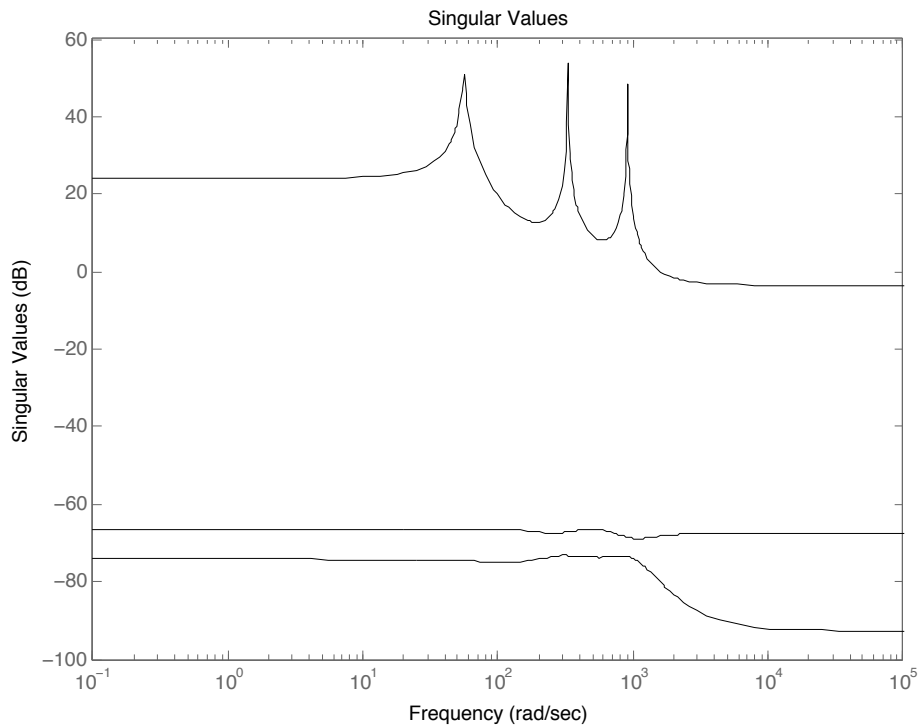


Figure 4.1: Bode magnitude plot of open-loop model of beam

4.2 A FLEXIBLE BEAM

We now apply HIFOO to compute a full-order controller for this system (the commands to load the flexible-beam model into Matlab as the ss object FLEXIBLE_BEAM have been suppressed).

```
>> [K] = hifoo(FLEXIBLE_BEAM,6);
```

We now overlay the Bode magnitude plot of the closed-loop system found by HIFOO (given by `tf(FLEXIBLE_BEAM,K)` in Matlab) on Figure 4.1.

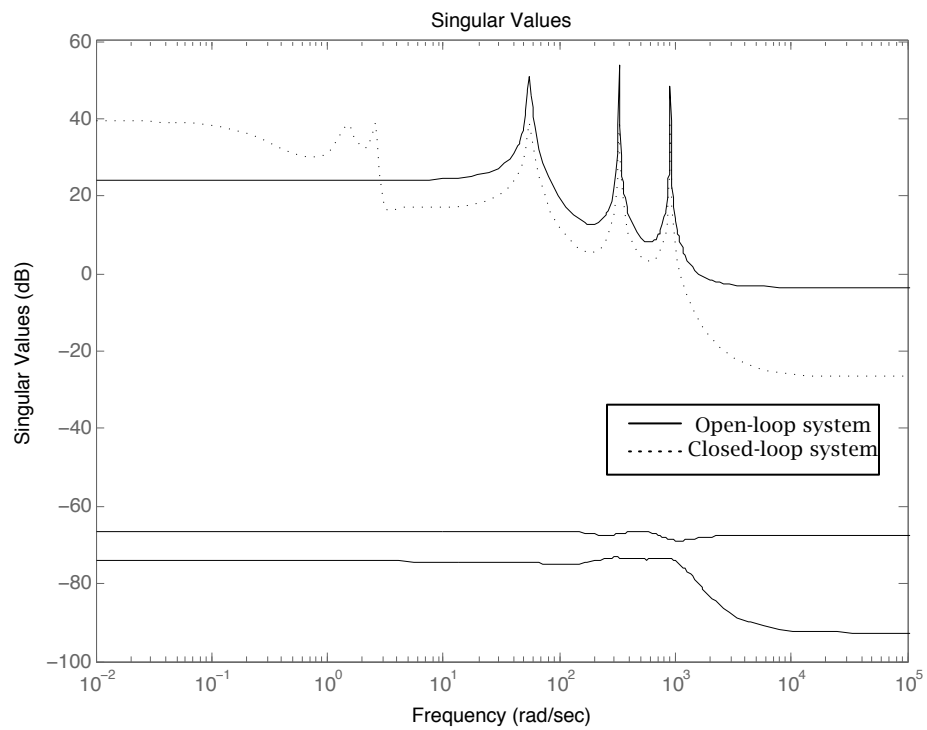


Figure 4.2: Bode Magnitude plot of both open and closed loop system computed by HIFOO

4 NUMERICAL EXPERIMENTS WITH HIFOO

We see that the controller found by HIFOO introduces a few small spikes at low frequencies, but in general does decrease the peaks of the system. For this particular problem, the controller computed by `hifsyn` performs better than that found by HIFOO, as shown in Figure 4.3. HIFOO quickly found a stabilizing controller for the flexible beam, however, due to the local optimization approach inherent in its design, was unable to compute a comparable controller. If, however, the controller from `hifsyn` was passed to HIFOO as an initial point, HIFOO was able to slightly improve upon that controller's \mathcal{H}_∞ performance.

4.3 Examples with Structure

This section provides examples of HIFOO utilizing the new controller structure functionality. As in the first section, we will give examples using plants from the `Compleib` library. A transfer function is *strictly proper* if the limit of each entry approaches zero as the frequency goes to infinity. The first example will utilize HIFOO to find a controller with a strictly proper transfer function for `Compleib` problem 'AC4'. A necessary and sufficient condition for this structure is that the controller's system realization satisfies $\hat{D} = 0$. We specify this in the following Matlab session (remembering that the terms not specified in `options.structure` are assumed to consist entirely of ones):

```
>> STRUCTURE.Dhat = zeros(1,2);
>> options.structure = STRUCTURE;
>> [Ks, fs] = hifoo('AC4',4, options)
Ks =
    a: [4x4 double]
    b: [4x2 double]
    c: [-2.34884535618074  0.71459710888066  1.74937103960848 -1.64849868570093]
    d: [0 0]

fs =
    0.56574372028149
```

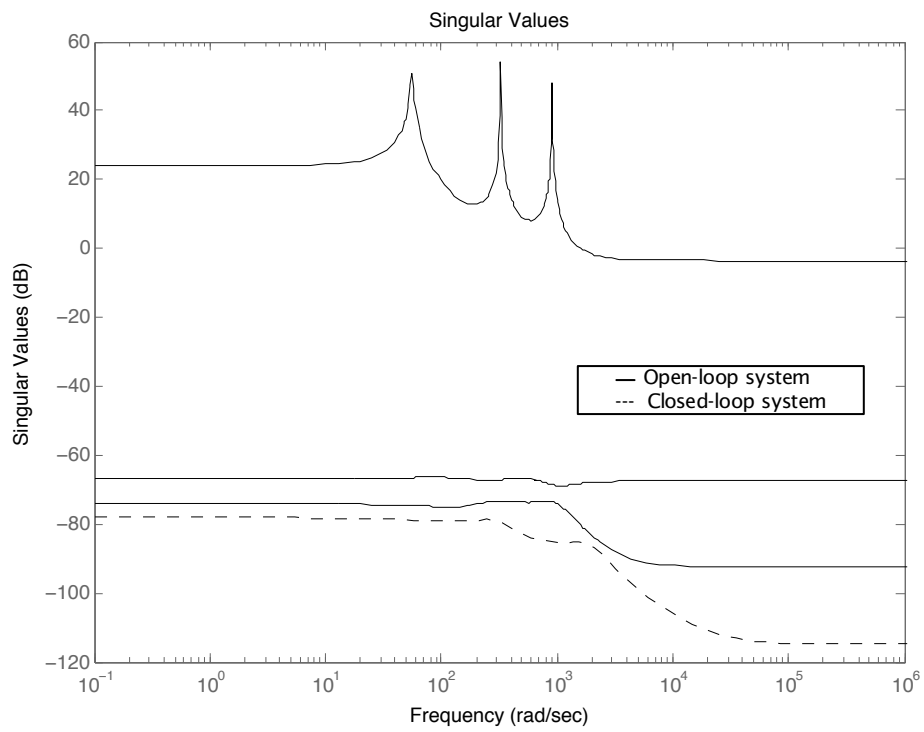


Figure 4.3: Bode Magnitude plot of both open and closed loop system computed by hinfsv

4 NUMERICAL EXPERIMENTS WITH HIFOO

Not only does HIFOO return the computed controller, but the user may also request the \mathcal{H}_∞ norm of the closed-loop system as we did above in variable `fs`. We verify that this is the desired structure by converting the controller returned by HIFOO to a `ss` object and calculating the controller's transfer function. Observe that the degree of the numerator is in fact less than the degree of the denominator, so the limit of each entry does tend to zero at infinity.

```
>> Kss = ss(Ks.a, Ks.b, Ks.c, Ks.d);
>> tf(Kss)
Transfer function from input 1 to output:
  -35.19 s^3 - 2401 s^2 - 1.506e05 s - 1.383e05
-----
s^4 + 141.7 s^3 + 5888 s^2 + 9.354e04 s + 9.197e04

Transfer function from input 2 to output:
  -32.2 s^3 - 1280 s^2 - 2.406e04 s - 8294
-----
s^4 + 141.7 s^3 + 5888 s^2 + 9.354e04 s + 9.197e04
```

As expected, the structured controller achieves a slightly worse \mathcal{H}_∞ norm (`fs` value) than the controllers found in the previous section, since adding structure effectively restricts the space of stabilizing controllers.

Another often desired structure requires that the controller's transfer function be diagonal or block diagonal. We will now apply HIFOO to the `Compl_eib` plant, 'HE2', a fourth order system describing the vertical motion of a helicopter. For this plant, a sufficient condition to guarantee that the controller has a diagonal transfer function is that the controller matrices

have the following structure:

$$\hat{A} = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & a_{43} & a_{44} \end{pmatrix} \quad \hat{B} = \begin{pmatrix} b_{11} & 0 \\ b_{21} & 0 \\ 0 & b_{32} \\ 0 & b_{42} \end{pmatrix}$$

$$\hat{C} = \begin{pmatrix} c_{11} & c_{12} & 0 & 0 \\ 0 & 0 & c_{23} & c_{24} \end{pmatrix} \quad \hat{D} = \begin{pmatrix} d_{11} & 0 \\ 0 & d_{22} \end{pmatrix}.$$

We first utilize HIFOO to calculate a full-order controller for comparison purposes:

```
>> [K,f] = hifoo('HE2',4)
```

```
K =
```

```
  a: [4x4 double]
```

```
  b: [4x2 double]
```

```
  c: [2x4 double]
```

```
  d: [2x2 double]
```

```
f =
```

```
  2.45203584204612
```

Note that for the same problem, `hinfsv` returns a comparable controller with an \mathcal{H}_∞ norm of 2.0122. Now we supply the above structure to HIFOO to find a controller with a diagonal transfer function.

```
>> STRUCTURE.Ahat = [ones(2,2) zeros(2,2); zeros(2,2) ones(2,2)];
```

```
>> STRUCTURE.Bhat = [ones(2,1) zeros(2,1); zeros(2,1) ones(2,1)];
```

```
>> STRUCTURE.Chat = [1 0; 1 0; 0 1; 0 1]';
```

```
>> STRUCTURE.Dhat = eye(2,2);
```

```
>> options.structure = STRUCTURE;
```

```
>> [Ks, fs] = hifoo('HE2', 4, options)
```

```
Ks =
```

```
  a: [4x4 double]
```

```
  b: [4x2 double]
```


4 NUMERICAL EXPERIMENTS WITH HIFOO

```
c: [2x4 double]
d: [2x2 double]t
fs =
4.20598687088419
```

As expected, we observe a loss in \mathcal{H}_∞ performance. Again, we will verify that the calculated controller does in fact have a diagonal transfer function with the following Matlab commands:

```
>> Kss = ss(Ks.a, Ks.b, Ks.c, Ks.d);
>> tf(Kss)
Transfer function from input 1 to output...
      18.31 s^2 + 39.81 s + 13.54
#1:  -----
      s^2 - 7.832 s + 217.4

#2:  0
```

Transfer function from input 2 to output...

#1: 0

$$5.177 s^2 + 64.63 s + 2.473$$

#2: -----

$$s^2 + 25.5 s + 189.4$$

4.4 Structured SOF control of an F-16 aircraft

Bates and Postlethwaite [1, §4.3.6] provide an example of designing a structured, static-output feedback, wing-leveller controller for an F-16 aircraft, first described in Stevens and Lewis [11]. The state-space realization of this plant is given by the following matrices:

$$A = \begin{bmatrix} -0.322 & 0.064 & 0.0364 & -0.9917 & 0.0003 & 0.0008 & 0 & 0 \\ 0 & 0 & 1 & 0.0037 & 0 & 0 & 0 & 0 \\ -30.6492 & 0 & -3.6784 & 0.6646 & -0.7333 & 0.1315 & 0 & 0 \\ 8.5395 & 0 & -0.0254 & -0.4764 & -0.0319 & -0.062 & 0 & 0 \\ 0 & 0 & 0 & 0 & -20.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -20.2 & 0 & 0 \\ 0 & 0 & 0 & 57.2958 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

4 NUMERICAL EXPERIMENTS WITH HIFOO

$$B_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \quad B_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 20.2 & 0 \\ 0 & 20.2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$C_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 57.2958 & 0 & 0 & -1 & 0 \end{bmatrix}$$

$$C_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -57.2958 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$D_{11} = D_{12} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$D_{21} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \quad D_{22} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

The goal is to find a static-output feedback controller of the form

$$K = \begin{bmatrix} k_1 & 0 & k_3 & k_4 \\ 0 & k_2 & 0 & 0 \end{bmatrix}.$$

4.4 STRUCTURED SOF CONTROL OF AN F-16 AIRCRAFT

Observe that this controller corresponds to the specific controller matrix, \hat{D} . They then provide two controllers for this system. The first is found using the LQ optimization approach of Stevens and Lewis [11] and the latter via \mathcal{H}_∞ loop-shaping. These controllers are given by the matrices

$$K_{LQ} = \begin{bmatrix} -18.6752 & 0 & 6.7479 & -25.379 \\ 0 & -0.568 & 0 & 0 \end{bmatrix}$$

$$K_{LS} = \begin{bmatrix} -17.5342 & 0 & 5.5001 & -28.0127 \\ 0 & -0.5311 & 0 & 0 \end{bmatrix}.$$

After converting the above system to an ss object, we compute a controller with the given structure utilizing HIFOO.

```
>> K = hifoo(P,options);
```

```
d =
```

```

          u1          u2          u3          u4
y1 -0.003702          0    0.7496   -0.4588
y2          0   -0.1909          0          0
```

The \mathcal{H}_∞ norms of each closed-loop system are calculated using the Matlab norm function via the command

```
>> f = norm(lft(P,K*),inf)
```

where K^* denotes each respective controller above. This information is given in Table 4.2. We now analyze the behavior of the open-loop plant as well as the effect of each controller

Controllers	K	K_{LQ}	K_{LS}
\mathcal{H}_∞ norms	1.0000	7.0305	9.0725

Table 4.2: Comparison of \mathcal{H}_∞ norms for closed-loop systems with different controllers on the closed-loop oscillations of the system. We first study the Bode magnitude plot of

4 NUMERICAL EXPERIMENTS WITH HIFOO

the system when controllers K_{LQ} and K_{LS} are connected with the open-loop system above in Figure 4.4. Both controllers provide similar performance, smoothing out the sharp peak around 80 rad/sec. We next plot the open-loop system along with the controller generated by HIFOO in Figure 4.5. Observe that the oscillations in the open-loop system are reduced dramatically by the controller found by HIFOO.

4.4 STRUCTURED SOF CONTROL OF AN F-16 AIRCRAFT

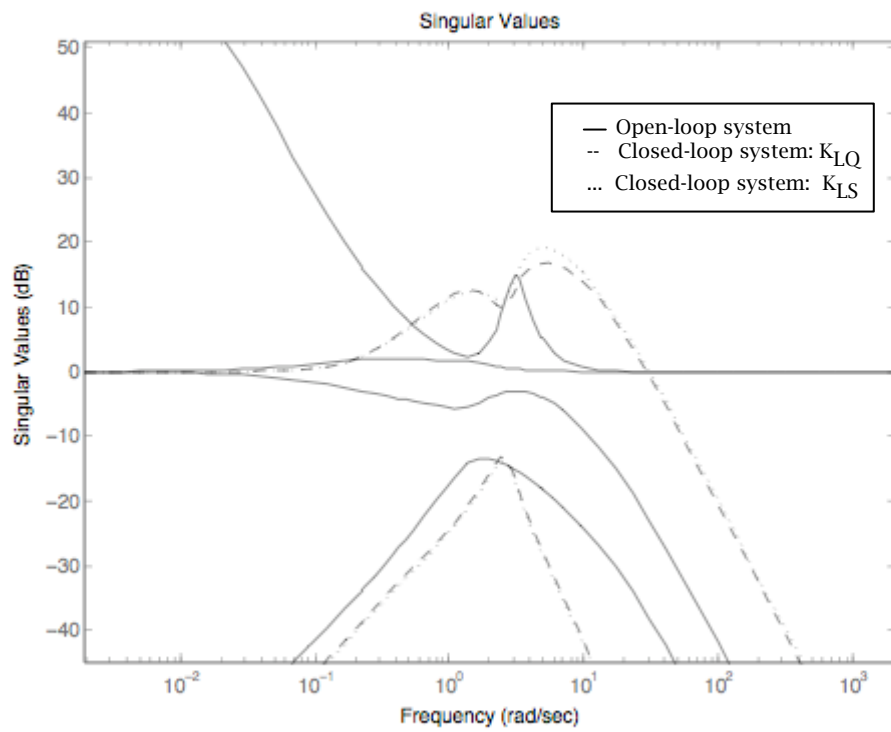


Figure 4.4: Closed-loop behavior for controllers K_{LQ} and K_{LS}

4 NUMERICAL EXPERIMENTS WITH HIFOO

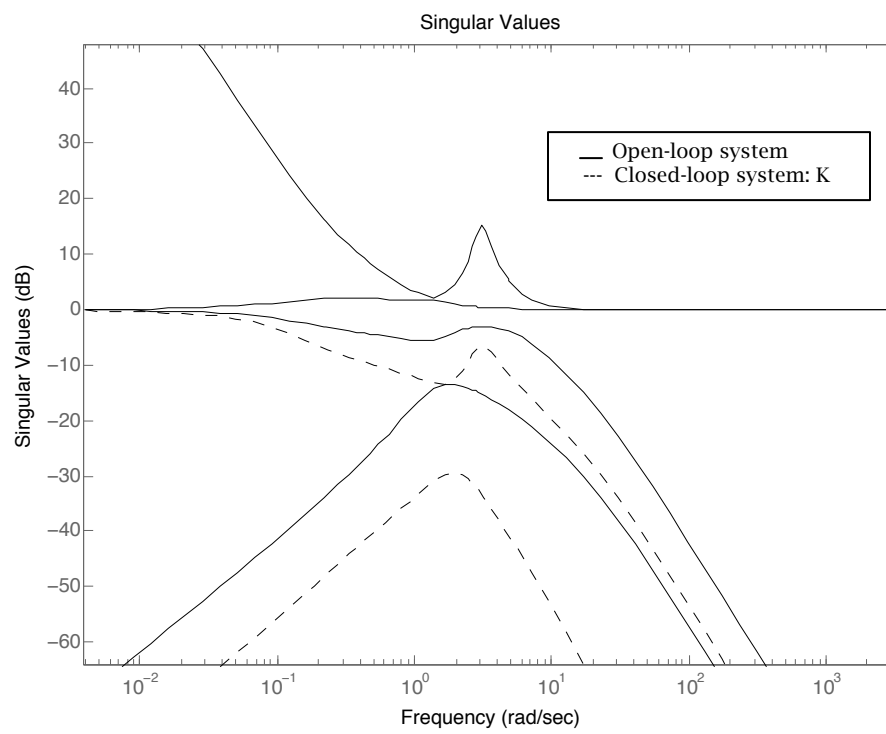


Figure 4.5: Closed-loop behavior for controller generated by HIFOO

FUTURE WORK

In the future, we hope to provide other useful extensions to HIFOO. One such possibility includes the ability to solve problems in simultaneous stabilization, such as Blondel's famous Chocolate Problem described in Blondel [2]. The user will be able to enter multiple plants to HIFOO, which will attempt to compute a single controller which stabilizes each plant separately. Another possible extension is the ability to require HIFOO to compute controllers that place the zeros and poles of the closed-loop transfer function in a specific region in the complex plane, such as inside (or outside) the unit disk. Finally, we plan to expand the objective functions HIFOO can optimize over to include both \mathcal{H}_2 and mixed $\mathcal{H}_2/\mathcal{H}_\infty$ norms. With the current advancements described in this work, as well as the planned extensions in the future, HIFOO could prove to be a useful tool in many control engineering applications.

A

MATHEMATICS

In this section we will review some basic results from linear algebra and functional analysis. For a more detailed introduction, see Lax [8] or Horn and Johnson [7].

A.1 Matrix Calculus

Let $F : \mathbb{C}^{n \times n} \rightarrow \mathbb{C}^{n \times n}$ denote a matrix-valued function. By analyzing the function $F(M + \Delta)$, where Δ is an arbitrary perturbation of matrix M , we loosely define the derivative of F with respect to matrix M as the term which is linear in the perturbation Δ . A formal definition can be given, utilizing the Kronecker product, however, that would be beyond the scope of this appendix.

We will first show the effect of a perturbation on the matrix inverse operator, which will be important later on.

Example A.1. Let $F(M) = M^{-1}$. We calculate the effect of a perturbation on F as follows

$$\begin{aligned} F(M + \Delta) &= (M + \Delta)^{-1} \\ &= ((M(I + M^{-1}\Delta))^{-1} \\ &= (I + M^{-1}\Delta)^{-1}M^{-1} \\ &= (I - M^{-1}\Delta + O(\Delta^2))M^{-1} && \text{(by geometric series)} \\ &= M^{-1} - M^{-1}\Delta M^{-1} + O(\Delta^2). \end{aligned} \tag{A.1}$$

A standard but very important result from linear algebra concerns the effect of a perturbation to the eigenvalues of a matrix. The result as stated and proved in Horn and Johnson [7, §6.3, Theorem 6.3.12] is paraphrased in the following:

Theorem A.1. *Let $A(t)$ be differentiable at $t = 0$. Assume that λ is an algebraically simple eigenvalue of $A(0)$ and that $\lambda(t)$ is an eigenvalue of $A(t)$, for small t such that $\lambda(0) = \lambda$. Let v be a right eigenvector of A and u a left eigenvector of A corresponding to eigenvalue*

λ , both normalized to 1. Then

$$\lambda'(0) = u^* A'(0) v.$$

Moreover, if $A(t) = A + tE$ for a fixed matrix perturbation, E , then for t small

$$\lambda(A + tE) = \lambda(A) + t u^* E v + O(t^2). \quad (\text{A.2})$$

Techniques discussed in Horn and Johnson [7] extend this result to perturbations of singular values as well, where v and u are now the corresponding left and right singular vectors of A .

An important example is provided by the maximum singular value function

$$f(A, B, C, D) = \bar{\sigma}(D + C(sI - A)^{-1}B)$$

where A, B, C, D are matrices of compatible dimensions. Note that the inner product on the space of matrices is defined as

$$\langle A, B \rangle = \text{tr}(A^* B)$$

where $\text{tr}(M)$ denotes the trace of matrix M . This equation arises naturally in control theory, as part of the \mathcal{H}_∞ norm of a given transfer function. See Section 1.3 for more details.

Example A.2. Let $f(A, B, C, D) = \bar{\sigma}(D + C(sI - A)^{-1}B)$

We will compute the derivative of f with respect to matrix A . Let $G = (sI - A)$ and let u and v denote the left and right singular vectors of $D + C(sI - A)^{-1}B$ corresponding to the singular value $\bar{\sigma}(D + C(sI - A)^{-1}B)$. We now compute the derivative by again analyzing the

A MATHEMATICS

effect of a perturbation to the A matrix.

$$\begin{aligned}
f(A + \Delta, B, C, D) &= \bar{\sigma}(D + C(sI - (A + \Delta))^{-1}B) \\
&= \bar{\sigma}(D + C((sI - A) - \Delta)^{-1}B) \\
&= \bar{\sigma}(D + C((G^{-1} + G^{-1}\Delta G^{-1})B) + O(\Delta^2)) \quad (\text{by Eq. A.1}) \\
&= \bar{\sigma}(D + CG^{-1}B + CG^{-1}\Delta G^{-1}B) + O(\Delta^2) \\
&= \bar{\sigma}(D + C(sI - A)^{-1}B) + u^*CG^{-1}\Delta G^{-1}Bv + O(\Delta^2) \quad (\text{by Eq. A.2}) \\
&= f(A, B, C, D) + \text{tr}(u^*CG^{-1}\Delta G^{-1}Bv) + O(\Delta^2) \\
&= f(A, B, C, D) + \text{tr}(G^{-1}Bvu^*CG^{-1}\Delta) + O(\Delta^2) \\
&= f(A, B, C, D) + \left\langle (G^{-1}Bvu^*CG^{-1})^*, \Delta \right\rangle + O(\Delta^2)
\end{aligned}$$

we therefore obtain the gradient with respect to matrix A

$$\nabla_A f = (G^{-1}Bvu^*CG^{-1})^*. \quad (\text{A.3})$$

A.2 The Space \mathcal{H}_∞

An important space in control theory is the Hardy Space, \mathcal{H}_∞ . Let

$F : \mathbb{C} \rightarrow \mathbb{C}^{n \times m}$ be an analytic, matrix-valued function. We define the norm of F as

$$\|F\| = \sup_{\text{Re}(s) > 0} \bar{\sigma}(F(s)). \quad (\text{A.4})$$

The Maximum-modulus theorem from complex analysis says that if F is analytic, it must obtain its maximum on the boundary. We can therefore rewrite the above norm as

$$\|F\| = \sup_{\omega \in \mathbb{R}} \bar{\sigma}(F(i\omega)) \quad (\text{A.5})$$

where the supremum is taken over the imaginary axis. We define the space \mathcal{H}_∞ as

$$\mathcal{H}_\infty = \{F \mid F : \mathbb{C} \rightarrow \mathbb{C}^{n \times m}, F \text{ is analytic}, \|F\| < \infty\}. \quad (\text{A.6})$$

The subspace of \mathcal{H}_∞ consisting of matrices whose entries are real, rational functions is denoted \mathcal{RH}_∞ . The space \mathcal{RH}_∞ describes an important class of functions corresponding to the bounded, stable transfer matrices as described in Section 1.1.2. This provides a theoretical approach to \mathcal{H}_∞ controller synthesis described in Francis [6].

B

CLOSED-LOOP SYSTEM CALCULATIONS

This appendix presents the calculation of the closed loop system for a controller with $D_{22} \neq 0$. Furthermore, we give an example of the new, required gradient calculations.

As in Chapter 2 we represent the plant, G , and controller, K , as the following systems of differential equations:

$$G: \begin{cases} \dot{x} &= Ax + B_1 w + B_2 u \\ z &= C_1 x + D_{11} w + D_{12} u \\ y &= C_2 x + D_{21} w + D_{22} u \end{cases} \quad (\text{B.1})$$

$$K: \begin{cases} \dot{d} &= \hat{A}d + \hat{B}y \\ u &= \hat{C}d + \hat{D}y \end{cases} \quad (\text{B.2})$$

We now close the loop by first substituting the system equation for y into the controller's equation for u to obtain

$$\begin{aligned} u &= \hat{C}d + \hat{D}[C_2 x + D_{21} w + D_{22} u] \\ u - \hat{D}D_{22}u &= \hat{C}d + \hat{D}C_2 x + \hat{D}D_{21} w \\ u &= \hat{C}d + \hat{D}C_2 x + \hat{D}D_{21} w \\ u &= [I - \hat{D}D_{22}]^{-1} (\hat{C}d + \hat{D}C_2 x + \hat{D}D_{21} w) \end{aligned}$$

and

$$y = C_2 x + D_{21} w + D_{22} [I - \hat{D}D_{22}]^{-1} (\hat{C}d + \hat{D}C_2 x + \hat{D}D_{21} w).$$

Introducing the notation, $F = [I - \hat{D}D_{22}]$, we now substitute our equation for u into our equation for \dot{x} to obtain:

$$\begin{aligned} \dot{x} &= Ax + B_1 w + B_2 [F^{-1} (\hat{C}d + \hat{D}C_2 x + \hat{D}D_{21} w)] \\ &= Ax + B_1 w + B_2 F^{-1} \hat{C}d + B_2 F^{-1} \hat{D}C_2 x + B_2 F^{-1} \hat{D}D_{21} w \\ &= (A + B_2 F^{-1} \hat{D}C_2) x + B_2 F^{-1} \hat{C}d + (B_1 + B_2 F^{-1} \hat{D}D_{21}) w. \end{aligned}$$

B CLOSED-LOOP SYSTEM CALCULATIONS

Similarly for \dot{d} and z we obtain

$$\begin{aligned}\dot{d} &= \hat{A}d + \hat{B}y \\ &= \hat{A}d + \hat{B} \left(C_2x + D_{21}w + D_{22}F^{-1} \left(\hat{C}d + \hat{D}C_2x + \hat{D}D_{21}w \right) \right) \\ &= \left(\hat{B}C_2 + \hat{B}D_{22}F^{-1}\hat{D}C_2 \right) x + \left(\hat{A} + \hat{B}D_{22}F^{-1}\hat{C} \right) d + \left(\hat{B}D_{21} + \hat{B}D_{22}F^{-1}\hat{D}D_{21} \right) w\end{aligned}$$

and

$$\begin{aligned}z &= C_1x + D_{11}w + D_{12}F^{-1} \left(\hat{C}d + \hat{D}C_2x + \hat{D}D_{21}w \right) \\ &= \left(C_1 + D_{12}F^{-1}\hat{D}C_2 \right) x + D_{12}F^{-1}\hat{C}d + \left(D_{11} + D_{12}F^{-1}\hat{D}D_{21} \right) w.\end{aligned}$$

Collecting all the terms into a single matrix, we obtain the compact representation of the closed-loop system

$$\begin{bmatrix} \dot{x} \\ \dot{d} \\ z \end{bmatrix} = \begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix} \begin{bmatrix} x \\ d \\ w \end{bmatrix} \quad (\text{B.3})$$

where

$$\begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix} = \begin{bmatrix} A + B_2F^{-1}\hat{D}C_2 & B_2F^{-1}\hat{C} & B_1 + B_2F^{-1}\hat{D}D_{21} \\ \hat{B}C_2 + \hat{B}D_{22}F^{-1}\hat{D}C_2 & \hat{A} + \hat{B}D_{22}F^{-1}\hat{C} & \hat{B}D_{21} + \hat{B}D_{22}F^{-1}\hat{D}D_{21} \\ C_1 + D_{12}F^{-1}\hat{D}C_2 & D_{12}F^{-1}\hat{C} & D_{11} + D_{12}F^{-1}\hat{D}D_{21} \end{bmatrix} \quad (\text{B.4})$$

and $F = (I - \hat{D}D_{22})$.

Not only does assuming a non-trivial feedthrough complicate the above system equations, it also requires additional chain-rule computations, as the above matrix is no longer linear in the controller variables. The \mathcal{H}_∞ norm defined in Equation A.5 of the closed-loop system can be viewed as a supremum over all frequencies of the composite function of controller variables, $\hat{A}, \hat{B}, \hat{C}$ and \hat{D} :

$$\bar{\sigma}(D_k(\hat{A}, \hat{B}, \hat{C}, \hat{D}) + C_k(\hat{A}, \hat{B}, \hat{C}, \hat{D})(sI - A_k(\hat{A}, \hat{B}, \hat{C}, \hat{D}))^{-1}B_k(\hat{A}, \hat{B}, \hat{C}, \hat{D})). \quad (\text{B.5})$$

To implement this in HIFOO, it is necessary to calculate the gradient of this objective function using the chain-rule. As an example, we will calculate the propagation of a perturbation of the \hat{D} variable on A_k and hence the objective function. An algebraic calculation utilizing

Equation A.1 gives the following:

$$A_k(\hat{A}, \hat{B}, \hat{C}, \hat{D} + \Delta) = A_k + \Delta_{A_k}$$

where Δ_{A_k} is given by

$$\begin{aligned} \Delta_{A_k} &= \begin{bmatrix} B_2 F^{-1} \Delta M C_2 & B_2 F^{-1} \Delta D_{22} F^{-1} \hat{C} \\ \hat{B} D_{22} F^{-1} \Delta M C_2 & \hat{B} D_{22} F^{-1} \Delta D_{22} F^{-1} \hat{C} \end{bmatrix} \\ &= \begin{bmatrix} B_2 F^{-1} \\ \hat{B} D_{22} F^{-1} \end{bmatrix} \Delta \begin{bmatrix} M C_2 & D_{22} F^{-1} \hat{C} \end{bmatrix} \end{aligned}$$

and $F = (I - \hat{D} D_{22})$ and $M = (I + D_{22} F^{-1} \hat{D})$. We now substitute Δ_{A_k} into the perturbation utilized in example A.2

$$f(A_k, B_k, C_k, D_k) + \left\langle (G^{-1} B v u^* C G^{-1})^*, \Delta_{A_k} \right\rangle + O(\Delta^2)$$

and expand the second term in the summation, replacing $(G^{-1} B v u^* C G^{-1})^*$ with ∇f :

$$\begin{aligned} \langle \nabla f, \Delta_{A_k} \rangle &= \left\langle \nabla f, \begin{bmatrix} B_2 F^{-1} \\ \hat{B} D_{22} F^{-1} \end{bmatrix} \Delta \begin{bmatrix} M C_2 & D_{22} F^{-1} \hat{C} \end{bmatrix} \right\rangle \\ &= \text{tr} \left(\nabla f^* \begin{bmatrix} B_2 F^{-1} \\ \hat{B} D_{22} F^{-1} \end{bmatrix} \Delta \begin{bmatrix} M C_2 & D_{22} F^{-1} \hat{C} \end{bmatrix} \right) \\ &= \text{tr} \left(\begin{bmatrix} M C_2 & D_{22} F^{-1} \hat{C} \end{bmatrix} \nabla f^* \begin{bmatrix} B_2 F^{-1} \\ \hat{B} D_{22} F^{-1} \end{bmatrix} \Delta \right) \\ &= \left\langle \left(\begin{bmatrix} M C_2 & D_{22} F^{-1} \hat{C} \end{bmatrix} \nabla f^* \begin{bmatrix} B_2 F^{-1} \\ \hat{B} D_{22} F^{-1} \end{bmatrix} \right)^*, \Delta \right\rangle \end{aligned}$$

giving us the new gradient with respect to the initial perturbation of controller variable \hat{A} as

$$\nabla f_{\hat{A}} = \left(\begin{bmatrix} M C_2 & D_{22} F^{-1} \hat{C} \end{bmatrix} \nabla f^* \begin{bmatrix} B_2 F^{-1} \\ \hat{B} D_{22} F^{-1} \end{bmatrix} \right)^*. \quad (\text{B.6})$$

Similar computations give the result for the other required gradients.

BIBLIOGRAPHY

- [1] D. Bates and I. Postlethwaite, editors. *Robust Multivariable Control of aerospace systems*. Delft University Press, 2002.
- [2] V. Blondel. *Simultaneous Stabilization of Linear Systems*. Lecture Notes in Control and Information Sciences 191. Springer, Berlin, 1994.
- [3] J. Burke, D. Henrion, A. Lewis, and M. Overton. HIFOO - a Matlab package for fixed-order controller design and \mathcal{H}_∞ optimization. 2006.
- [4] C.-T. Chen. *Linear System Theory and Design*. Oxford University Press, Inc., New York, NY, USA, 1999.
- [5] G. E. Dullerud and F. Paganini. *A course in robust control theory*, volume 36 of *Texts in Applied Mathematics*. Springer-Verlag, 2000.
- [6] B. A. Francis. *A Course in \mathcal{H}_∞ Control Theory*, volume 88 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, 1987.
- [7] R. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge, 1985.
- [8] P. Lax. *Linear Algebra*. John Wiley, New York, 1997.
- [9] F. Leibfritz. COMPL_eib: constraint matrix optimization problem library. Technical report, University of Trier, Germany, 2005.
- [10] S. O. R. Moheimani, B. J. G. Vautier, and B. Bhikkaji. Experimental implementation of extended multivariable ppf control on an active structure. *IEEE Transactions on Control Systems Technology*, to appear.
- [11] B. L. Stevens and F. L. Lewis. *Aircraft Control and Simulation*. Wiley, 1992.
- [12] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, 1996.