Efficient Design of Robust Controllers for $\mathcal{H}_2$ Performance

by

Kyle Yi-Ling Yang

B.S. Cornell University (1991)

S.M. Massachusetts Institute of Technology (1993)

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

Doctor of Philosophy

at the

Massachusetts Institute of Technology

February, 1997

Signature of Author ______________________________
Department of Aeronautics and Astronautics
October 2, 1996

Certified by ______________________________
Professor Steven R. Hall
Thesis Supervisor, Department of Aeronautics and Astronautics

Certified by ______________________________
Professor Michael Athans
Department of Electrical Engineering and Computer Science

Certified by ______________________________
Professor Eric Feron
Department of Aeronautics and Astronautics

Certified by ______________________________
Dr. Fernando G. Paganini
Department of Electrical Engineering and Computer Science

Certified by ______________________________
Professor Edward F. Crawley
Department of Aeronautics and Astronautics

Accepted by ______________________________
Professor Jaime Peraire
Chairman, Department Graduate Committee

Efficient Design of Robust Controllers for $\mathcal{H}_2$ Performance

by

Kyle Yi-Ling Yang

Submitted to the Department of Aeronautics and Astronautics
on October 2, 1996, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

This thesis presents a new method for designing robust, full order, dynamic, LTI controllers that provide guaranteed levels of $\mathcal{H}_2$ performance for systems with static, real, parametric uncertainties. Results are also discussed for systems with gain-bounded uncertainties. Dissipation theory, which is an extension of Lyapunov theory, is employed to analyze the performance of uncertain systems. For systems with real parametric uncertainties, the dissipation analysis is augmented by Popov stability multipliers. Popov multipliers reduce the conservatism of the analysis and do not overly complicate computations.

The problem of synthesizing robust $\mathcal{H}_2$ controllers, given fixed stability multipliers, is reformulated in a systematic manner. For fixed multipliers, it is shown that the synthesis problem is convex and can be solved by minimizing a linear cost function, subject to linear matrix inequality (LMI) constraints. Unfortunately, for large problems this formulation is found to be unusable with state-of-the-art software. Therefore, a second synthesis method is derived. The second method hinges upon separating a bound on the closed loop $\mathcal{H}_2$ cost into two parts: the first part deriving from a full information control problem and the second part deriving from an output estimation problem. This separation principle is used to derive sufficiency conditions for the existence of a robust controller. The sufficiency conditions have the form of coupled Riccati equations. Two new iterative techniques are developed to solve coupled Riccati equations efficiently. The problem of finding stability multipliers to analyze the system, given a fixed controller, is then reformulated without the use of LMIs. A solution technique is introduced for this problem that uses a gradient optimization technique. The analysis and synthesis routines are combined to form a $D-K$ iteration to design robust controllers. The design methodology is successfully demonstrated on a small benchmark problem as well as realistic structural control problems: two models of the Middeck Active Control Experiment (MACE).

Thesis Committee:

Steven R. Hall, Sc.D., Associate Professor of Aeronautics and Astronautics
Michael Athans, Ph.D., Professor of Electrical Engineering
Eric Feron, Ph.D., Assistant Professor of Aeronautics and Astronautics
Fernando G. Paganini, Ph.D., Visiting Scientist, Electrical Engineering
Edward F. Crawley, Sc.D., Professor of Aeronautics and Astronautics

Acknowledgments

I am grateful to the members of my thesis committee and, in particular, to my advisor, Professor Steven Hall, for helping me with this research. Without Steve's guidance and dogged determination, this research would not have been possible. Steve was always willing to share his deep understanding of controls and mathematics by discussing problems at even the lowest level. I am grateful for the active advice of Professor Eric Feron, who opened the world of LMIs to us and helped me define the new analysis algorithm. I would like to thank Professor Edward Crawley, who, as Director of the MIT Space Engineering Research Center (SERC), allowed me the freedom to pursue almost purely theoretical research. Professor Michael Athans, with his frank and insightful comments, foretold problems and helped me keep my thesis relevant to the real world. Dr. Fernando Paganini's valuable new perspective on our research helped me better understand where this research fits into the broader realm of robust control.

I would like to thank Mr. Carolos Livadas for countless hours of loyal toil with me at lab. His excellent and incredibly thorough work, both in controls and in all matters computer-related, provided me much comfort and aid. I am pleased to have him as friend.

The graduate students in SERC are a terrific bunch of engineers and I wish I had room to thank them all individually. I would like to thank in particular Mr. Simon Grocott, for his great willingness to help with the MACE models and to discuss a variety of theoretical matters. I would also like to thank Dr. Mark Campbell, Mr. Brett Masters, and Dr. Leonard Lublin for their aid and advice. Also I'd like to thank the many terrific members of the staff at Aero/Astro who've worked with me.

I am very grateful to Dr. David Miller, Director of the MIT Space Systems Laboratory, who generously funded me on NASA training grant NGT-132. Ms. Sharon L. Brown worked to make this possible. This project was also supported by NASA SERC grant NAGW-1335 and by contract DLH 476511 from the Charles Stark Draper Lab. I am also grateful to the National Science Foundation, for awarding me a Graduate Research Fellowship.

I would like to thank my dear friends Greg Cabe, Sarah Stock, Emily Sun, Steve Lee, Chris van Schalkwyk, Push Singh, and Mike Biorn, each of whom has, at one time or another, contributed to making life in Boston happy and memorable. Last, but not least, I would like to thank my brother, and, of course, my parents for, well, everything.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

EVP	LMI eigenvalue problem - consists of a linear cost function subject to LMI constraints
FI	Full information
FSFB	Full-state feedback
LMI	Linear matrix inequality
LTI	Linear, time-invariant
MIMO	Multiple-input, multiple-output
OE	Output estimation
OF	Output feedback
SISO	Single-input, single-output
SN	Structured norm (generalized structured singular value)

Symbols

A	plant dynamics matrix
A_c	controller dynamics matrix
B_c	controller input matrix
B_d	plant exogenous (white) disturbance input matrix
B_u	plant control input matrix
B_w	plant uncertainty loop input matrix
C_c	controller output matrix
C_e	plant state performance weighting matrix
C_y	plant measurement output matrix

C_z	plant uncertainty loop output matrix
D_{yd}	plant feedthrough matrix, exogenous (white) disturbance to measurement
D_{eu}	plant control performance weighting matrix
E_d	$D_{yd}D_{yd}^T$
E_u	$D_{eu}^T D_{eu}$
H	one of the Popov stability multipliers matrices, diagonal and $H > 0$
J	the cost function, the square of the worst-case $\mathcal{H}_2$ cost for LTI uncertainties
$\mathcal{J}$	a bound on the cost function
M_1	Lower sector bound in Popov analysis
M_2	Upper sector bound in Popov analysis
M_d	$(M_2 - M_1)^{-1}$
N	one of the Popov stability multiplier matrices, diagonal and $N \geq 0$
P	a Lyapunov matrix in the LMI formulation
Q	a matrix in the LMI formulation, typically P^{-1}
R	Lyapunov matrix in a FSFB or FI problem
$\mathcal{R}^{p\times q}$	set of real-valued matrices, of dimension $p \times q$
$\mathcal{S}^{p\times q}$	set of real-valued, symmetric matrices, of dimension $p \times q$
U	a term in the Elimination Lemma
V	a term in the Elimination Lemma
W_X	a basis for the nullspace of a matrix X
X	a Lyapunov matrix in a control problem
Y	a Lyapunov matrix in an estimation problem
Z	$\zeta + NC_zB_uE_u^{-1}B_u^TC_z^TN$
d	exogenous (white, unit intensity, zero mean) disturbance, implicitly $d(t)$
h_i	the ith nonzero element of H
n	number of states in the plant
n_d, n_u, etc.	size of signal d, of signal u, etc.

n_i	the ith nonzero element of N
u	control signal, implicitly $u(t)$
w	output from uncertainty block, implicitly $w(t)$
x	state variable, implicitly $x(t)$
y	measurement, implicitly $y(t)$
z	input to uncertainty block, implicitly $z(t)$
γ	$\mathcal{H}_\infty$ gain, $w \to z$, of nominal plant
Δ	the uncertainty block, can be real and diagonal or full and complex depending on the problem
ζ	$M_d H + H M_d - N C_z B_w - B_w^T C_z^T N$
Θ	matrix of unknowns in the Elimination Lemma
μ	the structured singular value
ξ	$\zeta - N C_z B_u E_u^{-1} B_u^T C_z^T N$
σ	scalar stability multiplier for the $\mathcal{H}_2/SN$ problem
Ψ	a term in the Elimination Lemma

Subscripts, Superscripts, and Modifiers

$(\tilde{\cdot})$	closed loop, *i.e.*, for system with loop closed between the nominal plant and the controller
$(\cdot)_c$	being of the controller, as opposed to the plant
$(\cdot)_{ij}$, $(\cdot)_i$	the (i, j) entry of a matrix or the (i, j) block entry of a matrix, the ith entry of a vector
$(\cdot)^\dagger$	pseudo-inverse

Chapter 1

Introduction

Future, high-fidelity measurement equipment on board satellites and other space systems will require a low vibration environment for successful operation. To increase precision, devices such as mirrors on interferometers will need to be actuated at ever wider frequency bandwidths. If the bandwidth of onboard actuation systems is increased to overlap with the satellite's structural modes, control-structure interactions could occur and degrade the instrument's performance. Linear, multivariable control systems could then be employed to control the resulting vibrations in the structure.

Optimal control systems are highly tuned to the model of their plant. Consequently, closed loop performance and even stability are highly dependent on the accuracy of design models. Of course, even the best models of a structure are only approximations of the actual hardware. Two types of uncertainties exist in a model. This first type involves unmodeled or non-parametric uncertainties. The designer attempts to deal with these, and other "unknown unknowns," by limiting or restricting the controller bandwidth. The second type of uncertainty is modeled or parametric uncertainty. These uncertainties are relatively well characterized and may even be anticipated — these are the "known unknowns." One example of a parametric uncertainty is the stiffness of a beam in a truss. The designer may have an expected range of values for this parameter. Designing optimal control systems to handle modeled and parametric uncertainties is the domain of *robust control* and the subject of this research.

Designing robust control systems can take significantly more time and computational power than designing an equivalently-sized, non-robustified controller for the same plant. It is desirable to reduce these burdens for several reasons. First, control design is an iterative process, and engineers need to have adequate time to simulate and test their controllers. If robust controllers are perceived as taking too long to design, then engineers will not attempt to use them. Robust design tools must be made nearly as easy to use as more standard design tools.

Secondly, if the computational burden of robust design is reduced, then engineers will be able to perform designs for practical, higher-order systems. Limits on computer memory and other issues have relegated some robust design methodologies to the realm of simple, textbook problems.

Lastly, it is desirable, particularly on a space system or any system that is expected to age or change with time, to be able to do periodic system identifications. System identification data can be used to update models of the plant. New controllers can then be designed for the updated model. Here, it clearly is desirable for the redesign to take as little time as possible, so that the new controller can be implemented.

This research focuses on control systems for $\mathcal{H}_2$ performance. An $\mathcal{H}_2$ performance metric is appropriate for systems whose performance is judged by calculating the root-mean-square (RMS) value of a signal over frequency. The system uncertainties (the "known unknowns") that will be examined are of two types. The first is a real, parametric uncertainty, such as the previously mentioned stiffness. The second is an $\mathcal{L}_2$ gain bounded uncertainty, which is also a natural representation for certain exogenous disturbance signals.

Robust controllers for such systems have previously been designed and successfully implemented in structural control experiments by How *et al.*[46,50] These controllers are referred to as $\mathcal{H}_2$/Popov or $\mathcal{H}_2$/*SN* controllers, depending on the type of uncertainty. To date, these controllers have typically been designed by performing a gradient search of an augmented cost function. The augmented cost function consists of a bound on the performance objective and related constraints on performance and stability robustness. Although the method should work in theory, in practice, it is difficult to use because of the large computational effort involved. For large, structural control problems, it can take days to solve for a controller.

It should be noted that this work concerns linear, finite-dimensional, time-invariant (LTI) plants and controllers. All work is performed in continuous, rather than discrete time. Also, some terminology, particular to this thesis must be clarified. In this thesis, "analysis" refers to the process of determining the performance and/or stability characteristics of a fixed (possibly closed loop) system. Typically, this involves the determination of optimal weights (*i.e.*, scalings or stability multipliers) for the uncertainty blocks to reduce the conservatism of the analysis. In contrast, "synthesis" is any process that finds a controller, given fixed weights. Finally, a controller "design" is a process by which a controller is found and associated weights (*i.e.*, scalings or stability multipliers) on the uncertainty are found.

The overall goal of this thesis is to develop novel design algorithms for robust $\mathcal{H}_2$ controllers that are **faster or more effective** than existing techniques for practical systems. By robust, we mean that the controller should be able to stabilize the system and provide *a priori* guarantees on a perturbed system's closed loop $\mathcal{H}_2$ performance.

1.1 Background and Literature Review

To survey the literature on robust control, we must categorize a paper based on three issues: the types of uncertainties that it deals with, the performance measure that it uses, and whether it deals with controller analysis or synthesis. This thesis focuses on two types of uncertainties, each of which has a distinct body of literature surrounding it. We are primarily interested in real, parametric uncertainties in the plant. A secondary type of uncertainty that we will examine is merely restricted to have a bounded $\mathcal{L}_2$-induced norm. We refer to these uncertainties as gain-bounded. Regarding the performance metric of interest, we are interested in a metric that, in the case of linear uncertainties, is equal to the $\mathcal{H}_2$ norm of a system. This stands in contrast to the more typical $\mathcal{H}_\infty$ performance metric that is used in most robust performance frameworks. Wrapped around the questions regarding the types of uncertainty and performance metrics are the questions of whether we wish to synthesize new controllers or to merely analyze the performance of existing, closed loop systems. Our goal is to develop new design techniques for linear, multivariable controllers. Thus, we will survey the literature on both analysis and synthesis. Each of the above topics has been individually investigated at some level by previous researchers, and we will review their works below.

1.1.1 Uncertainties and Robust Stability

For a structure, one example of a real, static, uncertain parameter is an uncertain stiffness. The determination of whether a structure is robustly stable to such an uncertainty is complicated by the fact that the uncertain parameters are real, rather than complex. This phase information is difficult to describe mathematically if the uncertainty block is structured. In fact, the problem has been shown to be NP-hard.[9,83] Because stability robustness cannot be predicted exactly, approximate tests are used. Approximate tests are conservative in the sense that they will always predict if a system can be destabilized, but may also falsely predict that a robustly stable system can be destabilized. The most famous of these methods is to calculate an upper bound for the real or mixed structured singular value (μ).[24,84,83] To prove that a system is robustly stable, the test requires that a frequency-domain inequality based on the nominal system transfer function be satisfied.

In practice, when used as part of a design algorithm, μ analysis has several drawbacks. First, the inequality may need to be checked at a large number of frequency points to provide confidence in or to reduce the conservatism in the test. This can require a great deal of time. Secondly, once the values of μ and any frequency-dependent scalings have been determined at a number of points, in order to use the information for, say, controller synthesis, one must fit a finite-dimensional transfer function to the data. This requires the use of curve-fitting algorithms, which can be difficult to use. When the transfer functions of the scalings are incorporated into the system, the overall system can be much higher order than the original system.

Because μ analysis is relatively difficult to use for design, a great deal of effort has gone into developing simpler, easy-to-use, state space tests that are not overly conservative. Most of these tests are founded upon the fundamental concepts of Lyapunov stability. Because we are concerned with nominally linear systems, these robust stability tests use quadratic Lyapunov functions. However, such functions can lead to overly conservative results, so much research has been performed on reducing their conservatism. Perhaps the most fruitful avenue of research has been the search for Lyapunov functions that indirectly depend on the uncertainties in the system.[41] One example of a parameter-dependent, Lyapunov-based test comes from the classical Popov criterion. A Popov analysis allows the mathematics to treat sector-bounded nonlinearities in the uncertainty blocks. A Popov criterion is essentially a real μ analysis performed with a particular, fixed function for the weights.[49] This implies that for certain classes of uncertainties, an analysis based on the Popov criterion is equivalent to a real μ analysis. The state space Popov test for robust stability requires that a Lyapunov matrix and its associated stability multipliers (*i.e.*, the weights in the real μ test) exist and solve a particular Riccati equation.[42,41] The Lyapunov approach to robust stability analysis has certain important advantages. The search for the Lyapunov matrix and the associated stability multipliers is convex. In particular these convex problems can be conveniently written in the form of linear matrix inequalities (LMIs).[1,8,67] Because off-the-shelf codes can be used to solve LMI problems in polynomial time, this can make these types of tests more convenient than a real μ analysis.

Of course, the use of the Popov criterion leads to just one form of parameter-dependent Lyapunov function. More complicated Lyapunov functions can be formed, and, in some cases, can reduce conservatism in the analysis. Useful instances of more complex functions are discussed in Refs. 26 and 27. The difficulty with these proposed analyses is that the size of the problem (both the number of variables as well as the size of the LMI constraints) must grow in order to achieve the desired improvements.

It should also be mentioned that other researchers have documented the relationship between real μ and concepts such as the passivity of an uncertain system.[1]

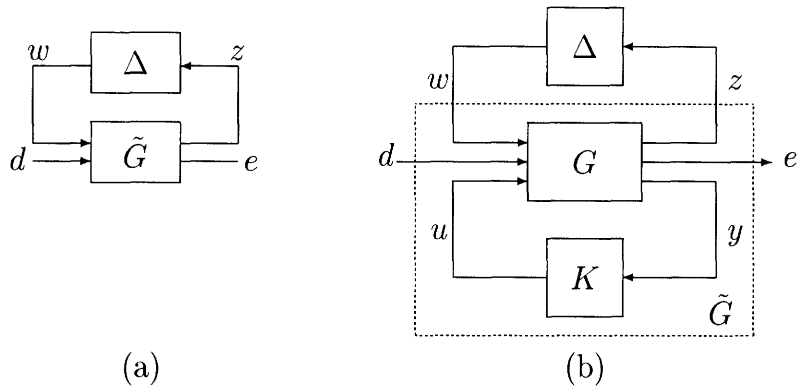

Figure 1-1: a. Analysis problem - uncertain system is represented as a nominal system wrapped into a feedback loop with an uncertainty block. b. Synthesis problem - uncertain system with controller feedback loop. G and $\tilde{G}$ are known, LTI systems.

From the above discussion, it seems clear that when Lyapunov type tests are made more complex in an intelligent fashion, the conservatism of the test can be reduced. However, the difficulty with adding complexity to a test is that, eventually, the test becomes so complicated that it cannot be used for practical systems. Either it takes too long or requires too much memory. An extreme example of can be considered. Given n_w parametric uncertainties, to check the robust stability of a linear system, one can check the stability of the system at each of the boundaries of the unknown parameters. Each of these boundary points yields a constraint that can be written in the form of an LMI. The resulting test is no longer conservative. However, one must perform 2^{n_w} tests! The size of the analysis problem depends exponentially on the number of uncertainties. This is unacceptable for use as a general method, because such tests would take too long to perform for systems with a reasonable number of uncertainties. Tests that grow exponentially with the number of uncertainties continue to be proposed in the literature (see, for example, Ref. 29). Such tests are not treated further in this work because of their limited practical utility.

A second type of uncertainty block that we wish to consider is the bounded-gain uncertainty. As shown in Figure 1-1a, an uncertainty block, Δ, will be considered if the $\mathcal{L}_2$ norm of w is less than a known value, γ, for an input signal, z, with a unit $\mathcal{L}_2$ norm. If the uncertainty is linear, than this corresponds to an $\mathcal{H}_\infty$ norm, *i.e.*, $\|\hat{\Delta}\|_\infty < \gamma$. The robust stability problem with this type of uncertainty is typically analyzed using complex μ tests.[65] This is the same as a real μ analysis problem with one of the scaling functions set to zero. If the uncertainty is nonlinear and/or time-varying, then such systems are analyzed using a generalization of μ known as the structured norm (*SN*).[18] Checking if a system is robustly stable to these uncertainties is equivalent to checking the BIBO stability of a system with to an exogenous disturbance signal. Furthermore, a robust stability analysis of such systems can be treated using a Popov analysis by setting some of the stability multipliers to zero.[49] This is because the bounded-gain uncertainty can be considered a sector-bounded nonlinearity with a particular set of sector limits.

A final point to consider regarding stability analysis is whether conservatism can be reduced by using a higher order test. Methods used to overbound μ or *SN* should be interpreted as quadratic tests on the system transfer function. Likewise, the Lyapunov results that have been discussed rely upon quadratic Lyapunov functions. It is tempting to believe that, for instance, a quartic stability test could be used instead.[4] However, evidence suggests that higher order tests cannot be relied upon to reduce conservatism, as the order of the test may be required to grow with the number of uncertainties.[45]

1.1.2 Robust $\mathcal{H}_2$ Performance Analysis

We now turn to a discussion of the performance metric we wish to optimize. Robust performance analysis has traditionally been performed using μ techniques, which can guarantee robust performance with an $\mathcal{H}_\infty$ metric.[65] In contrast, for an LTI system, our performance metric equals the the system $\mathcal{H}_2$ norm. The $\mathcal{H}_2$ norm is of interest both for historical reasons and because, for some systems, an $\mathcal{H}_2$ metric will more accurately reflect the desired objectives than an $\mathcal{H}_\infty$ metric would.

Because of the uncertainties in the system, we will need to extend the definition of the $\mathcal{H}_2$ performance metric to include nonlinear systems. For an excellent discussion of the performance metric and its definition for an uncertain system, the reader is referred to Ref. 74. For linear uncertainties, our performance metric is equivalent to the $\mathcal{H}_2$ norm, even in the multivariable case. This contrasts with performance metrics such as the generalized $\mathcal{H}_2$ norm of Rotea.[71]

Calculating the worst possible $\mathcal{H}_2$ performance of a system with real, parametric uncertainties is a robust performance problem. It is at least as difficult as determining if such a system is robustly stable. Thus, the robust performance problem is also NP-hard, and the worst-case performance of the system cannot be calculated exactly. Instead, it must be bounded from above. The problem of bounding the cost has been examined in many works including Refs. 28, 26, 66.

To be specific, we wish to examine the robust performance of the system, $\tilde{G}$ pictured in Figure 1-1a. This is the most general formulation of the robust performance problem because the nominal system has two distinct inputs and two distinct outputs. With d considered to be a white noise input to the system, the $\mathcal{H}_2$ performance is calculated from the error signal, e. Meanwhile, the nominal system is wrapped into a feedback loop with the uncertainty block, Δ, via z and w. This is the formulation considered by How.[46] How bounds the $\mathcal{H}_2$ cost of this system using Popov stability multipliers. His method yields an upper bound on the worst-case $\mathcal{H}_2$ performance of the system, given uncertainties within a specified range. The Popov robust performance analysis relies upon the same theory that was used for the Popov robust stability analysis. It requires the solution of a Riccati equation for an associated Lyapunov matrix, given a set of stability multipliers. The cost bound is obtained by recognizing that the Lyapunov matrix provides an upper bound to the worst-case observability Gramian of the system. The bound is equivalent to the bound discussed Haddad and Bernstein in Ref. 41. How's bound can be computed at a reasonable cost and has proven to be effective for structural control.[50]

Controllers designed using a Popov analysis are tested experimentally on a structural system in Ref. 50. The results demonstrate that a Popov analysis was not overly conservative for a system with static, real-parametric uncertainties. Additionally, the results of Ref. 7 suggest that Popov's criterion should provide reasonable bounds on the $\mathcal{H}_2$ performance for a wide variety of uncertain systems.

If desired, one can reduce the conservatism of a Popov performance analysis by using more complex multipliers. In Ref 26, Feron is able to reduce the conservatism of a robust $\mathcal{H}_2$ performance analysis compared to a Popov-type analysis by using non-causal stability multipliers. However, again, this comes at the cost of increasing the size of the analysis problem.

The situation for a system with a gain-bounded (as opposed to the real, parametric) uncertainty set is similar. As before, typically, only a bound on the worst case performance is calculated. However, recent results by Paganini[68] indicate that nearly exact bounds can be calculated via a set of LMIs evaluated at a finite set of frequency points. However, the accuracy of the calculation depends on the number of frequency points to be evaluated. Alternatively, this formulation can be modified to involve an LMI optimization over a predefined, finite set of basis functions, but this convenience comes at the cost of some conservatism.[36]

For the system with two inputs and two outputs pictured in Figure 1-1a, with Δ representing a gain-bounded uncertainty block, the same performance analysis tools used for the real, parametric case can be used. As in the analysis of stability robustness, this essentially involves setting parts of the stability multipliers to zero. These bounds are easily computable and based on quadratic Lyapunov functions. Furthermore, the bounds are well understood.[44,74,26]

1.1.3 Controller Design and Synthesis

A variety of methods have been used in practice to try and make LQG controllers less sensitive to parameter variations in structural systems. For a survey of the most promising of these techniques, the reader is referred to Ref. 38. However, these methods (such as the sensitivity-weighted LQG method or the multiple-model method), unfortunately, do not fulfill the criteria that we have stated a robust controller should meet. They do not provide *a priori* guarantees that the system will be robust nor can they provide a bound on the performance of the system.

One design method that will guarantee that the resulting controller is robustly stable is commonly referred to as a "$D-K$ iteration."[85] A $D-K$ iteration consists of alternating synthesis and analysis steps. In the analysis, or D, step, the controller is held fixed and optimal weights (stability multipliers) are found to calculate the performance of the closed loop system. In the synthesis, or K step, the weights are held fixed and a controller is synthesized. One advantage of this design methodology is that at each step, the closed loop cost should decrease monotonically. Also, a $D-K$ iteration is straightforward to use. The major disadvantage of the method is that the iteration is not convex. It can converge to a local minima and can even fail to converge. Nevertheless, $D-K$ iteration is well established as a standard design tool in the robust control community. It is, for instance, the method used to design robust, $\mathcal{H}_\infty$ controllers in most so-called "μ synthesis" algorithms.[2]

Not all robust control design techniques rely upon $D-K$ iteration. The controller design problem pictured in Figure 1-1b is investigated in detail by How.[46] In this case, Δ is a real, parametric uncertainty and the goal is to find a controller, K, that minimizes the worst-case $\mathcal{H}_2$ norm of the system, as measured at e. Because How uses Popov stability multipliers to describe the uncertainty blocks, his designs are referred to as $\mathcal{H}_2$/Popov controllers. How sets up an augmented cost function, composed of a bound on the $\mathcal{H}_2$ cost and an associated, nonlinear, matrix equality constraint on the performance. This augmented cost function is then minimized using a gradient search algorithm. This technique solves for both the controller and weighting functions (stability multipliers) simultaneously. This can be a problematic optimization, as it can be difficult to find an initial guess for the unknown parameters. The stability multipliers are not motivated by any physical or heuristic rules. In contrast, initial guesses for a controller could be, for instance, the non-robust LQG or $\mathcal{H}_\infty$ controllers. For this reason, it seems that it should be useful to examine a $D-K$ iteration approach for the $\mathcal{H}_2$/Popov design.

If the uncertainty structure is merely gain-bounded, rather than real and parametric, then the design problem in Figure 1-1b is called the $\mathcal{H}_2/SN$ problem.

We turn now to the problem of controller synthesis, *i.e.*, finding a controller for a given set of weights. A significant body of literature has developed for controller synthesis in the face of an $\mathcal{H}_\infty$ bounded uncertainty. This problem was first considered by Bernstein and Haddad,[5] who called it the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ problem. They set up an augmented cost functional in terms of bounds on the system covariance, rather than in terms of a Lyapunov matrix. Full and reduced-order controller synthesis was performed by solving the necessary conditions for optimality, a set of four coupled nonlinear matrix equations.[6] These are typically referred to as "the optimal projection equations." The solutions of these equations yielded upper bounds on the worst-case state covariance of the system, thus providing bounds on the $\mathcal{H}_2$ cost. However, the analysis was limited to the case in which the disturbance inputs, d and w, were identical. This is significant because it leads one to a different interpretation of the performance metric and the disturbance signals. With only one input, after having found the controller, it is possible to exactly calculate the closed loop system's exact $\mathcal{H}_2$ performance. With two disturbance inputs simultaneously acting on the system, this calculation is not possible, and only the bound on the performance is known. Furthermore, in order to try and reduce the conservatism of the control design, different, possibly dynamic, weights may be needed for the d and w inputs. This is not possible unless the inputs are explicitly separate.

As just mentioned, the optimal projection equations of Ref. 5 are a set of coupled, nonlinear matrix equations. They are also referred to as coupled Riccati equations, because when some of the variables are held constant in each equation, the equations take the form of Riccati equations in the remaining unknowns. The coupled Riccati equations can sometimes be solved by iteratively solving Riccati equations — holding some terms constant and solving for the remaining variables using standard Riccati solvers.[5] The second standard way to solve these equations is with a homotopy method.[15] Neither of these methods is guaranteed to yield a solution to the problem.

The problem of state feedback controller synthesis for the $\mathcal{H}_2/\mathcal{H}_\infty$ case with only a single disturbance input was discussed by Rotea and Khargonekar.[72] They demonstrated that, in this case, the solution for the optimal controller gain was an inherently convex problem. They showed that when this control gain was mated to an $\mathcal{H}_\infty$ estimator for a related problem, the resulting controller solved the single input $\mathcal{H}_2/\mathcal{H}_\infty$ problem.[56] This demonstrated that the controller had a useful separation structure. A dual form to the single input problem, with separate inputs but only a single performance variable, was discussed by Doyle *et al.*[22] Of significance, this controller was again derived using a separation principle.

The single input formulation has since been expanded to include dynamic scaling functions to make the bounds less conservative for real, parametric uncertainties.[43]

Ref. 5 also notes that the single input $\mathcal{H}_2/\mathcal{H}_\infty$ problem is a generalization of the maximum entropy controller,[63,62] *i.e.*, the central $\mathcal{H}_\infty$ controller.[23,85] If the outputs from the system in the single input $\mathcal{H}_2/\mathcal{H}_\infty$ formulation are assumed to be the same, then one of the optimal projection equations becomes superfluous. Furthermore, the remaining two decouple, leaving only the standard pair of $\mathcal{H}_\infty$ output feedback Riccati equations. Thus, the $\mathcal{H}_\infty$ control problem is the same as our problem of interest in the case that the two inputs are specified to be identical and the two outputs are also specified to be identical.

It is worthwhile to point out the connections between the $\mathcal{H}_2/SN$ problem and a closely related problem, typically called the General Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Problem, first introduced by Ridgely *et al.*[70] It seeks to minimize the *nominal* $\mathcal{H}_2$ performance of the system while guaranteeing stability robustness against $\mathcal{H}_\infty$ bounded uncertainties. This formulation cannot be used to guarantee the robustness of a system because it does not generate an overbound on the worst-case observability or controllability Gramian that can be encountered. Nevertheless, a great deal of work has been done on this problem. For instance, the controller synthesis problem has been shown to be convex,[76] solvable via LMIs,[12] and able to be solved exactly in the discrete-time SISO case.[75] A comprehensive review of these nominal $\mathcal{H}_2$ performance systems is beyond the scope of this work.

We now turn to briefly survey the literature on the usage of LMIs for controller synthesis and design. Recently, a great deal of work has been done to synthesize non-robust controllers using LMIs. Dynamic, output-feedback controllers that satisfy an $\mathcal{H}_\infty$ performance metric can be synthesized by using the Bounded Real Lemma.[33,52] However, for an $\mathcal{H}_2$ performance metric, the Lemma is not directly applicable, and controller synthesis methods that rely upon LMIs have been limited to state-feedback controllers,[25] or to static, output-feedback controllers.[53,34,13] Solving the $\mathcal{H}_2$/Popov problem or the $\mathcal{H}_2/SN$ problem requires the synthesis of a dynamic, $\mathcal{H}_2$, output-feedback controller, which is more complex than the previously mentioned controllers. A method to solve the dynamic, $\mathcal{H}_2$, output-feedback problem using LMIs is derived in this thesis. The method formed the basis for results presented by Livadas.[57]

Unfortunately, LMIs cannot be used to directly perform a robust controller design, because the resulting constraints are bilinear in the any unknown controller parameters and the unknown scalings (stability multipliers). (This fact will be made clear in this thesis in Chapter 4.) Consequently, some research has been done to try and directly solve bilinear matrix inequalities.[35] However, to date, effective algorithms do not exist to solve bilinear matrix inequalities, so the controllers must be designed by solving alternating LMI problems. Of course, solving alternating LMI problems makes the design method have the same complexity as a $D - K$ iteration that uses LMIs. Such a $D - K$ iteration, where both the analysis and synthesis steps require the solution of LMI problems, is used to design controllers in Ref 27. In this reference, the analysis step uses a more sophisticated multiplier than is used for a Popov analysis. The synthesis step uses the methods of Ref. 33 to derive $\mathcal{H}_\infty$ controllers. However, the results of the paper do not deal with $\mathcal{H}_2$ performance. Furthermore, it is interesting to note that the design methodology is demonstrated on an example system with only two states.

Another robust control design methodology that relies upon solving LMI problems in a $D - K$ iteration has been posed by D'Andrea.[19,20] The methodology uses the analysis technique developed by Paganini,[68] to bound the $\mathcal{H}_2$ performance of the system. The synthesis technique requires the solution of an LMI feasibility problem. However, the number of LMI constraints must increase to achieve a reduction in the conservatism of the performance bound, implying that the computational cost of the method can be high. It should also be mentioned that this synthesis technique is only demonstrated on an example system with one state in Ref 19.

Finally, we mention the separation properties of controllers. It is well known that for a given, fixed LTI system, any output feedback controller that stabilizes the system inherently possesses the ability to be separated into a state feedback gain and an output estimator (see, for instance, Refs. 58 or 18). This can be discerned via a

Youla parameterization of the controller. Note that, in general, unless the separation structure of a controller is explicitly used for its construction (*i.e.*, an estimator is found, then a control gain is found), then, to the best of the author's knowledge, this separation structure is ignored. The separation structure exists but apparently is not useful. Two obvious examples where the separation principle plays a significant role in controller synthesis are the $\mathcal{H}_2$ (LQG) controller and the $\mathcal{H}_\infty$ controller.[23] Of great interest to this research, as already mentioned, is the fact that the single input $\mathcal{H}_2/SN$ problem and single output $\mathcal{H}_2/SN$ problems have been solved by making use of the separation structure of a modified system.[56,22]

Other researchers[54,55,59] have made use of an observer-based separation structure to design controllers that robustly stabilize uncertain systems. However, these results have not dealt with the question of robust performance. Furthermore, these results were concerned with time-varying uncertainties, so they would be overly conservative for systems with static parametric uncertainties.

1.2 Thesis Overview and Objectives

As previously stated, the overall goal of the thesis is to develop efficient and fast methods to design robust controllers for $\mathcal{H}_2$ performance. Lyapunov stability-based Popov analysis techniques have been chosen to be used in the thesis for performance analysis. This choice is based on several reasons. First, a Popov analysis provides the user with *a priori* guarantees on a closed loop system's $\mathcal{H}_2$ performance, even when the system is perturbed. Secondly, compared with the other analysis techniques in existence, a Popov analysis is relatively non-conservative and does not require an excessive amount of computation. Other techniques can further reduce conservatism in an analysis, but this comes at the expense of speed. Finally, Popov analysis is chosen because it has previously been used for controller designs on structural control experiments with real parametric uncertainties.[50] Thus, the controllers that will be designed can be presumed to have at least as much utility as those used previously.

The need in the control community today is for design tools that are convenient to use on practical problems, rather than just on small example problems. If robust control design algorithms are perceived as being inconvenient to use, designers will avoid them and stick with conventional design tools. Whether a tool is perceived as being convenient is a function of two quantities: the time for the design algorithm to converge and the time required to obtain an initial solution to the problem at hand. The time for the design algorithm to converge is a measure of the raw computational requirements of a design tool. For a particular problem, this depends on the type of algorithm used (e.g. Newton's method vs. a convex optimization routine), the size of the problem (the number of constraints and the number of variables), and the structure of the problem (nonlinear, convex, etc.). The time required to obtain an initial solution is a catch-all phrase that refers to the amount of time required to set up a problem and then to apply the design algorithm so that an initial problem solution can be obtained. Controller design, like any design, is an iterative process. After an initial solution is tested, it is likely that design parameters will need to be adjusted and that a new design must be performed. For robust control problems, the time to obtain a first solution can be dominated by the time required to select an initial condition for the design algorithm, rather than by the time required for the design algorithm to converge.

To test whether or not a design algorithm is convenient for practical structural control problems, we can utilize models of the Middeck Active Control Experiment (MACE). MACE is a flexible structure with embedded actuators and sensors that is designed to simulate a flexible precision pointing spacecraft.[61] In the MACE space shuttle flight experiment, onboard STS-67 in March of 1995, the ability to design controllers quickly was a necessity. One of the goals of the experiment was to demonstrate that by experimentally identifying the structure of the MACE system while it was in space, controllers could be redesigned to improve performance.[60] The reidentification of the structure was necessary to quantify changes that occurred because of the shift from a 1-g to a 0-g environment and because of changes due to the daily disassembly and reassembly of the structure. During the course of the 14 day shuttle experiment, over 400 hundred control designs were tested on the structure.[10] Models of the MACE experimental hardware should serve as excellent test subjects for a design algorithm for several reasons. First, the models are relatively high order. Secondly, the uncertainties in the MACE models are real parametric quantities representing actual uncertain structural parameters. Together these factors combine to make the MACE problem representative of a practical structural control problem. The final reason to use the MACE system is that $\mathcal{H}_2$/Popov controllers have previously been designed for the system by How *et al.*[50,46] Models of the MACE system are discussed in more detail in Appendix B.

To accomplish our objectives, several general tasks must be performed in this thesis:

- Develop a new, systematic way to formulate a robust $\mathcal{H}_2$ controller synthesis problem, given fixed stability multipliers.
- Determine how a Lyapunov-based bound on the $\mathcal{H}_2$ cost of an uncertain system can be separated such that it is the sum of the cost from a robust full-information control problem with the cost of a robust output estimation problem. Determine what this implies about the structure of robust $\mathcal{H}_2$ controllers with regard to model-based compensators.
- Determine if the structure of the robust controllers is useful for synthesis.
- Develop and test new synthesis and analysis algorithms for high-order problems.
- Demonstrate the design algorithm on a practical, high-order, structural control problem such as the MACE system.

The following chapter of this thesis lays out some preliminary mathematics that will be used throughout the thesis. In particular, it contains theorems on Lyapunov-based methods to bound the $\mathcal{H}_2$ performance of an uncertain system. It considers both systems with real parametric as well as gain-bounded uncertainties.

Chapter 4 of this thesis details a new systematic formulation of our robust control synthesis problem. In particular, it demonstrates that full-order, dynamic, $\mathcal{H}_2$ controllers can be synthesized using LMIs. This includes both robust controllers as well as the standard LQG controller. The results of this chapter stand as a viable method to synthesize robust $\mathcal{H}_2$ controllers for small systems.

Afterwards, in Chapter 5, a second synthesis method is developed for the robust $\mathcal{H}_2$ control problem. The method is an alternative to that already derived in Chapter 4. It relies on identifying a useful separation structure in the bound on the cost. The method leads to a set of three equations that are sufficient to guarantee the existence of a robust controller.

The subject of Chapter 6 is how to transform the previously discussed synthesis and analysis theories into practical computational tools. This is a difficult task, because, for a design tool to be truly useful, it must be able to handle relatively high-order systems. The chapter contains discussions of a synthesis algorithm and then an analysis algorithm. Finally, these algorithms are combined to form a design algorithm which is eventually demonstrated on the MACE system.

The final chapter of this thesis contains a summary of the preceding work. It details the contributions this thesis makes to the field of robust control. Finally, it concludes with some remarks about further work that could be done to advance the field.

Chapter 2

Mathematical Preliminaries

This chapter is a tutorial. The first goal of this chapter is to discuss two fundamental concepts that are used throughout the thesis: linear matrix inequalities (LMIs) and dissipation theory. LMIs have become quite popular for use in control system design and analysis recently. They will be employed for robust $\mathcal{H}_2$ controllers in Chapter 4. Dissipation theory is essentially an extension of Lyapunov stability theory. Dissipation theory will be used to derive conditions which are sufficient to guarantee that an uncertain system can maintain certain levels of both stability and performance.

The second goal of this chapter is to highlight several known theorems on stability and performance robustness. These theorems contain state space tests that allow us to guarantee that a linear system will be robustly stable to its set of allowable uncertainties. The last section of the chapter contains theorems that allow us to guarantee a level of $\mathcal{H}_2$ performance for an uncertain system, in addition to guaranteeing its stability. These results can be derived using dissipation theory. The results are posed both in terms of LMIs as well as in terms of Riccati equations.

Most of the notation in this paper is standard, however, we note that $\mathcal{R}_+$ is the set of non-negative real numbers, $\mathcal{R}^{n \times m}$ is the vector space of $n \times m$ real-valued matrices, and $\mathcal{S}^n$ is the set of square, symmetric, real-valued matrices. Furthermore, $D^\dagger$ is the

Moore-Penrose pseudo-inverse of a matrix D. Lastly, the notation

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right],$$

denotes a state space realization for the transfer function $C(sI - A)^{-1}B + D$.

2.1 Introduction to Linear Matrix Inequalities

This section contains a definition of a linear matrix inequality (LMI), discusses some of the properties of LMIs, and highlights state-of-the-art software which can now be used to minimize a linear cost function subject to LMI constraints. The purpose of the section is not to discuss the history or usage of LMIs for control theory. However, for a comprehensive discussion of LMIs, including extensive coverage of their usage in control theory, the reader is referred to Boyd *et al.*[8]

A matrix inequality is a statement that a matrix is sign definite or semidefinite. We write $A > 0$ and $A \geq 0$ to denote the fact that the matrix A is positive definite and positive semidefinite, respectively. Throughout this thesis, any matrix that is written as forming a matrix inequality can be assumed to be symmetric. If a matrix is positive (negative) definite then each of its eigenvalues is a real, nonzero, positive (negative) number. This, of course, implies that the matrix is nonsingular (*i.e.*, it is invertible).

A linear matrix inequality is an algebraic matrix inequality in which some of the parameters in the matrix are variables. These variables must appear in the matrix linearly (not quadratically, not bilinearly, etc.). Typically, whether or not a given matrix will satisfy a matrix inequality depends on the values chosen for the variables in the matrix. If the set of variables that will cause a matrix inequality to be satisfied is non-empty, then the matrix inequality is referred to as *feasible*.

We present a common example of an LMI, the Lyapunov inequality, which happens to come from control theory. If a linear system has dynamics given by $\dot{x} = Ax$, then the system is asymptotically stable if there exists a matrix $P \in \mathcal{S}^n$ such that

$$PA + A^T P < 0 \,.$$

Here, A is a known parameter, and P is the variable (every element of P is a variable). Even though P is used in a matrix multiplication, each element of the P matrix will still appear linearly on the left hand side of the above inequality. Thus, this is an LMI. Note that the matrix that is the left hand side of the expression has explicitly been made symmetric.

One important property of LMIs is that they form convex constraints for their variables. In other words, a convex set is formed by the variables that will cause a given LMI to be satisfied. For example, the above Lyapunov inequality is seen to be convex. Convexity is important to optimization problems that have constraints that can be written in the form of LMIs.

2.1.1 Schur Complements

Often, we shall encounter matrix inequalities with a quadratic variable term. Fortunately, quadratic matrix inequalities in Schur complement form have an equivalent representation as an LMI.[8] In particular, a variable x exists such that

$$\begin{bmatrix} Q(x) & S(x)^T \\ S(x) & R(x) \end{bmatrix} \geq 0 \,, \tag{2.1}$$

if and only if

$$R(x) \geq 0 \,, \; Q(x) - S(x)^T R(x)^\dagger S(x) \geq 0 \,, \text{ and } \; (I - RR^\dagger)R = 0 \,. \tag{2.2}$$

In the case of strict inequalities, the third condition of (2.2) becomes superfluous, and the pseudo-inverse in the second condition is replaced by an inverse.

2.1.2 The Elimination Lemma

We mention an important result that can be used to simplify a specific form of matrix inequality and turn it into a pair of matrix inequalities. This lemma originally appeared in Ref. 67. We present the lemma as it appears in Ref. 33; a proof of the lemma appears in that reference.

Lemma 2.1 (Elimination Lemma) *Consider the matrix inequality*

$$\Psi + U^T \Theta V + V^T \Theta^T U < 0 \,, \tag{2.3}$$

with Ψ symmetric; U, V not necessarily square; and Θ unknown. It can be shown that there exists a matrix Θ satisfying the above matrix inequality iff

$$W_U^T \Psi W_U < 0 \quad \text{and} \quad W_V^T \Psi W_V < 0 \,, \tag{2.4}$$

where N_U and N_V are bases for the null spaces of U and V respectively.

2.1.3 Optimization with LMI constraints

As previously mentioned, a feasible LMI defines a convex set in its variable space. Therefore, minimization of a convex cost function subject to LMI constraints constitutes a convex problem. Of particular interest to our work, the minimization of a *linear* cost function subject to LMI constraints is a convex problem. This is commonly referred to as an eigenvalue problem (EVP).[8]

Convex problems have a number of noteworthy properties. First, the feasible space defined by the constraints is convex. Also, linear cost functions have a unique minimum (maximum) over the feasible space. Certain optimization methods have been developed to take advantage of these properties. In particular, ellipsoid methods and interior point method solvers have been employed to solve EVPs.[8] These methods theoretically guarantee that if the solver is initialized at a point within the feasible region, the optimum can be found. Furthermore, they guarantee that the optimum will be found in polynomial time. In other words, given any number of variables, say

n variables, there is some fixed constant k, such that problem will be solved in time proportional to n^k.

In this thesis, LMI problems are solved using a test version of the commercial software known as LMI Lab.[30,32,31] This software employs the sophisticated interior point method solvers proposed in Ref. 64.

2.2 Introduction to Dissipation Theory

Dissipation theory is an extension of Lyapunov stability theory (Lyapunov's Second or Direct method). The importance of dissipation theory is that it provides a means to very quickly and straightforwardly write down the form of a candidate Lyapunov function for a linear system with uncertainties. The majority of the results in the succeeding sections of this chapter can be derived using dissipation theory.

This material was developed primarily by Willems,[81,79,80] but for a more recent discussion the reader is referred to Refs. 44 or 57. Consider the dynamic system, M, given by

$$\begin{aligned} \dot{x}(t) &= g(x(t), w(t)) \\ z(t) &= h(x(t), w(t)) , \end{aligned} \tag{2.5}$$

where $x : \mathcal{R}_+ \to \mathcal{R}^{n\times 1}$ and $z, w : \mathcal{R}_+ \to \mathcal{R}^{n_w \times 1}$. Definitions are now given for storage function and its corresponding supply rate.

Definition 2.1 (Willems, Refs. 79, 80) *A system M of the form in equation (2.5) with state x, is said to be passive with respect to a scalar function, $r(w, z)$, called a supply rate, if there exists a positive definite function $V_M : \mathcal{R}^n \to \mathcal{R}$, called a storage function, that satisfies the dissipation inequality*

$$V_M(x(t_2)) \leq V_M(x(t_1)) + \int_{t_1}^{t_2} r(w(\tau), z(\tau))\tau \, , \tag{2.6}$$

for all t_1, t_2 and all x, z, and w satisfying the system definition of equation (2.5).

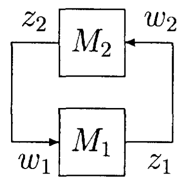

Figure 2-1: Neutral interconnection of two systems M_1 and M_2.

Assuming that V_M is differentiable, an alternate form of (2.6) is

$$\dot{V}_M(x(t)) \leq r(w(t), z(t)) \, , \; t \geq 0 \, , \tag{2.7}$$

where $\dot{V}_M$ denotes the total derivative of $V_M(x)$ along the state trajectory $x(t)$.

We now show how to use dissipation theory to derive Lyapunov functions that prove the stability of a feedback system.

Lemma 2.2 (Willems, Refs. 79, 80) *Consider two dynamic systems M_1 and M_2 with state space representations as in equation (2.5), and input output pairs (w_1, z_1) and (w_2, z_2) respectively. For the system interconnection illustrated in Figure 2-1, with $w_1 = z_2$ and $w_2 = z_1$, assume that the individual systems are associated with states, supply rates and storage functions of the form x_1, $r_1(w_1, z_1)$, $V_{M_1}(x_1) > 0$; and x_2, $r_2(w_2, z_2)$, $V_{M_2}(x_2) > 0$ respectively. Suppose that the supply rates satisfy $r_1(w_1, z_1) + r_2(w_2, z_2) = 0$, for all $w_1 = z_2$ and $w_2 = z_1$. In that case, the equilibrium point $(x_1, x_2) = 0$ of the feedback interconnection of M_1 and M_2 is Lyapunov stable. Furthermore, a Lyapunov function which proves the stability of the system is $V = V_{M_1} + V_{M_2}$.*

Finding a Lyapunov function for an interconnected system is a sufficient condition to guarantee its stability. To extend this to *robust* stability problems, suppose that system M_1 in Fig. 2-1 is a known LTI system and that M_2 represents an uncertainty block, Δ. Typically, the system supply rate is assumed to be a quadratic function of the states. If we are given a supply rate for the uncertainty block, $r_2(t)$, then we need merely to determine under what conditions the system is stable if $r_1(t) = -r_2(t)$. Fortunately, for many types of uncertainty blocks, there are well known supply rates in the literature. For instance, if Δ is an $\mathcal{L}_2 \to \mathcal{L}_2$ norm bounded system such that $\|z\|_2/\|w\|_2 < 1$, then we can choose $r_2(t) = \sigma(w_2^T w_2 - z_2^T z_2)$, where σ is any positive scalar.[44]

The preceding discussion can also be easily extended to cases with multiple independent blocks. For instance, if there are two uncertainty blocks, Δ_1 and Δ_2, then a sufficient condition to guarantee stability of a system, M, is that $r_M = -(r_{\Delta_1} + r_{\Delta_2})$.

2.3 A Review of Robust Stability Analysis

The purpose of this section is to present known state space robustness tests that can be proven using dissipation theory. We will mention equivalent frequency domain criteria for pedagogical and historical reasons, but these are not utilized in this thesis.

The test used to determine if a system is stable for all allowable values of an uncertainty depends on what information is known about the uncertainty. The most general type of uncertainty considered in this thesis is a square, gain-bounded uncertainty block. This is considered first. Tests for systems with real parametric uncertainty are more complex and are considered afterwards.

For simplicity, this discussion of gain-bounded uncertainties will be limited to single blocks, *i.e.*, uncertainties without structure. However, the material will be easily generalizable to block-diagonal uncertainties.

Consider the following LTI system, M, in a feedback loop with an uncertainty block, Δ,

$$\begin{aligned} \dot{x} &= Ax + B_w w \\ z &= C_z x \\ w &= \Delta z\,, \end{aligned} \tag{2.8}$$

where we assume that A is stable and that Δ is a square operator and all of the matrices are of the appropriate sizes. Assume that the only information known about the uncertainty block is that its $\mathcal{L}_2$-induced norm is bounded, in particular that

$$\sup_{z \neq 0} \frac{\|\Delta z\|_2}{\|z\|_2} = \sup_{z \neq 0} \frac{\|w\|_2}{\|z\|_2} < \frac{1}{\gamma}\,. \tag{2.9}$$

This system is essentially the same as the system shown in Fig. 2-1, with, say, M replacing M_1 and Δ replacing M_2. The Small Gain Theorem tells us that the system is robustly stable if we can guarantee that the infinity norm of the plant is less γ. However, to reduce the conservatism of the Small Gain Theorem, we can appeal to the Structured Small Gain Theorem (see Section 7.2 of Ref. 85). First, we define the structured norm, $\mathcal{SN}$.

Definition 2.2 (Dahleh & Diaz-Bobillo, Ref. 18) *The structured norm, $\mathcal{SN}$, is a map from the space of stable systems to the nonnegative reals, defined as*

$$\mathcal{SN} = \mathcal{SN}_{\Delta,2}(M) = \frac{1}{\inf_\Delta \left\{ \|\Delta\|_{\mathcal{L}_2-\text{ind}} \middle| \, (I - M\Delta)^{-1} \text{ is not } \mathcal{L}_2\text{-stable} \right\}}\,, \tag{2.10}$$

if, for every allowable Δ, $(I - M\Delta)^{-1}$ is $\mathcal{L}_2$-stable, then $\mathcal{SN} \triangleq 0$.

Note that $\mathcal{SN}$ is not a true norm in the mathematical sense, because it does not satisfy the triangle inequality and there are nonzero matrices for which $\mathcal{SN}$=0. Clearly, $\mathcal{SN}$ is a generalization of μ that is used for systems with potentially nonlinear and/or time-varying uncertainties. The following Structured Small Gain Theorem is adapted from Ref. 18.

Theorem 2.1 *The system of equation (2.8) is stable in an $\mathcal{L}_2$ sense to all allowable uncertainties if and only if*

$$\mathcal{SN} \leq \gamma\,. \tag{2.11}$$

Unfortunately, as mentioned in the introduction, it is not possible to exactly calculate $\mathcal{SN}$ nor μ in general. However, in the case of nonlinear and/or time-varying perturbations, bounds on $\mathcal{SN}$ become exact. For the single uncertainty block, we define a scaling function, $\mathcal{D} = \sigma I$, where σ is a real number greater than zero. This form is chosen for the scaling such that for all allowable uncertainties

$$\|\mathcal{D}^{-1}\Delta\mathcal{D}\|_{\mathcal{L}_2-\text{ind}} = \|\Delta\|_{\mathcal{L}_2-\text{ind}} ,$$

even though the scaling does not necessarily commute with the Δ block. The following theorem tells us how to compute *SN*.

Theorem 2.2 (Dahleh & Diaz-Bobillo, Thm. 7.4.2, Ref. 18) *For nonlinear, time-varying or time-invariant perturbations,*

$$SN = \inf_{\sigma>0} \|\mathcal{D}^{-1}\hat{M}\mathcal{D}\|_{\infty} . \tag{2.12}$$

Usage of the Structured Small Gain Theorem has two distinct advantages over the Small Gain Theorem. First, the Structured Small Gain Theorem allows us to consider uncertainties with multiple blocks, *i.e.*,

$$\Delta \in \Big\{\text{diag }(\Delta_1, \Delta_2, ...\Delta_l) | \, \Delta_i \text{ is a } p_i \times p_i \text{ causal, nonlinear, time-varying operator; and } \sum_{i=1}^{l} p_i = n_w \Big\} .$$

In this case, the scaling function would also become block diagonal. Secondly, conservatism is reduced by use of the scaling functions, $\mathcal{D}$, which are also easily generalized for multiple blocks. In the case of linear uncertainties, the theory presented for *SN* is equivalent to complex μ theory. We will not utilize μ nor *SN* machinery directly in this thesis. They have been presented so that dissipation-based results can be compared to them.

For pedagogical reasons, let us now apply dissipation theory to the system of equation (2.8) and derive a sufficient condition for robust stability. We will derive a condition that ensures that $\|\mathcal{D}\hat{M}\mathcal{D}\|_{\infty} \leq \gamma$. A supply rate for the uncertainty block can be chosen to be

$$r_{\Delta} = \sigma(z^T z - \gamma^2 w^T w) , \tag{2.13}$$

where σ is, again, a real scalar greater than zero. We will assume that the system has a storage function that is a quadratic function of the states. Given some positive definite matrix, P, say that the system storage function is

$$V_M = x^T P x \,. \tag{2.14}$$

We wish to show that the system is dissipative with respect to its supply rate, r_M, *i.e.*, $\dot{V}_M \leq r_M$. To guarantee that the system is stable, according to Lemma 2.2, we need that $r_M = -r_\Delta$, which implies that we need

$$\dot{V}_M \leq -\sigma(z^T z - \gamma^2 w^T w) \,, \tag{2.15}$$

where the matrix P in the definition of V is, as of yet, unknown. Equation (2.15) is a sufficient condition to guarantee that the system is robustly stable. If we evaluate the derivative of the storage function using the plant dynamics, we quickly find that this condition is equivalent to

$$x^T(PA + A^T P + \sigma C_z^T C_z)x + x^T P B_w w + w^T B_w^T P x - \sigma\gamma^2 w^T w \leq 0 \,. \tag{2.16}$$

We need this condition to hold for all x and w. This implies, then, that we need the following matrix inequality to hold

$$\begin{bmatrix} PA + A^T P + \sigma C_z^T C_z & PB_w \\ B_w^T P & -\sigma\gamma^2 I \end{bmatrix} \leq 0 \,. \tag{2.17}$$

Using Schur complements then, this matrix inequality is equivalent to the following inequality (only the non-trivial condition is shown)

$$PA + A^T P + \sigma C_z^T C_z + \frac{1}{\sigma\gamma^2} P B_w B_w^T P \leq 0 \,, \tag{2.18}$$

which is an Algebraic Riccati inequality. If there exists a positive definite matrix P that satisfies the above inequality, then the system is robustly stable. Of course, these inequalities are very familiar. They are statements of the Bounded Real Lemma, which tells us that they are equivalent to

$$\|\mathcal{D}^{-1}\hat{M}(s)\mathcal{D}\|_\infty = \|\hat{M}(s)\|_\infty = \|C_z(sI - A)^{-1}B_w\|_\infty \leq \gamma\,, \tag{2.19}$$

which agrees with the desired *SN* condition. Furthermore, inequality (2.18) is familiar because it is the inequality form of the standard $\mathcal{H}_\infty$ Riccati equation. It can be shown[78,81,23] that an equivalent method to guarantee the stability of this system is to solve the familiar $\mathcal{H}_\infty$ Riccati equation

$$PA + A^TP + \sigma C_z^TC_z + \frac{1}{\sigma\gamma^2}PB_wB_w^TP = 0\,. \tag{2.20}$$

If a unique, stabilizing, positive-definite solution to the Riccati equation exists, it possesses the smallest eigenvalues of all positive definite matrices satisfying inequalities (2.17) or (2.18).

In the frequency domain, these conditions also have a familiar interpretation, commonly known as the "Circle Criterion." If the plant is SISO, then the Circle Criterion requires that the Nyquist plot of the system remains inside a circle of radius γ.

We now consider a system with independent, static, real, parametric uncertainties. This type of uncertainty is complicated to describe because it does not change with time and it has constant (zero) phase. In fact, to definitively show whether a system with real parametric uncertainties is robustly stable is an NP-hard problem.[9,83] Robust stability analysis of such a system can be analyzed using the mixed or real μ type formulation of Fan *et al.*[24] However, because we are interested in controller design, we examine Lyapunov-based, state space methods of analysis. If a gain-bounded uncertainty formulation is used, then the results will be overly conservative. To make the results more accurate, we can make the Lyapunov function depend on the uncertain parameters. In particular, we will present methods of analysis that use Popov multipliers[41,42,46] and then discuss the connections between this and μ.

Consider again the stable system given in equation (2.8), where Δ is a size $n_w \times n_w$, unknown, diagonal, static, real matrix. The Δ block is diagonal because the uncertainties are independent. Assume that, in addition to being real, it is known

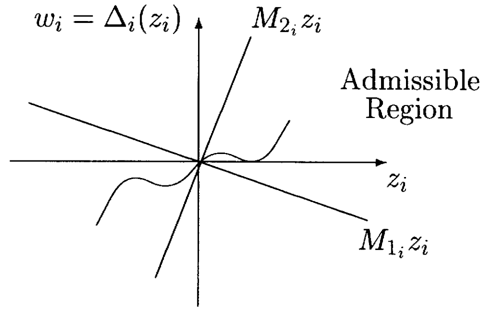

Figure 2-2: A sector-bounded nonlinearity.

that each of the uncertainties, δ_i, will lie within a certain range. In particular, say that we know $2n_w$ real scalar parameters, M_{1_i}, M_{2_i}, such that $M_{1_i} \leq \delta_i \leq M_{2_i}$ for $i \in [1, n_w]$. A Popov analysis constrains the phase of each uncertainty by considering it to be a static, sector-bounded nonlinearity. Defining M_1 and M_2 to be diag(M_{1_i}) and diag(M_{2_i}), respectively, and

$$M_d = (M_2 - M_1)^{-1}\,, \tag{2.21}$$

the Popov analysis restricts the analysis to consider only uncertainty blocks that are elements of the following set

$$\mathcal{U} \triangleq \{\Delta(z)|\ (\Delta(z) - M_1 z)^T M_D (M_2 z - \Delta(z)) \geq 0\,,\ \text{and}\ \Delta(0) = 0\Big\}\ . \tag{2.22}$$

For each of the individual uncertainties, this restricts it to belong to the sector pictured in Fig. 2-2. At this point, the formulation is very similar to the Lur'e problem (see, for instance, Ref. 8).

The following theorem presents a sufficient condition to guarantee that the system is robustly stable to a real parametric uncertainty.

Theorem 2.3 (Livadas, Ref. 57) *Given a system as defined in equation (2.8) and the uncertainty set, $\mathcal{U}$, defined in equation (2.22), if there exists a positive definite matrix P together with diagonal matrices H and N, where P and H are positive* *definite and N is positive semidefinite, and the following matrix inequality is satisfied,*

$$\begin{bmatrix} \begin{array}{c} A^T\left(P - C_z^T N M_1 C_z\right) + \left(P - C_z^T M_1 N C_z\right) A \\ -C_z^T M_1 M_d H M_2 C_z - C_z^T M_2 H M_d M_1 C_z \end{array} & (\cdots)^T \\ \begin{array}{c} B_w^T\left(P - C_z N M_1 C_z\right) + N C_z A \\ +M_d H M_2 C_z + H M_d M_1 C_z \end{array} & \begin{array}{c} B_w^T C_z^T N + N C_z B_w \\ -H M_d - M_d H \end{array} \end{bmatrix} \leq 0\,, \quad (2.23)$$

then the system is stable for all $\Delta \in \mathcal{U}$.

The reader is referred to Ref. 57 for a proof of this theorem. The derivation of the constraint can be done using dissipation theory and an uncertainty supply rate of

$$r_\Delta = 2(w - M_1 z)^T N \dot{z} + 2(w - M_1 z)^T M_d H (M_2 z - w)\,. \qquad (2.24)$$

The matrices H and N are referred to as the "stability multipliers" of the problem. One could also derive a similar condition that allows more general multipliers. If the restriction that N be positive semi-definite is lifted, then the stability multipliers become non-causal and conservatism is reduced.[26]

As was found for the case with gain-bounded uncertainties, this stability robustness criterion can also be written in terms of a Riccati equation. The following theorem can be derived using the Schur complements of the constraint from Theorem 2.3.

Theorem 2.4 (How, Ref.46) *Given a system as defined in equation (2.8) and an uncertainty set $\mathcal{U}$, as defined in equation (2.22), suppose a set of diagonal matrices H and N exist such that H is positive definite, N is positive semidefinite, and*

$$R_0 \triangleq H M_d + M_d H - B_w^T C_z^T N - N C_z B_w \qquad (2.25)$$

is positive definite. Furthermore, suppose there exists a matrix $P \geq 0$ satisfying

$$(A + B_w M_1 C_z)^T P + P(A + B_w M_1 C_z) + [H C_z + B_w^T P + N C_z (A + B_w M_1 C_z)]^T \cdot$$
$$R_0^{-1}[H C_z + B_w^T P + N C_z (A + B_w M_1 C_z)] = 0\,. \qquad (2.26)$$

Then, the system is robustly stable for all $\Delta \in \mathcal{U}$.

This theorem is similar to results derived in Ref. 41. Note that Theorems 2.3 and 2.4 are effectively the same.

We can relate the conditions in Theorems 2.3 and 2.4 to a mixed or real μ analysis and a modified circle criterion. Using the results of Refs. 81,46, one can demonstrate that the test that a system be dissipative is equivalent to a test that the transfer function of a related system is positive real at all frequencies. In this manner, one can transform the dissipation tests from a state space condition to the frequency domain. Consider a case with an even-sided sector-bounded uncertainty, *i.e.*, with $M_2 = -M_1 = \beta I$. In Ref. 49, it is shown that checking the conditions of Theorem 2.4 is equivalent to checking if the transfer function of the nominal plant, $G(j\omega)$, satisfies the following condition at all frequencies

$$G^* H G + j(\bar{N} G - G^* \bar{N}) - \alpha^2 H \leq 0 \tag{2.27}$$

where H is the usual stability multiplier, $\bar{N} = 2N\omega/\beta$, and $\alpha = \beta^{-1}$. We can compare this to the mixed or real μ test posed by Fan *et al.*[24] At any given frequency, μ is overbounded by a parameter $\bar{\mu}$, if the maximum eigenvalue of the expression

$$G^* \mathcal{D} G + j(\mathcal{N} G - G^* \mathcal{N}) - \bar{\mu} H \tag{2.28}$$

is non-negative, and where the scaling matrices, $\mathcal{D} = \mathcal{D}(\omega)$ and $\mathcal{N} = \mathcal{N}(\omega)$, are chosen to minimize the eigenvalue. (The $\mathcal{D}$ scaling has the same purpose as the scaling matrix used to calculate *SN* in equation (2.12).) Clearly, the Popov and real μ tests have the same form. In fact, it is clear that the Popov multipliers, H and N, are equivalent to specific realizations of the scaling matrices in μ theory

$$\mathcal{D}(\omega) = \mathcal{D} = H \tag{2.29}$$

$$\mathcal{N}(\omega) = \mathcal{N} = 2N\omega/\beta\,. \tag{2.30}$$

The Popov analysis can also be interpreted as an off-axis circle criterion. Consider a SISO system, where the plant transfer function, $G(s)$, is written as $x + jy$. In this case, Ref. 49 shows that condition (2.27) can be written as

$$x^2 + \left(y - \frac{\bar{N}}{H}\right)^2 \leq \alpha^2 + \left(\frac{\bar{N}}{H}\right)^2 . \tag{2.31}$$

This condition should be interpreted as requiring that the Nyquist plot of the system lie within a circle centered on the $j\omega$ axis at $\bar{N}/H$. Thus, the Popov test can be interpreted as a sophisticated extension of the small gain circle criterion. Furthermore, it is also noticed that, just as mixed or real μ is a generalization of complex μ, the Popov analysis is a generalization of the analysis technique used for a system with gain-bounded uncertainties. In terms of the Popov stability multipliers, the multipliers used for the gain-bounded case are $H = \sigma I$, and $N = 0$.

2.4 A Review of Robust Performance Analysis

This section has two purposes. The first purpose is to define the $\mathcal{H}_2$ performance metric that will be used throughout the thesis. The second is to present results that extend the stability robustness concepts of the previous section to allow us to bound the performance of an uncertain system.

First, consider a system with real parametric uncertainties in the dynamics matrix, defined by

$$\begin{aligned} \dot{x} &= (A + \Delta A)x + B_d d \\ e &= C_e x , \end{aligned} \tag{2.32}$$

where $A \in \mathcal{R}^{n \times n}$ is the nominal dynamics matrix, d is an exogenous disturbance, e is the performance signal, and the parametric uncertainty, ΔA, is defined as follows. Recall the definition of $\mathcal{U}$ in equation (2.22). Using this definition, we can define ΔA as some element of the set $\mathcal{U}_A$ where

$$\mathcal{U}_A \triangleq \left\{ \Delta A \in \mathcal{R}^{n \times n} \mid \Delta A = B_w \Delta C_z, \text{ where } \Delta \in \mathcal{U} \right\} , \tag{2.33}$$

and where $B_w \in \mathcal{R}^{n \times n_w}$ and $C_z \in \mathcal{R}^{n_w \times n}$ are known matrices. For convenience, the system can also be written as

$$\begin{aligned} \dot{x} &= Ax + B_w w + B_d d \\ z &= C_z x \\ e &= C_e x \\ z &= \Delta w \, , \end{aligned} \tag{2.34}$$

where Δ is a member of $\mathcal{U}$.

We wish to analyze the $\mathcal{H}_2$ performance of our system. The difficulty with this uncertainty description is that it has described the parametric uncertainty in terms of uncertainty blocks, Δ, that are nonlinear. Indeed, in the case of gain-bounded uncertainties, we explicitly wish to consider uncertainty blocks that are nonlinear. Unfortunately, the $\mathcal{H}_2$ norm of a nonlinear system may not be defined. Therefore, we will define a generalized $\mathcal{H}_2$ performance metric, that, in the case of a linear perturbation, is equal to the square of the worst-case $\mathcal{H}_2$ norm of the system given all possible system perturbations. As an aside, it is important to note that, unlike the generalized $\mathcal{H}_2$ metric of Rotea,[71] our performance metric reduces to the usual $\mathcal{H}_2$ norm as long as the system is linear, even if the system is MIMO. We initially give two definitions for our performance metric. If d is taken to be a zero-mean, unit-covariance, white noise signal, then the performance metric, J, is the expected value of the Euclidean norm of e, or

$$J(\mathcal{U}_A) = \sup_{\Delta A \in \mathcal{U}_A} \lim_{\tau \to \infty} \frac{1}{\tau} \int_0^{\tau} e^T e \, dt \, . \tag{2.35}$$

A second definition for the performance metric depends on the impulse response of the system. For an LTI system, these definitions are equivalent. This definition is taken to be the sum of the squares of the $\mathcal{L}_2$ norms of the responses due to a set of impulsive inputs, δd_i, where the d_i form a basis for the input space, or

$$J(\mathcal{U}_A) = \sup_{\Delta A \in \mathcal{U}_A} \sum_{i=1}^{n_d} \|e_i\|_2^2 \tag{2.36}$$

$$= \sup_{\Delta A \in \mathcal{U}_A} \sum_{i=1}^{n_d} \int_0^\infty e_i^T e_i \, dt \,. \tag{2.37}$$

Either definition of the cost is generalizable to nonlinear or slowly time-varying systems. However, in these cases, equation (2.35) and equation (2.37) are not necessarily equal. In the cases of interest to this thesis, we typically will not be able to calculate these cost functions exactly. Instead we will calculate bounds on the cost functions. We postulate that, when applicable, a bound on equation (2.37) will also provide an an upper bound to equation (2.35). Therefore, equations (2.35) and (2.37) will be used interchangeably to define the cost function. Stoorvogel[74] provides an excellent discussion of the meaning of the $\mathcal{H}_2$ norm for a nonlinear system and gives a comprehensive discussion of this performance metric.

It remains to be seen how the performance metric can be calculated. The performance metric is equivalent to the function defined in the following lemma.

Lemma 2.3 (Haddad and Bernstein, Ref. 41) *Given the system of equation (2.32), with $\mathcal{U}_A$ as defined in equation (2.33), assume $A + \Delta A$ is stable for all allowable ΔA. In this case, the performance metric function is equivalent to*

$$J(\mathcal{U}_A) \triangleq \sup_{\Delta A \in \mathcal{U}_A} \operatorname{tr} B_d^T P_{\Delta A} B_d \tag{2.38}$$

where $P_{\Delta A}$ is the unique, positive semidefinite solution to the Lyapunov equation

$$(A + \Delta A)^T P_{\Delta A} + P_{\Delta A}(A + \Delta A) + C_e^T C_e = 0 \,. \tag{2.39}$$

This lemma is based directly on the manner in which the $\mathcal{H}_2$ cost is typically calculated for a linear system. The definition seems straightforward. Unfortunately, in general it is not possible to calculate $J(\mathcal{U}_A)$. This would require a search over all possible ΔA; it is at least as difficult as solving the μ stability robustness problem, which is NP-hard. Instead, given specifications about the class of uncertainties, we must utilize methods that overbound our performance metric, $J(\mathcal{U}_A)$.

The first of these bounds comes from Refs. 82,57. It is an extension of Theorem 2.3.

Theorem 2.5 (Livadas, Ref. 57) *Given a system as defined in equation (2.32) and an uncertainty set, $\mathcal{U}_A$, defined in equation (2.33), if there exists a positive definite matrix P together with diagonal matrices H and N, where P and H are positive definite and N is positive semidefinite, and the following matrix inequality is satisfied,*

$$\begin{bmatrix} \begin{array}{c} A^T\left(P - C_z^T N M_1 C_z\right) + \left(P - C_z^T M_1 N C_z\right) A \\ -C_z^T M_1 M_d H M_2 C_z - C_z^T M_2 H M_d M_1 C_z \\ +C_e^T C_e \end{array} & (\cdots)^T \\ \begin{array}{c} B_w^T \left(P - C_z N M_1 C_z\right) + N C_z A \\ +M_d H M_2 C_z + H M_d M_1 C_z \end{array} & \begin{array}{c} B_w^T C_z^T N + N C_z B_w \\ -H M_d - M_d H \end{array} \end{bmatrix} \leq 0\,, \quad (2.40)$$

then the system is stable for all $\Delta \in \mathcal{U}$ and the performance metric, $J(\mathcal{U}_A)$, is bounded from above by the following cost function,

$$J(\mathcal{U}_A) \leq \mathcal{J} = \text{tr}\,\left[B_d^T\left(P + C_z^T\left(M_2 - M_1\right) N C_z\right) B_d\right]\,. \quad (2.41)$$

The reader is referred to Ref. 82 for a proof of this theorem and, in particular, a derivation of the cost function. The authors use the storage functions of dissipation theory to capture the impulsive nature of the performance metric.

As was found in the case of the stability robustness tests in the previous section, the LMI-based performance test has an analog in a test that uses a Riccati equation. This is presented in the following theorem.

Theorem 2.6 (How, Ref. 46) *Given a system as defined in equation (2.32) and an uncertainty set, $\mathcal{U}_A$, defined in equation (2.33), suppose a set of diagonal matrices H and N, exist such that H is positive definite and N is positive semidefinite, and that R_0, defined in equation (2.25), is positive definite. Furthermore, suppose there exists a matrix $P \geq 0$ satisfying*

$$(A + B_w M_1 C_z)^T P + P(A + B_w M_1 C_z) + [H C_z + B_w^T P + N C_z (A + B_w M_1 C_z)]^T \cdot R_0^{-1}[H C_z + B_w^T P + N C_z (A + B_w M_1 C_z)] + C_e^T C_e = 0\,. \quad (2.42)$$

Then, the system is robustly stable for all $\Delta \in \mathcal{U}$. Furthermore, the cost, $J(\mathcal{U}_A)$ is bounded from above by the following cost function,

$$J(\mathcal{U}_A) \leq \mathcal{J} = \mathrm{tr}\ \left[B_d^T \left(P + C_z^T (M_2 - M_1) N C_z\right) B_d\right] .$$

This gives us two methods two bound our performance metric. Note that, for any feasible H and N, if the bound given in Theorem 2.5 is minimized with respect to P, then it is equal to the bound given in Theorem 2.6. Furthermore, these bounds could be rewritten in terms of a stability multiplier, N, that was not restricted to be positive semidefinite in order to reduce conservatism.

We now consider a system with a gain-bounded uncertainty block or, equivalently, a system subject to a $\mathcal{L}_2$ norm bounded exogenous disturbance. Recall the system defined in equation (2.34). In this case, the only restriction on Δ is that its $\mathcal{L}_2$-induced norm is bounded by $1/\gamma$. The previously given definition for the generalized $\mathcal{H}_2$ performance metric applies to this uncertainty set as well.

When we were only concerned with stability robustness, rather than performance robustness, the analysis of this uncertainty set led to $\mathcal{H}_\infty$ type conditions. The following theorems extend the $\mathcal{H}_\infty$ results to include bounds on the $\mathcal{H}_2$ performance metric. Note that, because a Popov analysis is a generalization of the analysis used for these cases, these theorems can be derived by setting $H = \sigma I$ and $N = 0$ in Theorems 2.5 and 2.6. However, the results are important in their own right. We first present a theorem that uses LMIs and then present its Riccati equation counterpart.

Theorem 2.7 (Ref. 82) *Given a system as defined in equation (2.34), if there exists a scalar parameter $\sigma > 0$, such that there exists a positive definite matrix $P \in \mathcal{S}^n$ satisfying*

$$\begin{bmatrix} A^T P + PA + C_e^T C_e + \sigma^2 C_z^T C_z & \frac{1}{\sigma\gamma} P B_w \\ \frac{1}{\sigma\gamma} B_w^T P & -I \end{bmatrix} \leq 0 , \tag{2.43}$$

then $\|G_{zw}\|_\infty < \gamma$, and the cost, J, is bounded from above by a function

$$J \leq \mathcal{J} = \mathrm{tr}\ \left(B_d^T P B_d\right) . \tag{2.44}$$

Theorem 2.8 (Hall and How, Ref. 44) *Given a system as defined in equation (2.34), if there exists a scalar parameter* $\sigma > 0$*, such that there exists a positive definite matrix* $P \in \mathcal{S}^n$ *satisfying*

$$PA + A^T P + C_e^T C_e + \sigma C_z^T C_z + \frac{1}{\sigma\gamma^2} P B_w B_w^T P = 0 \,, \tag{2.45}$$

then $\|G_{zw}\|_\infty < \gamma$*, and the cost,* J*, is bounded from above by a function*

$$J \leq \mathcal{J} = \operatorname{tr}\left(B_d^T P B_d\right) \,. \tag{2.46}$$

As in the real parametric case, for any feasible σ, if the bound given by equation (2.44) is minimized with respect to P, then the minimal value is equal to the bound given in equation (2.46).

Chapter 3

Problem Statement

This chapter describes the robust $\mathcal{H}_2$ controller design problem in detail. The following chapters in this thesis are devoted to describing new ways of solving this problem. Briefly, the problem is to design a linear, time-invariant, dynamic controller for a linear, finite-dimensional plant that can guarantee that the closed loop system is stable and that the system will deliver, at worst, a known level of $\mathcal{H}_2$ performance, even if the plant is not known exactly. The problem of interest is similar to the problems investigated by previous researchers, as given in Chapter 2. Our problem, is effectively the same as that investigated by How.[46] In this chapter, the problem will be defined, and notation that will be used throughout the remainder of the thesis will be specified.

First, consider a linear, time-invariant plant with no uncertainty, given by

$$\begin{Bmatrix} \dot{x} \\ e \\ y \end{Bmatrix} = \begin{bmatrix} A & B_d & B_u \\ C_e & 0 & D_{eu} \\ C_y & D_{yd} & 0 \end{bmatrix} \begin{Bmatrix} x \\ d \\ u \end{Bmatrix}, \tag{3.1}$$

where $A \in \mathcal{R}^{n \times n}$, $B_u \in \mathcal{R}^{n \times n_u}$, $B_d \in \mathcal{R}^{n \times n_d}$, $C_y \in \mathcal{R}^{n_y \times n}$, $C_e \in \mathcal{R}^{n_e \times n}$, $D_{yd} \in \mathcal{R}^{n_y \times n_d}$, and $D_{eu} \in \mathcal{R}^{n_e \times n_u}$. This system can be considered the nominal system description of the uncertain plant that we want to model for the full problem description. We assume that the disturbance, d, is white noise with zero mean and unit covariance.

Note that the system process noise and any measurement noise are both captured by this disturbance, d. Correlation between the process and measurement noises can be modelled by setting $B_d D_{yd}^T \neq 0$. Also, the control signal is u and the measured signal is y. We assume that (A, B_u) is stabilizable and that (A, C_y) is detectable. The performance of the system is measured at e. We assume that both C_e and D_{eu} are full column rank. Finally, the cost due to the states and the cost due to the control are assumed to be uncoupled, *i.e.*, $C_e^T D_{eu} = 0$, though this assumption can be relaxed.

Our goal is to find the LTI controller, given by

$$\left\{ \begin{array}{c} \dot{x}_c \\ u \end{array} \right\} = \left[\begin{array}{cc} A_c & B_c \\ C_c & 0 \end{array} \right] \left\{ \begin{array}{c} x_c \\ y \end{array} \right\}, \tag{3.2}$$

that minimizes the $\mathcal{H}_2$ norm of the system, as measured at output error signal, e. We assume *a priori* that the controller is full-order, thus $A_c \in \mathcal{R}^{n \times n}$, and the other matrices are of appropriate dimensions. The controller has no feedthrough term from y to u because this would make the resulting $\mathcal{H}_2$ norm of the system be infinite. For the nominal system, the controller that minimizes the $\mathcal{H}_2$ norm is given by the celebrated LQG controller, commonly written as

$$\begin{aligned} A_c &= A - B_u (D_{eu}^T D_{eu})^{-1} B_u^T R - (QC_y^T + B_d D_{yd}^T)(D_{yd} D_{yd}^T)^{-1} C_y && (3.3) \\ B_c &= (QC_y^T + B_d D_{yd}^T)(D_{yd} D_{yd}^T)^{-1} && (3.4) \\ C_c &= -(D_{eu}^T D_{eu})^{-1} B_u^T R\,, && (3.5) \end{aligned}$$

where R and Q are the unique, stabilizing solutions to the Riccati equations

$$RA + A^T R - RB_u (D_{eu}^T D_{eu})^{-1} B_u^T R + C_e^T C_e = 0 \tag{3.6}$$

$$AQ + QA^T + B_d^T B_d - (QC_y^T + B_d D_{yd}^T)(D_{yd} D_{yd}^T)^{-1}(C_y Q + D_{yd} B_d^T) = 0\,. \tag{3.7}$$

Furthermore, the square of the $\mathcal{H}_2$ cost of the closed loop system with the LQG controller is

$$J = \operatorname{tr}\left(RB_d B_d^T\right) + \operatorname{tr}\left(QC_e^T C_e\right)\,. \tag{3.8}$$

Of course, our interest lies with uncertain systems. In this thesis, we consider two types of uncertainties in our system, real parametric uncertainties and gain-bounded uncertainties. The uncertain systems will be perturbed versions of this nominal system.

3.1 Real Parametric Uncertainties: The $\mathcal{H}_2$ /Popov Problem

We examine the problem of designing robust $\mathcal{H}_2$ controllers for a linear, time-invariant system with certain parameters that are unknown. The unknown parameters are real-valued and static. Furthermore, they are known to lie within a certain prespecified range.

Consider a perturbed version of the system discussed in equation (3.1), where the actual system dynamics are captured by a matrix A_δ, rather than the A, but all the input/output matrices (C_y, B_u, etc.) are as given. We can then define a matrix, δA, to be the difference between the actual and the nominal dynamics such that

$$A_\delta = A + \delta A \,. \tag{3.9}$$

Given that the system dynamics matrix is real-valued, δA can be considered real-valued as well. Furthermore, δA is static, since A_δ does not vary with time, and the elements of δA are assumed to be independent of each other. Note that, in general, we do not put any restrictions on the structure of the δA matrix. For a well-known system, δA should be sparsely populated or should be low rank, due to the structure in the system. In this thesis, we shall only consider systems that have an uncertainty in their system dynamics matrix. This restriction is not as limiting as it might seem. In most cases, parametric uncertainties in the "B" and "C" matrices can be incorporated to a modified system dynamics matrix.[46]

We presume that we can bound the magnitude of the variation in any parameter that is expected to be different from its nominal value. Therefore, we assume that we can bound the magnitude of the elements in the δA matrix. To apply the standard tools developed for the robust analysis of linear systems with uncertainties, we place the uncertain parameter in a feedback loop with the plant. This allows us to effectively control an entire set of possible plants. For n_w uncertain parameters, the uncertainty feedback loop is defined as

$$\delta A = B_w \Delta C_z \,, \tag{3.10}$$

and

$$\begin{aligned} w &= \Delta z & (3.11)\\ z &= C_z x \,, & (3.12) \end{aligned}$$

where $\Delta \in \mathcal{R}^{n_w \times n_w}$, $B_w \in \mathcal{R}^{n \times n_w}$, and $C_z \in \mathcal{R}^{n_w \times n}$.

In many cases, elements of the A matrix may themselves be modelled as the uncertain parameters. In other cases, primitive quantities that form the A matrix may be more naturally modelled as the uncertain parameters. In the former case, if a given element, say a_{ij}, of the A matrix is uncertain, then B_w is a column vector of zeros with row i set to unity, and C_z is a row vector of zeros with the j entry set to unity. The case with matrix primitives being more easily modelled as uncertain can appear, for instance, in structural control problems. For instance, the dynamics matrix of a single spring system with a unit mass can be written as

$$A = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix} , \tag{3.13}$$

where ω is the natural frequency and ζ is the damping ratio of the spring. If ζ is an uncertain parameter in the A matrix, then this situation is easily modelled using the framework discussed. We can choose, for instance,

$$B_w = \begin{bmatrix} 0 \\ -2\omega \end{bmatrix} , \text{ and } \; C_z = \begin{bmatrix} 0 & 1 \end{bmatrix} .$$

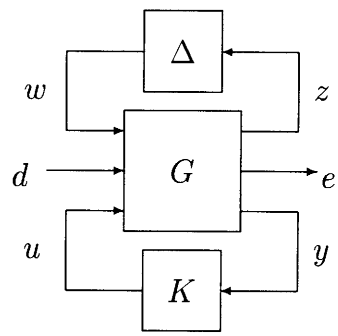

Figure 3-1: The controller design problem

However, in the case where ω is uncertain, we find that the Popov framework cannot handle this exactly. Only one parameter is uncertain, but to place uncertain parameters in both entries of the bottom row of the A matrix would require a diagonal Δ block of size 2×2. Unfortunately, we are unable to constrain both of the elements of an uncertainty block to be equal. However, letting both elements vary independently is overly conservative. In this case, acceptable results have been obtained by instead assuming that the frequency variations occur only in the squared frequency term.[39,10,38] This leads to the following formulation

$$B_w \Delta C_z = \begin{bmatrix} 0 \\ -\omega \end{bmatrix} \delta \begin{bmatrix} \omega & 0 \end{bmatrix} . \tag{3.14}$$

The case with both ω and ζ both uncertain presents further difficulties. The reader is referred to Refs. 46, 38 for further details on the modelling and the selection of the uncertainty matrices. It should be noted that if the formulation allowed the stability multipliers to be block diagonal, rather than diagonal, then this framework would be able to fully accommodate an uncertainty block with a repeated term.

In any case, for real parametric uncertainties, we require that the Δ block is square and diagonal, and that its terms are independent of one another.

We also know that the unknown parameters in the Δ block are real. Therefore, we model the uncertainty block using Popov multipliers, as discussed in Chapter 2. A Popov analysis of real uncertainties is typically less conservative than a small-gain type analysis, because Popov multipliers can bring phase information into the description of the uncertainty block.

It is also assumed that bounds on the expected variation of the parameters are known. Each uncertain parameter, δ_i, is considered to lie between the real values M_{1_i} and M_{2_i}, where $M_{2_i} > M_{1_i}$. These parameters give us the slopes of the sector bound used in the Popov analysis. For n_w uncertain parameters, we can define diagonal matrices M_1 and M_2 of size $n_w \times n_w$ whose nonzero elements are the bounds on the individual uncertain parameters.

We can now concisely summarize the above discussion and state the problem below.

The $\mathcal{H}_2$/Popov design problem: Consider the system in Figure 3-1. The LTI plant, G, in the system is given by

$$\left\{\begin{array}{c} \dot{x} \\ e \\ z \\ y \end{array}\right\} = \left[\begin{array}{c|ccc} A & B_d & B_w & B_u \\ \hline C_e & 0 & 0 & D_{eu} \\ C_z & 0 & 0 & 0 \\ C_y & D_{yd} & 0 & 0 \end{array}\right] \left\{\begin{array}{c} x \\ d \\ w \\ u \end{array}\right\}, \tag{3.15}$$

with $A \in \mathcal{R}^{n \times n}$, $D_{eu} \in \mathcal{R}^{(n+n_u) \times n_u}$, $D_{yd} \in \mathcal{R}^{n_y \times n_d}$, $B_w \in \mathcal{R}^{n \times n_w}$, and $C_z \in \mathcal{R}^{n_w \times n}$, and the other matrices of compatible dimensions. The realization is assumed to be minimal. Furthermore, C_e and D_{eu} are both full column rank, and the control cost is decoupled from the state cost, so that $C_e^T D_{eu} = 0$. Also, we assume that (A, B_u) is stabilizable and that (A, C_y) is detectable. However, we do not put any requirements on the stabilizability and detectability of the uncertainty loop, (A, B_w) and (A, C_z). Lastly, we assume that the estimation problem is non-singular, implying that D_{yd} is full row rank.

The uncertainty block, Δ, is a diagonal matrix whose diagonal entries are the n_w static, independent, real, uncertain parameters. Each of the uncertain parameters, δ_i, is bounded by a known quantity so that we can define diagonal matrices M_1 and M_2 whose diagonal elements are such that $M_{1_i} \leq \delta_i \leq M_{1_2}$. For convenience, we define $M_d \triangleq (M_2 - M_1)^{-1}$.

The goal is to find two size n_w diagonal matrices H and N, with $H > 0$ and $N \geq 0$, which serve as Popov multipliers for the uncertainty block, as well as a full-order LTI controller, given by equation (3.2). The controllers and Popov multipliers should guarantee that the system is stable for any allowable uncertainty block. Moreover, they should provide an overbound on the $\mathcal{H}_2$ cost of the system given any allowable uncertainty.

Note that the constraint that N be positive semidefinite is not necessary. It can be lifted, and the resulting stability multiplier makes the analysis less conservative. However, this constraint is included to allow these designs to be compared to the results in Refs. 46, 49, 50, 57.

3.2 Gain-Bounded Uncertainties: The $\mathcal{H}_2/SN$ Problem

This problem is similar to the $\mathcal{H}_2$/Popov problem. We are still interested in the $\mathcal{H}_2$ performance of a linear, time-invariant plant with a full order LTI controller. However, rather than investigating real parametric uncertainties, we will be interested in robustness against nonlinear, time-varying uncertainties and/or $\mathcal{L}_2$ gain bounded exogenous disturbances. Thus, the system will be made robust to a more general set of uncertainties than was investigated for the $\mathcal{H}_2$/Popov controller. As before, we will use dissipation analysis to analyze the performance of the new systems. This effectively makes the $\mathcal{H}_2/SN$ problem a subset of the $\mathcal{H}_2$/Popov problem, because less complex stability multipliers must be used to describe a more general uncertainty block.

We would like to describe the uncertainty set for the $\mathcal{H}_2/SN$ problem. We will first consider that the uncertainty comes from a parametric uncertainty in the system dynamics matrix. Recall the parameterization of the uncertainty in the A matrix described by equations (3.9), (3.10), and (3.15). We adopt identical parameterization and identical notation for the new case. However, we now allow that the Δ block can be a nonlinear, time-varying operator. Furthermore, we do not require that the n_w uncertainties are independent. However, we do require that the gain of the uncertainty block is bounded. Referring to Figure 3-1, the gain constraint is that

$$\sup_{z \neq 0} \frac{\|\Delta z\|_2}{\|z\|_2} \leq \frac{1}{\gamma}, \tag{3.16}$$

where γ is a positive, real number. Clearly, from the Small Gain Theorem, we know that the system is guaranteed to be robustly stable if we can control the z output of the plant such that $\|z\|_2 / \|w\|_2 < \gamma$ for all allowable Δ.

We note that the gain constraint can be written as

$$z^T z - \gamma^2 w^T w \geq 0\,. \tag{3.17}$$

Written in this form, this constraint is used to define the uncertainty supply rate in a dissipation analysis. The constraint clearly contains no information about the phase of the uncertainty block (as would be useful if we were interested in a *real* uncertainty). It merely constrains the $\mathcal{L}_2$-induced norm of the block. According to dissipation theory,[49] the correct stability multiplier for this type of uncertainty block is $\sigma^2 I_{n_w}$, with σ a real scalar. The Popov multiplier, $H + Ns$, used for the $\mathcal{H}_2$/Popov problem is not appropriate. However, it is clear that the solution of the $\mathcal{H}_2$/*SN* problem can be thought of as a particular instance of the Popov multipliers, with $H = \sigma^2 I$ and $N = 0$. This implies that the methods of solution for the $\mathcal{H}_2$/Popov problem and the $\mathcal{H}_2$/*SN* problem can be similar.

The fact that we are restricted to use a single scalar block, $\sigma^2 I$, for the multiplier is the same as the fact that the scaling matrix, $\mathcal{D}$ in *SN* theory was restricted to this form in equation (2.12). The $\mathcal{H}_2$/*SN* problem derives its name from the fact that it combines an $\mathcal{H}_2$ performance requirement with the same nonlinear, time-varying uncertainty description that can be analyzed using *SN* analysis.

There is a second interpretation of the robustness condition that can be given in the $\mathcal{H}_2$/*SN* case. Consider the system of Figure 3-2, with the plant dynamics given by equation (3.15). We want the controller to deliver robust stability and performance in the face of a disturbance, d, which has a bounded spectrum, together with a

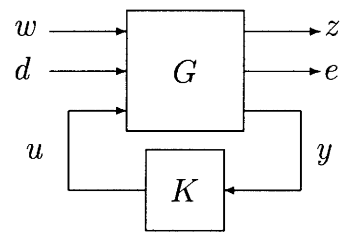

Figure 3-2: Robustness against an exogenous signal. The w disturbance is an $\mathcal{L}_2$ signal. A new signal, z, is measured at the output for the $\mathcal{H}_\infty$ norm of the system. The d signal may still be considered to be white noise. The e signal still represents the error signal by which the system's $\mathcal{H}_2$ performance is measured.

second disturbance, w, which has bounded power. We require that the closed loop system's first output, z, has bounded power. This is an $\mathcal{H}_\infty$ performance metric. Simultaneously, we require that the controller minimize a bound on the closed loop system's $\mathcal{H}_2$ norm, as measured from the error signal, e.

This second interpretation of the robustness problem leads to a problem formulation that is similar to the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ proposed in Ref. 5 and examined in other works.[86,56,72] However, the current problem formulation differs from the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ problem and various related problems because there are two distinct disturbance inputs with two distinct performance outputs. This is necessary so that the definition of the system's $\mathcal{H}_2$ performance makes sense. In the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ problem, only one input is defined. When discussing the $\mathcal{H}_\infty$ characteristics of the system, the input is considered to be an $\mathcal{L}_2$ signal, but when discussing the $\mathcal{H}_2$ performance of the system, the input is considered to a white noise signal. The implication of this fact is that, given a controller, it is possible, in the single input case, to exactly calculate the $\mathcal{H}_2$ performance of the system. In contrast, for the two input system, an exact means to calculate the $\mathcal{H}_2$ cost is not known, so only bounds on the system performance are calculated. Furthermore, it is likely that to reduce the conservatism of a design, different weights should be applied to the d and w inputs. Thus, the single input approach is not flexible enough to handle a system with an $\mathcal{H}_\infty$ performance metric that also possesses a distinct and meaningful $\mathcal{H}_2$ performance metric. Similarly, in Ref. 86, the two output signals, z and e are considered the same signal.

Fortunately, robustness to an exogenous, bounded-power signal can be analyzed using the same tools that were discussed for the case of the time-varying parametric uncertainty. This is because it is equivalent to requiring that the nominal closed loop system has infinity norm less than γ. We will typically find it more convenient to discuss the $\mathcal{H}_2/SN$ problem as if it derived solely from a parametric uncertainty framework. We will leave implicit the fact that the problem can also be derived from a system with dual exogenous disturbance inputs and a mixed performance criterion.

We can now concisely summarize the above discussion and state the problem below.

The $\mathcal{H}_2/SN$ design problem: Consider the system in Figure 3-1. The LTI plant, G, in the system is given by equation (3.15), with n states, n_u control inputs, a size n_d white-noise disturbance input, n_y measurements, and an error measurement of size n_e. The realization is assumed to be minimal. Furthermore, the control cost is decoupled from the state cost, so that $C_e^T D_{eu} = 0$. Also, we assume that (A, B_u) is stabilizable and that (A, C_y) is detectable. However, we do not put any requirements on the stabilizability and detectability of the uncertainty loop, (A, B_w) and (A, C_z). Lastly, we assume that the estimation problem is non-singular, implying that D_{yd} is full row rank.

The uncertainty block, Δ, is full, square, and size n_w. The uncertainty block has bounded $\mathcal{L}_2$-induced norm, such that $\|\Delta z\|_2/\|z\|_2 \leq 1/\gamma$ for all z.

The goal is to find a stability multiplier, σI, to describe the uncertainty block and a full-order, LTI controller given by equation (3.2). The controllers and stability multiplier should guarantee that the system is stable for any allowable uncertainty block. Moreover, they should provide an overbound on the $\mathcal{H}_2$ cost of the system given any allowable uncertainty.

Chapter 4

Linear Matrix Inequalities for Robust $\mathcal{H}_2$ Controller Synthesis

This chapter presents a novel solution to the controller synthesis problem posed in Chapter 3. The chapter does not tackle the full design problem, which is left to Chapter 6. Throughout the chapter, the stability multipliers (H and N for the $\mathcal{H}_2$/Popov case, σ for the $\mathcal{H}_2/SN$ case) will be considered fixed. The goal will be to find the controller that minimizes the bound on the $\mathcal{H}_2$ cost of the uncertain system, using fixed multipliers.

The solution is made possible because we are able to transform the robust $\mathcal{H}_2$ controller synthesis problem into a problem of minimizing a linear cost function subject to LMI constraints, *i.e.*, an eigenvalue problem. The procedure can be performed relatively independently of the type of stability multiplier that is used to characterize the system uncertainty. Thus, the synthesis procedure is derived in detail for the case of an $\mathcal{H}_2/SN$ controller only. Afterwards, the case of the $\mathcal{H}_2$/Popov controller is summarized. The LMI formulation for the $\mathcal{H}_2$/Popov case was derived by Livadas[57] by following the derivation of the $\mathcal{H}_2/SN$ case, which appears here. These synthesis procedures build upon the LMI-based synthesis procedures derived for $\mathcal{H}_\infty$ controllers in Ref. 33.

This synthesis procedure is interesting from a historical perspective, because it presents the first known method to use LMIs to synthesize a dynamic controller that optimizes the closed loop $\mathcal{H}_2$ cost (or a bound on the $\mathcal{H}_2$ cost) for an output feedback system. Previous LMI-based control synthesis procedures have only dealt with static $\mathcal{H}_2$ controllers using either full-state feedback[25] or output feedback.[53]

4.1 LMIs for $\mathcal{H}_2/SN$ Controllers

Recall the nominal plant, G, of equation (3.15) and the controller, K, given by equation (3.2). We wish to consider the nominal system formed by closing the loop between the plant and the controller. This is the lower linear fractional transformation of the system, $\mathcal{F}_l(G, K)$,

$$\begin{Bmatrix} \dot{\tilde{x}} \\ e \\ z \end{Bmatrix} = \begin{bmatrix} \tilde{A} & \tilde{B}_d & \tilde{B}_w \\ \tilde{C}_e & 0 & 0 \\ \tilde{C}_z & 0 & 0 \end{bmatrix} \begin{Bmatrix} \tilde{x} \\ d \\ w \end{Bmatrix}, \tag{4.1}$$

where

$$\tilde{A} = \begin{bmatrix} A & B_u C_c \\ B_c C_y & A_c \end{bmatrix}, \tag{4.2}$$

and

$$\tilde{B}_d = \begin{bmatrix} B_d \\ B_c D_{yd} \end{bmatrix} \quad \text{and} \quad \tilde{B}_w = \begin{bmatrix} B_w \\ 0 \end{bmatrix}, \tag{4.3}$$

$$\tilde{C}_e = \begin{bmatrix} C_e & D_{eu} C_c \end{bmatrix} \quad \text{and} \quad \tilde{C}_z = \begin{bmatrix} C_z & 0 \end{bmatrix}. \tag{4.4}$$

Note that the closed loop matrices contain the unknown terms from the compensator. Of course, this closed loop system is still a nominal system. The relation of the closed loop system to the uncertainty, Δ, is shown schematically in Figure 1-1b. From the figure, it is clear that the closed loop system and the uncertainty block are interconnected in the manner of the systems analyzed in Section 2.4. This means that we can use dissipation theory to bound the closed loop cost of the system. Applying the results of Theorem 2.7 to this system reveals that the cost is bounded by

$$J \leq \mathcal{J} = \text{tr}\left(\tilde{B}_d^T \tilde{P} \tilde{B}_d\right), \tag{4.5}$$

if a positive definite matrix $\tilde{P} \in \mathcal{S}^{2n}$ exists and satisfies the Riccati inequality

$$\tilde{A}^T\tilde{P} + \tilde{P}\tilde{A} + \tilde{C}_e^T\tilde{C}_e + \sigma^2\tilde{C}_z^T\tilde{C}_z + \frac{1}{\sigma^2\gamma^2}\tilde{P}\tilde{B}_w\tilde{B}_w^T\tilde{P} \leq 0\,. \tag{4.6}$$

Furthermore, if such a $\tilde{P}$ exists, the closed loop system will be stable for all uncertainties, Δ, where $\|\Delta z\|_2/\|z\|_2 \leq 1/\gamma$ for all z. Recall, once again, that the stability multiplier, σ, is considered to be fixed.

Unfortunately, while equation (4.5) and inequality (4.6) are perfectly useful for determining whether or not a *given* controller is acceptable, they give us no insight into how to solve for an *unknown* controller. The difficulty is that both the cost bound and matrix inequality constraint are nonlinear in the unknowns (the controller parameters and the Lyapunov matrix, P). This occurs because the closed loop matrix parameters contain the unknown controller variables. Thus, the robust $\mathcal{H}_2$ controller synthesis problem appears to require a nonlinear optimization. The main result of this section is given in the following theorem. It converts the nonlinear optimization problem into a linear, convex optimization problem.

Theorem 4.1 *For the system of equation (4.1) and a fixed stability multiplier, σ, define a matrix $\bar{B}_c \in \mathcal{R}^{n\times n_u}$ that is full column rank. Furthermore, define*

$$\bar{B}_d \triangleq \begin{bmatrix} B_d \\ \bar{B}_c D_{yd} \end{bmatrix}^T. \tag{4.7}$$

Consider the cost function, $\mathcal{J}$, given in equation (4.5), subject to the matrix inequality constraint given in equation (4.6). Then a positive definite matrix $\tilde{P}$ exists and minimizes the cost function subject to the constraint, if and only if a positive definite matrix $P \in \mathcal{S}^{2n}$ exists and minimizes

$$\mathrm{tr}\left(\bar{B}_d^T P \bar{B}_d\right)\,, \tag{4.8}$$

while satisfying

$$\begin{bmatrix} A^T P_{11} + P_{11} A + C_e^T C_e + \sigma^2 C_z^T C_z & \frac{1}{\sigma\gamma} P_{11} B_w \\ + C_y^T \bar{B}_c^T P_{21} + P_{21}^T \bar{B}_c C_y & \\ \frac{1}{\sigma\gamma} B_w^T P_{11} & -I \end{bmatrix} \leq 0 \,, \tag{4.9}$$

$$\begin{bmatrix} AQ_{11} + Q_{11}A^T - B_u(D_{eu}^T D_{eu})^{-1} B_u^T + \frac{1}{\sigma^2\gamma^2} B_w B_w^T & \sigma Q_{11} C_z^T & Q_{11} C_e^T \mathcal{W}_{eu} \\ \sigma C_z Q_{11} & -I & 0 \\ \mathcal{W}_{eu}^T C_e Q_{11} & 0 & -I \end{bmatrix} \leq 0 \tag{4.10}$$

where $\mathcal{W}_{eu}$ is an orthonormal basis for the nullspace of D_{eu}^T, and

$$\begin{bmatrix} P_{11} & P_{21}^T & I \\ P_{21} & P_{22} & 0 \\ I & 0 & Q_{11} \end{bmatrix} \geq 0 \,, \tag{4.11}$$

for some matrix $Q_{11} \in \mathcal{S}^n$. Furthermore, $\mathcal{J}$ is equal to the constrained minimum of (4.8).

Proof. To synthesize a controller, we would like to find the matrix $\tilde{P}$ that minimizes equation (4.5) and satisfies inequality (4.6). However, this is not a straightforward LMI minimization that we can solve. Inequality (4.6) contains terms that are the product of unknown compensator terms with the unknown $\tilde{P}$. Furthermore, the cost is also not linear in the unknown matrices. We wish to have a cost function that is linear in the unknown terms with inequality constraints that are also linear in the unknowns. This would form an EVP (see Section 2.1). We shall demonstrate that we do not, in fact, need to find $\tilde{P}$. Instead we shall prove that it exists and find a related matrix, P, that will serve to minimize the bound on the cost and fulfill the robustness constraints.

We first examine the cost function,

$$\mathcal{J} = \text{tr}\,(\tilde{B}_d^T \tilde{P} \tilde{B}_d) = \text{tr}\left(\begin{bmatrix} B_d \\ B_c D_{yd} \end{bmatrix}^T \tilde{P} \begin{bmatrix} B_d \\ B_c D_{yd} \end{bmatrix}\right) . \tag{4.12}$$

The cost contains the product of the unknown term B_c with $\tilde{P}$. The key to this proof is to recognize that we can, without loss of generality, assume that the B_c matrix has a fixed form, say $\bar{B}_c$. For instance, we might choose

$$\bar{B}_c = \begin{bmatrix} I_{n_y} \\ 0_{(n-n_y)\times n_y} \end{bmatrix} , \tag{4.13}$$

or

$$\bar{B}_c = C_y^T , \tag{4.14}$$

if C_y is full row rank. As long as $\bar{B}_c$ has full column rank, then we can use a similarity transform to change any realization of the controller to have a realization with $B_c = \bar{B}_c$. Note that if this were the typical state space realization for an LQG or a suboptimal $\mathcal{H}_\infty$ controller, then one would write the B_c matrix as the product of C_y^T with a $n \times n$ matrix that is the solution to a Riccati equation. Thus, choosing $\bar{B}_c = C_y^T$ makes the realization of the $\mathcal{H}_2/SN$ controller appear somewhat familiar.

At any rate, in specifying B_c, we have formed a cost function that is a linear function of the unknowns. For reasons that will be made clear later, we define a new matrix $P \in \mathcal{S}^{2n}$. Replacing $\tilde{P}$ with P in the expression for the bound, leads to the desired bound, equation (4.8).

We now turn our attention to finding a way to linearize the constraints. We define some new expressions that will allow us to separate out the compensator terms in our closed loop matrices. Note that only A_c and C_c remain undefined in the realization of the compensator. Define

$$\Theta = \begin{bmatrix} A_c \\ C_c \end{bmatrix} . \tag{4.15}$$

Also, matrices A_1, A_2, and A_3 are defined such that

$$\tilde{A} = \begin{bmatrix} A & B_u C_c \\ \bar{B}_c C_y & A_c \end{bmatrix} = \begin{bmatrix} A & 0 \\ \bar{B}_c C_y & 0 \end{bmatrix} + \begin{bmatrix} 0 & B_u \\ I & 0 \end{bmatrix} \begin{bmatrix} A_c \\ C_c \end{bmatrix} \begin{bmatrix} 0 & I \end{bmatrix}$$

$$= A_1 + A_2 \Theta A_3 \,. \tag{4.16}$$

Similarly,

$$\tilde{C}_e = \begin{bmatrix} C_e & D_{eu} C_c \end{bmatrix} = \begin{bmatrix} C_e & 0 \end{bmatrix} + \begin{bmatrix} 0 & D_{eu} \end{bmatrix} \begin{bmatrix} A_c \\ C_c \end{bmatrix} \begin{bmatrix} 0 & I \end{bmatrix}$$

$$= C_1 + C_2 \Theta A_3 \,. \tag{4.17}$$

Furthermore, we replace occurrences of $\tilde{P}$ in the Riccati inequality (4.6) by the previously defined P. With these definitions and by making use of Schur complements, the Riccati inequality can be rewritten as the matrix inequality

$$\begin{bmatrix} \tilde{A}^T P + P\tilde{A} + \sigma^2 \tilde{C}_z^T \tilde{C}_z & \frac{1}{\sigma\gamma} P\tilde{B}_w & \tilde{C}_e^T \\ \frac{1}{\sigma\gamma} \tilde{B}_w^T P & -I & 0 \\ \tilde{C}_e & 0 & -I \end{bmatrix} \leq 0 \,. \tag{4.18}$$

Again, this is not a *linear* matrix inequality since it contains products of unknown terms. Substituting the new definitions for the closed loop matrices into inequality (4.18) yields

$$\begin{bmatrix} A_1^T P + PA_1 + \sigma^2 \tilde{C}_z^T \tilde{C}_z & \frac{1}{\sigma\gamma} P\tilde{B}_w & C_1^T \\ \frac{1}{\sigma\gamma} \tilde{B}_w^T P & -I & 0 \\ C_1 & 0 & -I \end{bmatrix}$$

$$+ \begin{bmatrix} PA_2 \\ 0 \\ C_2 \end{bmatrix} \Theta \begin{bmatrix} A_3 & 0 & 0 \end{bmatrix} + \begin{bmatrix} A_3^T \\ 0 \\ 0 \end{bmatrix} \Theta^T \begin{bmatrix} A_2^T P & 0 & C_2^T \end{bmatrix} \leq 0 \,. \tag{4.19}$$

This has the form of the inequality in the Elimination Lemma (Lemma 2.1). In applying the lemma, the definitions of Ψ, U, and V are obvious from the form of inequality (4.19). Strictly speaking, the Elimination Lemma is written in terms of strict inequalities, while (4.18) and (4.19) are non-strict. It can be shown that the lemma can be applied to (4.19), because the upper left and lower right blocks of (4.18) are non-singular.

One could apply the Elimination Lemma without specifying bases for the required nullspaces. However, this would not allow one to learn about the structure of the LMIs. Therefore, we choose to specify bases for the nullspaces. Accordingly, we write

$$\begin{aligned} U &= \begin{bmatrix} A_2^T P & 0 & C_2^T \end{bmatrix} \\ &= U' \begin{bmatrix} P & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}, \end{aligned}$$

where

$$U' = \left[\begin{array}{cc|c|c} 0 & I & 0 & 0 \\ B_u^T & 0 & 0 & D_{eu}^T \end{array}\right],$$

and all matrix blocks are defined such that the corresponding multiplications make sense. Because B_u can be assumed to have full column rank, we can choose a basis for the nullspace of U' as

$$W_{U'} = \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \\ -(D_{eu}^T)^{\dagger} B_u^T & \mathcal{W}_{eu} & 0 \end{bmatrix},$$

where $\mathcal{W}_{eu}$ is an orthonormal basis for the nullspace of D_{eu}^T. Thus,

$$W_U = \begin{bmatrix} P^{-1} & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} W_{U'} .$$

For notational ease, we define the inverse of P to be Q. Furthermore, we define the matrices P_{ij} and Q_{ij} to be the size $n \times n$ blocks in the (i, j) partitions of the P and Q matrices, respectively. Then, multiplying out $W_U^T \Psi W_U$ leads to inequality (4.10). Inequality (4.9) is derived similarly. It comes from the second constraint stipulated in the Elimination Lemma,

$$W_V^T \Psi W_V \leq 0\,.$$

It is straightforward to show that we can choose

$$W_V = \begin{bmatrix} I_n & 0 & 0 \\ 0_{n\times n} & 0 & 0 \\ 0 & I_{n_w} & 0 \\ 0 & 0 & I_{2n} \end{bmatrix}$$

to find the desired LMI.

We now need only derive the third LMI of Theorem 4.1. Because we are minimizing the trace of $(\bar{B}_d^T P \bar{B}_d)$, the requirement that $P = Q^{-1}$ leads us to constrain P such that

$$P \geq Q^{-1}\,. \tag{4.20}$$

Clearly, because P is minimized, the above inequality will be driven to the edge of its constraint boundary, *i.e.*, such that the quantity $(P - Q^{-1})$ becomes singular. By Schur complements, this is equivalent to an LMI,

$$\begin{bmatrix} P_{11} & P_{21}^T & I & 0 \\ P_{21} & P_{22} & 0 & I \\ I & 0 & Q_{11} & Q_{21}^T \\ 0 & I & Q_{21} & Q_{22} \end{bmatrix} \geq 0\,. \tag{4.21}$$

Technically, this may not be enough to ensure that the matrices P and Q that emerge from the minimization are inverses. This is why the proof has been written in terms of a matrix P, rather than the desired matrix, $\tilde{P}$. The inverse condition will fail in the case where the cost function does not effectively minimize all of the eigenvalues of P. This is the case with most systems; it occurs when the B_d and B_c matrices have rank less than n. Fortunately, this is not a problem. As detailed in Refs. 82 and 57, the resulting P matrix can always be transformed to be $\tilde{P}$, such that the inverse condition is achieved, *i.e.*, $\tilde{P}^{-1} = Q$. The transform leaves the portions of the P matrix that affect either the cost function or the constraints unchanged, and only affects the unconstrained terms.

Finally, we can derive the final desired LMI, inequality (4.11). We note that portions of Q are not needed in the problem formulation. Matrices Q_{21} and Q_{22} appear nowhere in LMIs (4.9) and (4.10), nor in cost function (4.8). Thus, they are superfluous and can be dropped from LMI (4.21), leading to the desired LMI. ■

The proof of Theorem 4.1 is related to the work in Ref. 33, which deals with a method to synthesize suboptimal $\mathcal{H}_\infty$ controllers using LMIs. In the $\mathcal{H}_\infty$ case, the synthesis LMIs were derived from the Bounded Real Lemma. That lemma guarantees that if a given closed loop Riccati inequality (essentially inequality (4.6) without the $\tilde{C}_e$ term) is satisfied, then the desired level of $\mathcal{H}_\infty$ performance is achieved. Thus, in the $\mathcal{H}_\infty$ case, the development need only linearize the Riccati inequality. However, in the $\mathcal{H}_2/SN$ case, in addition to linearizing a Riccati inequality, we must explicitly minimize a bound on the $\mathcal{H}_2$ cost function. Thus, the key to the proof of Theorem 4.1 was the substitution of expression (4.8) for the cost function in equation (4.5).

Theorem 4.1 is significant because neither the bound on the cost, equation (4.8), nor the constraints given in inequalities (4.9)–(4.11) contain the unknown controller parameters. They are written solely in terms of the unknown Lyapunov matrices. Thus, the constraints are LMIs. The bound is also linear in the unknowns.

Note that there is no guarantee that two matrices P and Q exist that can satisfy the LMIs of Theorem 4.1. There is no reason to believe that any given $\mathcal{H}_2/SN$ problem necessarily has a solution. When the system has significant levels of uncertainty (*i.e.*, γ is small), the stability robustness constraint can become too restrictive for any controller. Clearly, no controller can yield a closed loop system that delivers a a level of γ below the minimum level of γ that can be achieved with an $\mathcal{H}_\infty$ controller. Also, note that the existence of a solution to these LMIs is sufficient to guarantee a solution to the problem, but these conditions are not necessary.

Assuming that the problem has a solution, after minimizing the bound on the cost function, equation (4.8), subject to LMIs (4.9)–(4.11), it may not be the case that the resulting matrix P is equal to the desired closed loop Lyapunov matrix, $\tilde{P}$. This occurs when P is not equal to the inverse of the resulting Q matrix. This is the case when the B_d and B_c matrices have rank less than n, so that the EVP does not constrain all of the eigenvalues of P. However, it is always possible to derive the matrix $\tilde{P}$, given P. This is done via a straightforward transformation detailed by Livadas[57,82] and others.[33] The details of this transform are secondary to the discussion at hand. Therefore, for purposes of discussion, we will always presume that the $\tilde{P}$ matrix is effectively determined by solving the EVP.

Given the closed loop Lyapunov matrix, $\tilde{P}$, the controller parameters can be found from the closed loop Riccati inequality. The Riccati inequality is equivalent to inequality (4.18), with $\tilde{P}$ replacing P. Because $\tilde{P}$ is known, this is an LMI for the unknown controller parameters, Θ, defined in equation (4.15). This becomes a feasibility problem for Θ, *i.e.*, any Θ which satisfies the LMI yields a valid controller. It is also a convex problem. Thus, there is a convex space in $\mathcal{R}^{(n+n_u)\times n}$ such that every point in the space fulfills the LMI. Each point in the space corresponds to a different (but entirely equivalent) realization for the optimal controller.

Rather than solving a feasibility problem, it is also the case that the controller parameters can be solved for algebraically. This is discussed in Ref. 33, where this is the suggested procedure for finding the controller parameters in a suboptimal $\mathcal{H}_\infty$ design.

Interestingly, we can derive yet a third, even simpler, method to solve for the controller parameters. A novel, closed form set of equations exists for the controller in terms of the Lyapunov matrix. This is the subject of the next theorem.

Theorem 4.2 *Given a matrix P that satisfies inequalities (4.9) and (4.10), and a matrix Q such that $Q = P^{-1}$, a realization for the controller that achieves the cost* *given by (4.8) is*

$$
\begin{aligned}
A_c &= -P_{21}^{-T}(P_{11}A + P_{21}^T B_c C_y + \sigma^2 C_z^T C_z + C_e^T C_e) Q_{11} Q_{21}^{-1} \\
&\quad -P_{21}^{-T}(A + \frac{1}{\sigma^2\gamma^2} B_w B_w^T P_{11} - B_u (D_{eu}^T D_{eu})^{-1} B_u^T P_{11})^T Q_{21}^{-1} && (4.22) \\
B_c &= \bar{B}_c && (4.23) \\
C_c &= -(D_{eu}^T D_{eu})^{-1} B_u^T Q_{21}^{-1}\,. && (4.24)
\end{aligned}
$$

Proof. This proof is essentially an extension of the proof of Theorem 4.1. As such, we shall reuse some of the notation defined in that proof.

Recall the definitions of Θ, $\tilde{A}_1$, $\tilde{A}_2$, $\tilde{A}_3$, $\tilde{C}_1$, $\tilde{C}_2$, Ψ, U, and V from the previous proof. We form the matrix $[\ W_V \quad W_U\ ]$. However, we drop certain redundant columns for clarity, leaving a matrix

$$
W_{UV} \triangleq \begin{bmatrix} I_n & 0 & Q_{11} & 0 & 0 \\ 0 & 0 & Q_{21} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{n_w} \\ 0 & I_{n_w} & -\left(D_{eu}^T\right)^{\dagger} B_u^T & \mathcal{W}_{eu} & 0 \end{bmatrix}.
$$

Following the ideas in the previous proof, we form the expression

$$
W_{UV}^T \left(\Psi + U^T\Theta V + V^T\Theta^T U\right) W_{UV} \triangleq \begin{bmatrix} \Sigma_{11} & \Sigma_{21}^T \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \le 0\,, \qquad (4.25)
$$

where

$$
\Sigma_{11} = \begin{bmatrix} P_{11}A + P_{21}^T \bar{B}_c C_y + A^T P_{11} + C_y^T \bar{B}_c^T P_{21} + \sigma^2 C_z^T C_z & (\cdots)^T \\ C_e & -I \end{bmatrix}
$$

$$
\Sigma_{21} = \begin{bmatrix} A - B_u D_{eu}^{\dagger} C_e + Q_{21}^T (A_c^T P_{21} + C_c^T B_u^T P_{11}) & Q_{11} C_e^T + B_u D_{eu}^{\dagger} \\ +Q_{11}(A^T P_{11} + C_y^T \bar{B}_c^T P_{21} + \sigma^2 C_z^T C_z) & +Q_{21}^T C_c^T D_{eu}^T \\ \mathcal{W}_{eu}^T C_e & -\mathcal{W}_{eu}^T \\ \frac{1}{\sigma\gamma} B_w^T P_{11} & 0 \end{bmatrix}
$$

$$
\Sigma_{22} = \begin{bmatrix} AQ_{11} + Q11A^T + \sigma^2 Q_{11} C_z^T C_z Q_{11} & & \\ -Q11C_e^T \left(D_{eu}^T\right)^{\dagger} B_u^T - B_u D_{eu}^{\dagger} C_e Q_{11} & (\cdots)^T & (\cdots)^T \\ -B_u D_{eu}^{\dagger} \left(D_{eu}^T\right)^{\dagger} B_u^T & & \\ \mathcal{W}_{eu}^T C_e Q_{11} & -\mathcal{W}_{eu}^T C_e Q_{11} & (\cdots)^T \\ \frac{1}{\sigma\gamma} B_w^T & 0 & -I \end{bmatrix}.
$$

The matrix inequality (4.25) can then be simplified by replacing it with its Schur complements, formed by reducing out the 2nd, 4th, and 5th rows. This leads to a trivial matrix inequality and a new inequality:

$$
\begin{bmatrix} \Phi_{11} & \Phi_{21}^T \\ \Phi_{21} & \Phi_{22} \end{bmatrix} \leq 0\,, \tag{4.26}
$$

where

$$
\Phi_{11} = \begin{bmatrix} P_{11} A + P_{21}^T \bar{B}_c C_y + A^T P_{11} + C_y^T \bar{B}_c^T P_{21} & (\cdots)^T \\ +\sigma^2 C_z^T C_z + C_e^T \mathcal{W}_{eu} \mathcal{W}_{eu}^T C_e & \\ C_e - \mathcal{W}_{eu} \mathcal{W}_{eu}^T C_e & -I \end{bmatrix}
$$

$$
\Phi_{21} = \left[\begin{array}{c|c} +A - B_u D_{eu}^{\dagger} C_e + \frac{1}{\sigma} B_w B_w^T P_{11} & \\ +Q_{21}^T (A_c^T P_{21} + C_c^T B_u^T P_{11}) & Q_{11} \left(C_e^T - C_e^T \mathcal{W}_{eu} \mathcal{W}_{eu}^t\right) \\ +Q_{11} \left(A^T P_{11} + C_y^T \bar{B}_c^T P_{21}\right. & +Q_{21}^T C_c^T D_{eu}^T + B_u D_{eu}^{\dagger} \\ \left.+\sigma^2 C_z^T C_z + C_e^T \mathcal{W}_{eu} \mathcal{W}_{eu}^T C_e\right) & \end{array}\right]
$$

$$
\begin{aligned} \Phi_{22} = {} & (A - B_u D_{eu}^{\dagger} C_e) Q_{11} + Q_{11} (A - B_u D_{eu}^{\dagger} C_e)^T - B_u (D_{eu}^T D_{eu})^{-1} B_u^T \\ & + \frac{1}{\sigma^2 \gamma^2} B_w B_w^T + \sigma^2 Q_{11} C_z^T C_z Q_{11} + Q_{11} C_e^T \mathcal{W}_{eu} \mathcal{W}_{eu}^T C_e Q_{11} \end{aligned}
$$

These expressions can then be simplified by recognizing that since $D_{eu}^T C_e = 0$, then $D_{eu}^{\dagger} C_e = 0$, $C_e^T \mathcal{W}_{eu} \mathcal{W}_{eu}^T C_e = C_e^T C_e$, and $C_e - \mathcal{W}_{eu} \mathcal{W}_{eu}^T C_e = 0$. Then, $\Phi_{11} \leq 0$ is seen to be a Schur complement of LMI constraint (4.9), and $\Phi_{22} \le 0$ has the form of a Schur complement of LMI constraint (4.10). Thus, setting $\Phi_{21} = [\ 0 \quad 0\ ]$, is a sufficient condition to require that (4.9) and (4.10) are satisfied. This condition then reveals two matrix equations

$$A + \frac{1}{\sigma} B_w B_w^T P_{11} + Q_{21}^T (A_c^T P_{21} + C_c^T B_u^T P_{11}) \\ + Q_{11} \left(A^T P_{11} + C_y^T \bar{B}_c^T P_{21} + \sigma^2 C_z^T C_z \right) = 0 \tag{4.27}$$

and

$$Q_{21}^T C_c^T D_{eu}^T + B_u D_{eu}^\dagger = 0 \,, \tag{4.28}$$

These two equations can be explicitly solved for A_c, and C_c, leading to equations (4.22) and (4.24). ■

Theorem 4.2 clearly provides a faster method for obtaining the controller parameters than either solving an LMI feasibility problem or solving an algebraic equation. However, it is also clear that the realization for the controller given in equations (4.22)–(4.24) does not provide us any insight into the workings of this controller. The realization is an extremely complicated expression. It does not have the convenient, observer-based form that we typically see for, say, an LQG controller or a suboptimal $\mathcal{H}_\infty$ controller.

Nevertheless, together, Theorems 4.1 and 4.2 form a complete solution to the $\mathcal{H}_2/SN$ controller synthesis problem for many systems. To use the theorems, one must minimize the cost function of equation (4.8) subject to the LMI constraints in equations (4.9)–(4.10) with respect to P and Q_{11}. This is an EVP. After solving the EVP, one can substitute the resulting Lyapunov matrix to equations (4.22)–(4.24) to obtain the controller. Because the EVP is convex, if the problem has a solution it can be found in polynomial time using off-the-shelf codes. For a problem with n states, we have observed solution times for this EVP proportional to between n^4 and n^5 using LMI Lab.[30,32,31] For relatively small problems, this solution speed is probably acceptable. A synthesis problem will be solved using this LMI technique in

Chapter 6.

It is worthwhile to examine the size of the EVP problem posed by Theorem 4.1. LMIs (4.9)–(4.10) are of outer dimension $n + n_w$, $n + n_w + n_e$, and $3n$. We must solve for four size $n \times n$ matrices, P_{11}, P_{21}, P_{22}, and Q_{11}. For systems of even moderate size, this would typically be considered a large optimization problem. The difficulty is that an optimization routine needs to find the Jacobian and, in most cases, at least an approximation to the Hessian of the problem. The Hessian's size is on the order of the square of the number of unknowns. For many real-world systems, the memory required to calculate and store these matrices makes an LMI approach to the synthesis problem impractical. This is the major drawback of the procedures proposed in this Chapter.

It should also be pointed out that the development in Theorems 4.1 and 4.2 is general enough that it encompasses the standard $\mathcal{H}_2$ optimal (LQG) and $\mathcal{H}_\infty$ suboptimal controllers. If the uncertainty terms are set to zero, then we have a derivation for an LQG controller. If the uncertainty terms are set to zero and we accept any point that satisfies the LMIs rather than minimizing the cost function, then we obtain an $\mathcal{H}_\infty$ controller. Of course, there is no reason that one should ever want to try to synthesize either an LQG or an $\mathcal{H}_\infty$ controller (full order) using the given LMI method. These controllers can be efficiently synthesized using well-established techniques that rely upon separating the problem into a state-feedback problem and an observer problem. These sub-problems are then solved efficiently using Riccati equations. The apparent similarities between the robust $\mathcal{H}_2/SN$ controller and these simpler, non-robust controllers lead us ask whether or not our robust controllers could be synthesized using a separation principle. This question is the subject of Chapter 5.

4.2 LMIs for $\mathcal{H}_2$/Popov Controllers

This section discusses how LMIs may be used to synthesize $\mathcal{H}_2$/Popov controllers. The uncertainty in the system is assumed to be best described by using both of the two Popov stability multipliers, H and N. These multipliers will be considered fixed, but their presence makes the synthesis procedure different than the procedure outlined for the $\mathcal{H}_2/SN$ controller in the previous section.

Although the final synthesis procedure for the $\mathcal{H}_2$/Popov system is more complex than that of the $\mathcal{H}_2/SN$ controller, the theory behind its derivation is fundamentally the same. Therefore, we will merely give the applicable theorems without proof. For a full discussion and proof of the results in this section, the reader is referred to either Ref. 82 or 57.

For the $\mathcal{H}_2$/Popov system, closing the loop between the plant and the unknown compensator leads to the same closed loop matrices defined in equations (4.2)–(4.4). With these definitions, Theorem 2.5 allows us to say that the square of the closed loop $\mathcal{H}_2$ cost, J, is bounded by

$$J \leq \mathcal{J} \triangleq \text{tr}\, \left[\tilde{B}_d^T \left(\tilde{P} + \tilde{C}_z^T (M_2 - M_1) N \tilde{C}_z\right) \tilde{B}_d\right] , \tag{4.29}$$

if a positive definite matrix $\tilde{P} \in \mathcal{S}^{2n}$ exists and satisfies the matrix inequality

$$\begin{bmatrix} \begin{array}{c} \tilde{A}^T(\tilde{P} - \tilde{C}_z^T N M_1 \tilde{C}_z) + (\tilde{P} - \tilde{C}_z^T M_1 N \tilde{C}_z)\tilde{A} + \tilde{C}_e^T \tilde{C}_e \\ -\tilde{C}_z^T M_1 M_d H M_2 \tilde{C}_z - \tilde{C}_z^T M_2 H M_d M_1 \tilde{C}_z \end{array} & (\cdots)^T \\ \begin{array}{c} \tilde{B}_w^T(\tilde{P} - \tilde{C}_z N M_1 \tilde{C}_z) + N\tilde{C}_z\tilde{A} \\ +M_d H M_2 \tilde{C}_z + H M_d M_1 \tilde{C}_z \end{array} & \begin{array}{c} \tilde{B}_w^T \tilde{C}_z^T N + N\tilde{C}_z \tilde{B}_w \\ -H M_d - M_d H \end{array} \end{bmatrix} \leq 0. \tag{4.30}$$

Furthermore, if a valid $\tilde{P}$ exists, then the closed loop system will be stable for all allowable uncertainties.

As in the case with the $\mathcal{H}_2/SN$ controller, the cost bound, equation (4.29) and the constraint (4.30) are both nonlinear in the unknowns. Fortunately, we do not have to solve this nonlinear optimization. The following theorem explains how the cost function and constraint can be linearized, making the problem convex.

Theorem 4.3 *For the system of equation (4.1), given fixed stability multipliers, H and N, define a matrix $\bar{B}_c \in \mathcal{R}^{n \times n_u}$ that is full column rank. Furthermore, define $\bar{B}_d$ as in equation (4.7). Then, a positive definite matrix $\tilde{P}$ exists and minimizes* *the bound on the cost function, $\mathcal{J}$, given in equation (4.29), subject to the matrix inequality constraint given in equation (4.30), if and only if a positive definite matrix $P \in \mathcal{S}^{2n}$ exists and minimizes*

$$\text{tr}\ \left[\bar{B}_d^T \left(P + \tilde{C}_z^T (M_2 - M_1) N \tilde{C}_z\right) \bar{B}_d\right] , \tag{4.31}$$

while satisfying

$$\begin{bmatrix} \begin{array}{c} \hat{A}^T P_{11} + P_{11}\hat{A} + C_e^T C_e \\ + C_y^T \bar{B}_c^T P_{21} + P_{12}\bar{B}_c C_y \end{array} & C_z^T H + \hat{A}^T C_z^T N + P_{11} B_w \\ HC_z + NC_z\hat{A} + B_w^T P_{11} & \begin{array}{c} B_w^T C_z^T N + NC_z B_w \\ -HM_d - M_d H \end{array} \end{bmatrix} \leq 0 , \tag{4.32}$$

$$\begin{bmatrix} Q_{11}\hat{A}^T + \hat{A}Q_{11} - B_u^T (D_{eu}^T D_{eu})^{-1} B_u & (\cdots)^T & (\cdots)^T \\ \begin{array}{c} HC_z Q_{11} + NC_z \hat{A} Q_{11} + B_w^T \\ -NC_z B_u (D_{eu}^T D_{eu})^{-1} B_u^T \end{array} & \begin{array}{c} -NC_z B_u (D_{eu}^T D_{eu})^{-1} B_u^T C_z^T N \\ + B_w^T C_z^T N + NC_z B_w \\ -HM_d - M_d H \end{array} & 0 \\ C_e Q_{11} & 0 & -I \end{bmatrix} \leq 0, \tag{4.33}$$

and

$$\begin{bmatrix} P_{11} & P_{21}^T & I \\ P_{21} & P_{22} & 0 \\ I & 0 & Q_{11} \end{bmatrix} \geq 0 , \tag{4.34}$$

for some matrix $Q_{11} \in \mathcal{S}^n$, where $\hat{A} \triangleq A + B_w M_1 C_z$. Furthermore, $\mathcal{J}$ is equal to the constrained minimum of (4.31).

Proof. See Ref. 57. ■

Theorem 4.3 reveals that an $\mathcal{H}_2$/Popov controller can be derived by minimizing a linear cost function with respect to a Lyapunov matrix, subject to LMI constraints.

Unfortunately, unlike in the $\mathcal{H}_2/SN$ case, we are unable to give a closed form solution for the $\mathcal{H}_2$/Popov controller in terms of the Lyapunov matrix. It does not appear possible to derive a set of controller equations using the same procedure that was used to derive Theorem 4.2. In trying to derive the analogous theorem, we would attempt to find a set of conditions that are sufficient, but not necessary, to guarantee that LMIs (4.32)–(4.34) are satisfied. Unfortunately, the presence of the Popov multipliers would complicate this procedure. In comparison to the robustness constraint in the $\mathcal{H}_2/SN$ case (see inequality (4.25)), additional terms would appear in the comparable inequality for the $\mathcal{H}_2$/Popov case. These terms would contain the product of the Popov multiplier N and the the system dynamics matrix, $\tilde{A}$. It can be shown that this would preclude us from explicitly solving for A_c, and, thus, from deriving a controller realization.

In the previous section, comments were made regarding the utility of LMIs for synthesizing an $\mathcal{H}_2/SN$ controller. Topics included the speed of optimization routines and the amount of memory that was required to solve the problem. These comments apply equally to the case of the $\mathcal{H}_2$/Popov controller and need not be repeated.

Chapter 5

A Separation Principle for Robust $\mathcal{H}_2$ Controllers

This chapter presents a second solution to the controller synthesis problem posed in Chapter 3. Unlike the solution presented in Chapter 4, this latest synthesis technique does not rely upon LMIs. Rather, it relies upon dividing the problem into two simpler problems. Since this chapter deals with controller synthesis, rather than controller design, the stability multipliers are considered fixed throughout the chapter.

The chapter demonstrates that a bound on the closed loop $\mathcal{H}_2$ cost of a system with a robust $\mathcal{H}_2$ controller can be separated into the sum of two parts. The first of these parts is a bound on the cost of a related full-state feedback or full information control problem. The second part is equal to a bound on the cost of a related output estimation problem. The separation property of the bound on the cost implies that robust $\mathcal{H}_2$ controllers possess an identifiable observer-based structure. The problem of finding the optimal controller can then be separated into two sub-problems. The first of these is to solve the aforementioned state feedback or full information control problem. The second sub-problem is to solve the related output estimation problem. The final control signal is formed by multiplying the state estimate found from the output estimation problem by the gain matrix found from the state feedback problem.

The motivation behind this work lies in the desire to develop efficient means to synthesize robust $\mathcal{H}_2$ controllers. The efficacy of using this separation structure will not be demonstrated until Chapter 6. There, it will be mentioned that, in practice, controllers synthesized by solving the state feedback and output estimation problems are identical to controllers synthesized by using the LMI-based techniques. This is apparently the case even though theorems in this chapter only demonstrate that the separation-based cost function (formed by summing the costs from the two sub-problems) overbounds the cost function used in the LMI-based minimization. However, as will be discussed, there is a close relation between the two cost functions in the $\mathcal{H}_2$/*SN* case, and we speculate that controllers derived via the two methods should, even for the Popov case, be equal.

It is well known that a Youla parameterization allows any stabilizing controller to be written as the combination of a state feedback gain with an observer (see, for instance, Ref. 18). However, except for a few significant exceptions, this separation structure is typically not exploited in the controller synthesis process. Either the structure is too difficult to identify, or, presumably, it is not useful. It is the fact that the separation principle simplifies the synthesis of robust $\mathcal{H}_2$ controllers that makes the principle significant.

The separation principle for robust $\mathcal{H}_2$ controllers resembles that found in the state-space derivation of the $\mathcal{H}_\infty$ controller[23] but differs from it because of the stochastic nature of the $\mathcal{H}_2$ performance index. This makes the derivation unique.

For pedagogical reasons, the separation structure will be derived first for the case of the $\mathcal{H}_2$/*SN* controller (which has a simplified set of stability multipliers). This will allow us to compare the structure to that of the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ controller of Bernstein & Haddad.[5] Then, in the second section in this chapter, an expanded separation principle is presented to cover the case of Popov multipliers. The last section of the chapter discusses how the separation principle leads to a synthesis technique for a class of suboptimal controllers.

5.1 A Separation Principle for $\mathcal{H}_2/SN$ Controllers

In this section, we will show that an $\mathcal{H}_2/SN$ controller can be derived by solving a robust estimation problem and a state feedback problem. The control signal will be the optimal state estimate multiplied by the state feedback gain. We will consider each of the two smaller problems first, before proceeding to the main problem of interest, the output feedback problem. The smaller problems are considered in a more general context than the full problem, which will assume that some of the weights on the cost and system parameters are normalized. Finally, the last part of this section discusses the relationship between controllers derived using the separation principle and controllers derived using LMI techniques.

5.1.1 $\mathcal{H}_2/SN$ Full-State Feedback

The full-state feedback (FSFB) problem is to find the static optimal control gain that robustly minimizes the system performance metric, given exact knowledge of the states of the system. None of the disturbance terms is measured, as this information is superfluous if one knows the states. This problem retains most of the structure from the original system (3.15). However, now we set $C_y = I$ and $D_{yd} = 0$, so that the system is

$$\begin{Bmatrix} \dot{x} \\ e \\ z \\ y \end{Bmatrix} = \begin{bmatrix} A & B_d & B_w & B_u \\ C_e & 0 & 0 & D_{eu} \\ C_z & 0 & 0 & 0 \\ I & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} x \\ d \\ w \\ u \end{Bmatrix}. \tag{5.1}$$

This problem is different from the "full information" problem traditionally associated with $\mathcal{H}_\infty$ control problems[23] for two reasons. First, this FSFB problem retains the original system's separate outputs for the performance and stability criteria. Secondly, the measurement vector does not include any information about the disturbances.

It is assumed *a priori* that the optimal control for this problem is a linear function of the states. The optimal control gain is determined in the next theorem.

Theorem 5.1 *For the system in (5.1), given a fixed scaling, σ, if there exists a* *positive semidefinite matrix* $R \in \mathcal{S}^n$ *that is the stabilizing solution to the Riccati equation*

$$RA + A^T R + C_e^T C_e + \sigma^2 C_z^T C_z - RB_u (D_{eu}^T D_{eu})^{-1} B_u^T R + \frac{1}{\sigma\gamma^2} RB_w B_w^T R = 0 \,, \quad (5.2)$$

then the optimal control that stabilizes the system against all uncertainty blocks with $\mathcal{L}_2$*-induced norm less than* $(1/\gamma)$ *and minimizes the cost is given by*

$$u = -(D_{eu}^T D_{eu})^{-1} B_u^T R x \,. \quad (5.3)$$

In this case, an upper bound on the cost, J*, is given by*

$$\mathcal{J}_{\text{FS}} = \text{tr}\, \left(B_d^T R B_d\right) \,. \quad (5.4)$$

Proof. This proof is based on dissipation theory. Without the presence of the error term, e, this would essentially be the same as the $\mathcal{H}_\infty$ full-information problem. Then, this discussion would essentially be the same as the Lyapunov-type analysis performed for the standard $\mathcal{H}_\infty$ state-feedback system found in Ref. 8. Similarly, if the inputs were equal ($B_w = B_d$), then this controller should yield the same result as that discussed in Ref. 72. Furthermore, were it not for the presence of the unknown control term, the following dissipation argument would be the same as that of Ref. 44.

For stability and performance, we desire the sum of the supply rate for the uncertainty, r_Δ, a performance supply rate, r_P, and the derivative of the storage function for the plant, $\dot{V}_G$, to be negative for all values of x and w. In particular, we require that

$$\dot{V}_G + r_\Delta + r_P \leq 0 \,, \quad (5.5)$$

where

$$\begin{aligned} V_G &= x^T R x \,, \quad \text{for some positive definite matrix } R \in \mathcal{S}^n \\ r_\Delta &= \sigma\left(z^T z - \gamma^2 w^T w\right) \end{aligned}$$

$$r_P = e^T e\,.$$

Note that the uncertainty supply rate is required to be positive to be part of a dissipation argument. Substituting in the dynamics from (5.1) to the dissipation inequality and assuming that $u = \bar{K}x$, we find that

$$x^T R(Ax + B_w w + B_u \bar{K}x) + (Ax + B_w w + B_u \bar{K}x)^T Rx$$
$$+(C_e x + D_{eu}\bar{K}x)^T (C_e x + D_{eu}\bar{K}x) + \sigma x^T C_z^T C_z x - \sigma\gamma^2 w^T w \leq 0\,. \quad (5.6)$$

We can set the optimal control gain by locating where the left hand side of (5.6) is extremal with respect to $\bar{K}$. Differentiating the left hand side with respect to $\bar{K}$ and setting the result equal to zero yields

$$\bar{K} = -(D_{eu}^T D_{eu})^{-1} B_u^T R\,. \quad (5.7)$$

This demonstrates (5.3).

Next, note that (5.6) must hold for all values of x and w. Thus, it is equivalent to

$$\begin{bmatrix} R(A+B_u\bar{K}) + (A+B_u\bar{K})^T R + \sigma C_z^T C_z \\ +(C_e + D_{eu}\bar{K})^T (C_e + D_{eu}\bar{K}) & RB_w \\ B_w^T R & -\sigma\gamma^2 I \end{bmatrix} \leq 0\,. \quad (5.8)$$

By Schur complements, this last inequality is equivalent to

$$R(A+B_u\bar{K})+(A+B_u\bar{K})^T R+(C_e+D_{eu}\bar{K})^T (C_e+D_{eu}\bar{K})+\sigma C_z^T C_z+\frac{1}{\sigma\gamma^2} RB_w B_w^T R \leq 0. \quad (5.9)$$

At the optimal control gain, however, this inequality is pushed against the constraint boundary, so, in fact, it is an equality. Substituting in the optimal gain from equation (5.7) and recalling that $D_{eu}^T C_e = 0$ yields (5.2). Finally, an explanation for why equation (5.4) provides a useful bound for the cost is provided in Ref. 82. ∎

A second method to derive equation (5.2) can be found in Ref. 44. We wish to find the worst-case disturbance, $w = w^*$, *i.e.*, the signal that will maximize the bound on the cost function. Similarly to before, we differentiate the left hand side of equation (5.6) with respect to w, and find that

$$w^* = (\sigma\gamma^2)^{-1} B_w^T R x \,. \tag{5.10}$$

Substituting the expression for w^* and the optimal gain from equation (5.7) to equation (5.6) yields the desired expression. The expression can be written as a Riccati equation, rather than an inequality, because, together, the optimal control and worst-case disturbance make the expression singular.

It is worthwhile to discuss the FSFB Riccati equation, equation (5.2), in some detail. The equation combines terms that we would expect from an $\mathcal{H}_2$ Riccati equation with those of an $\mathcal{H}_\infty$ Riccati equation. For negligible levels of uncertainty ($\gamma \to \infty$), the equation looks like an LQR Riccati equation (recall equation (3.6)). Given that (A, B_u) is stabilizable, this equation will always have a stabilizing solution. However, for higher levels of uncertainty, the $(B_w B_w^T/(\sigma\gamma^2))$ term affects the quadratic coefficient in the equation. As γ decreases, the quadratic term eventually becomes sign indefinite, and/or the quantity

$$\left[B_u (D_{eu}^T D_{eu})^{-1} B_u^T + \frac{1}{\sigma\gamma^2} B_w B_w^T \right]^{1/2}$$

fails to form a stabilizing pair with A. In these cases, the Riccati equation can fail to have a stabilizing solution. Thus, for high levels of uncertainty, not even a full-state feedback controller will necessarily be able to guarantee the stability of the system.

5.1.2 $\mathcal{H}_2/\mathcal{SN}$ Output Estimation

The motivation for the Output Estimation (OE) problem will become clear when the full output feedback problem is examined. For now, consider a system

$$\begin{Bmatrix} \dot{x} \\ v \\ y \end{Bmatrix} = \begin{bmatrix} A & B_d & B_r & B_u \\ C_v & 0 & 0 & I \\ C_y & D_{yd} & 0 & 0 \end{bmatrix} \begin{Bmatrix} x \\ d \\ r \\ u \end{Bmatrix}, \tag{5.11}$$

with just two outputs, the measurement, y, and a performance signal, v. The system is subject to a white noise disturbance, d, and a bounded power disturbance, r. The performance signal contains a direct feed-through contribution from the control, u. This problem retains part of the mixed nature of the original problem because of the twin inputs.

The goal in this problem is to find a dynamic filter that minimizes the expected value of the quantity $(v^T v - r^T r)$. Such a filter can be found by applying the following theorem.

Theorem 5.2 *For the system of equation (5.11), which is subject to a zero-mean, unit covariance, white-noise signal, d, and signal $r \in \mathcal{L}_2$, if there exist positive semidefinite matrices $X \in \mathcal{S}^n$ and $Y \in \mathcal{S}^n$ that are the stabilizing solutions to the coupled Riccati equations*

$$\begin{aligned} X\left(A - (YC_y^T + B_d D_{yd}^T)\, E_d^{-1} C_y\right) + \left(A - (YC_y^T + B_d D_{yd}^T) E_d^{-1} C_y\right)^T X \\ + X B_r B_r^T X + C_v^T C_v = 0 \end{aligned} \tag{5.12}$$

and

$$\begin{aligned} (A + B_r B_r^T X)Y + Y(A + B_r B_r^T X)^T - (YC_y^T + B_d D_{yd}^T) E_d^{-1} (C_y Y + D_{yd} B_d^T) \\ + B_d B_d^T = 0\,, \end{aligned} \tag{5.13}$$

where $E_d \triangleq D_{yd}D_{yd}^T$*, then a filter exists such that the expected value of* $(v^Tv - r^Tr)$ *is bounded by*

$$\mathcal{J}_{\mathrm{OE}} = \mathrm{tr}\ (B_d - HD_{yd})^T X (B_d - HD_{yd})\,, \tag{5.14}$$

where $H = (YC_y^T + B_d D_{yd}^T)(E_d)^{-1}$*. Lastly, a realization for the filter which achieves this is given by*

$$\left[\begin{array}{c|c} A - HC_y - B_u C_v & H \\ \hline -C_v & 0 \end{array}\right]. \tag{5.15}$$

Proof. We presume *a priori* that the optimal filter has a fixed, model–based, observer structure, given by

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + B_u u + H(y - C_y\hat{x}) \\ u &= -C_v\hat{x}\,, \end{aligned} \tag{5.16}$$

where $\hat{x}$ is the state estimate, H is an unknown filter gain, and the other terms are from (5.11). If we define the error coordinates, $\tilde{x} \triangleq x - \hat{x}$, then the closed loop system with the filter is

$$\begin{aligned} \dot{\tilde{x}} &= (A - HC_y)\tilde{x} + (B_d - HD_{yd})d + B_r r \\ v &= C_v\tilde{x}\,. \end{aligned} \tag{5.17}$$

Note that, because the control input is known, it does not appear in the error dynamics. As was done for the FSFB case, we can use dissipation to form a constraint that bounds the closed loop system's performance and guarantees its stability. The dissipation inequality is

$$\dot{V}_K + r_P \leq 0\,, \tag{5.18}$$

and we note that there is no uncertainty term. Taking

$$\begin{aligned} V_K &= \tilde{x}^T X \tilde{x}\,, \quad \text{for some positive definite matrix } X \in \mathcal{S}^n \\ r_P &= v^T v - r^T r\,, \end{aligned}$$

and substituting in the state dynamics, the dissipation inequality can be written as

$$\tilde{x}^T X((A - HC_y)\tilde{x} + B_r r) + ((A - HC_y)\tilde{x} + B_r r)^T X x + \tilde{x}^T C_v^T C_v \tilde{x} - r^T r \leq 0 \,. \quad (5.19)$$

This inequality must hold for all $(\tilde{x}, r)$, so it is equivalent to

$$\begin{bmatrix} X(A - HC_y) + (A - HC_y)^T X + C_v^T C_v & X B_r^T \\ B_r X & -I \end{bmatrix} \leq 0 \,. \quad (5.20)$$

By appealing to the results of Theorem 2.8, we can see that, given a matrix X that satisfies this inequality, a bound on the cost is given by

$$\text{tr}\, (B_d - HD_{yd})^T X (B_d - HD_{yd}) \,. \quad (5.21)$$

We note that the smallest symmetric matrix X that satisfies inequality (5.20) is also a solution to the Riccati equation,[81]

$$(A - HC_y)^T X + X(A - HC_y) + X B_r B_r^T X + C_v^T C_v = 0 \,. \quad (5.22)$$

Thus, for a given gain H, the bound on the cost can be minimized by solving Riccati equation (5.22). It should be noted that this Riccati equation could also be derived by differentiating inequality (5.20) with respect to r, to determine the extremal value for r. Substituting the extremal value for r back to the inequality would yield the Riccati equation.

As an aside, it should be mentioned that, for a stabilizing H, Riccati equation (5.22) also happens to be a constraint on the $\mathcal{H}_\infty$ norm of the closed loop system. Thus, if the disturbance r had a bounded $\mathcal{L}_2$ norm, then this constraint would guarantee that $\|v\|_2^2 / \|r\|_2^2 < 1$.

Note that cost function (5.21) has the form of the desired cost function (5.14). We wish to minimize this cost function, subject to the Riccati constraint. We form an augmented cost functional

$$\begin{aligned}\mathcal{J}_{\text{aug}} &= \text{tr } (B_d - HD_{yd})^T X (B_d - HD_{yd}) \\ &+ \text{tr } Y \left[(A - HC_y)^T X + X(A - HC_y) + XB_rB_r^TX + C_v^TC_v \right] , \qquad (5.23)\end{aligned}$$

where the Lagrange multiplier, $Y \in \mathcal{R}^n$, is a symmetric matrix. The performance bound is a steady state bound, so $\mathcal{J}_{\text{aug}}$ must be stationary with respect to the optimal gain, H^*. Taking the necessary derivatives and finding the stationary point leads to two necessary conditions

$$\begin{aligned}&(A + B_rB_r^TX - H^*C_y)Y + Y(A + B_rB_r^TX - H^*C_y)^T \\ &-H^*D_{yd}B_d^T - B_dD_{yd}^TH^{*T} + H^*D_{yd}D_{yd}^TH^{*T} + B_dB_d^T \;=\; 0 \qquad (5.24)\end{aligned}$$

and

$$H^* = (YC_y^T + B_dD_{yd}^T)(E_d)^{-1} , \qquad (5.25)$$

where $E_d \triangleq (D_{yd}D_{yd}^T)$. Note that E_d is always invertible since D_{yd} was assumed to be full row rank. Substituting the form for H^* into (5.22) and (5.24) leads to the desired conditions (5.12) and (5.13). Finally, the formulae for the filter parameters come from substituting the form for H^* into equation (5.16). ■

It is interesting to note that the gain matrix, H, has same form as the gain in a Kalman Filter (see the LQG controller input matrix in equation (3.4)) or as in an $\mathcal{H}_\infty$ optimal estimator. However, H is different, because the matrix Y, which has the effect of a covariance matrix, is not the same as the covariance matrix from either of those problems. Instead, Y must satisfy a pair of coupled, nonlinear equations (5.12) and (5.13).

Together, equations (5.12) and (5.13) serve as a set of sufficiency conditions for the existence of a solution to the output estimation problem. Neither equation is a Riccati equation. However, they are referred to as "coupled Riccati equations." The first of these, equation (5.12), can be thought of as an equation for X. Given a fixed Y, it is a Riccati equation that can be solved for the worst-case disturbance input,

$B_r^T X$. Likewise, given a fixed X, the second equation (5.13) is a Riccati equation in Y that can be solved for the best possible estimator gain, $(YC_y^T + B_d D_{yd}^T)(E_d)^{-1}$. The first equation retains its inherent $\mathcal{H}_\infty$-type of form, while the second has an $\mathcal{H}_2$-like structure (due to the negative definite quadratic term), reflecting the mixed nature of this problem.

5.1.3 $\mathcal{H}_2/SN$ Output Feedback

We turn to the real problem of interest. The full output feedback problem will now be broken down into two smaller problems via a separation principle.

For the sake of clarity, we will examine the case where various weights have been normalized. We now assume that the static scaling, σ, is unity and that the closed loop $\mathcal{H}_\infty$ gain of the system, γ, is unity. Given a problem with σ or γ different from unity, we can absorb these constants into the C_z and B_w terms and continue the problem with these modified matrices. Similarly, we assume that $D_{eu}^T D_{eu} = I$.

Theorem 5.3 *For the system of equation (3.15), with $\sigma = 1$ and $\gamma = 1$, if a matrix $R \in \mathcal{S}^n$ exists such that it is the unique, positive semidefinite, stabilizing solution to the related full-state feedback Riccati equation*

$$RA + A^T R + C_e^T C_e + C_z^T C_z - RB_u B_u^T R + RB_w B_w^T R = 0\,, \tag{5.26}$$

and positive semidefinite matrices $X \in \mathcal{S}^n$ and $Y \in \mathcal{S}^n$ exist and are the stabilizing solutions to the related output estimation coupled Riccati equations

$$\begin{aligned} X\left(A + B_w B_w^T R - (YC_y^T + B_d D_{yd}^T)E_d^{-1}C_y\right) + XB_w B_w^T X + RB_u B_u^T R \\ + \left(A + B_w B_w^T R - (YC_y^T + B_d D_{yd}^T)E_d^{-1}C_y\right)^T X = 0 \end{aligned} \tag{5.27}$$

and

$$\begin{aligned} \left(A + B_w B_w^T (R + X)\right) Y + Y\left(A + B_w B_w^T (R + X)\right)^T \\ -(YC_y^T + B_d D_{yd}^T)E_d^{-1}(C_y Y + D_{yd} B_d^T) + B_d B_d^T = 0\,, \end{aligned} \tag{5.28}$$

where $E_d = (D_{yd} D_{yd}^T)$*, then a controller given by*

$$A_c = A + B_w B_w^T R - B_c C_y - B_u C_c \quad (5.29)$$

$$B_c = \left(Y C_y^T + B_d D_{yd}^T \right) E_d^{-1} \quad (5.30)$$

$$C_c = -B_u^T R \quad (5.31)$$

will stabilize the system for all Δ *with* $\mathcal{L}_2$*-induced norm less than* $(1/\gamma)$*. Furthermore, for all such* Δ*, the closed-loop performance of the system,* J*, will be bounded by*

$$\mathcal{J}_{\mathrm{OF}} = \mathrm{tr}\ \left\{ B_d^T R B_d + \left[B_d - B_c D_{yd} \right]^T X \left[B_d - B_c D_{yd} \right] \right\} . \quad (5.32)$$

Proof. The proof of this theorem relies upon the results from the FSFB problem and the OE problem. First, assume that a positive definite solution exists for R in (5.26). To see how the full problem separates into the state feedback problem and the estimation problem, examine the steady state expected value of a candidate Lyapunov function given by $x^T R x$. Along state trajectories, we have that

$$\begin{aligned} \lim_{t\to\infty} \frac{d}{dt} E\{x^T R x\} &= \frac{d}{dt} \lim_{t\to\infty} E\{x^T R x\} = 0 \quad (5.33) \\ &= \lim_{t\to\infty} \frac{d}{dt} E\{x^T R x\} \\ &= \lim_{t\to\infty} E\{x^T R (Ax + B_w w + B_u u) + (Ax + B_w w + B_u u)^T R x\} \\ &\quad + \mathrm{tr}\ (B_d^T R B_d) \\ &= \lim_{t\to\infty} E\left\{ x^T (RA + A^T R) x + x^T R B_w w + w^T B_w^T R x \right. \\ &\quad \left. + x^T R B_u u + u^T B_u^T R x \right\} + \mathrm{tr}\ (B_d^T R B_d) , \quad (5.34) \end{aligned}$$

where the last equality is obtained by substituting in for the state dynamics, and the term outside the expected value arises from taking the derivative of the expected value of the white noise input, d.

We now examine a dissipation inequality that will guarantee that the closed loop system is stable and will allow us to obtain a bound on the closed loop cost. The inequality is the same as that discussed in the proof of Theorem 5.1, *i.e.*,

$$\dot{V}_G + r_\Delta + r_P \leq 0\,. \tag{5.35}$$

If we use the same forms for the supply rates and storage function used before, then the dissipation inequality tells us that we are guaranteed to achieve both robust stability and performance if we require that

$$\begin{aligned} x^T R(Ax + B_w w + B_u u) + (Ax + B_w w + B_u u)^T Rx \\ +(C_e x + D_{eu} u)^T (C_e x + D_{eu} u) + x^T C_z^T C_z x - w^T w \leq 0\,, \end{aligned} \tag{5.36}$$

We can obtain extremal values for the control and the disturbance by differentiating the above inequality with respect to u and w. The extremal values will be needed to complete the square in the dissipation inequality so that the separation principle can be identified. The extremal control and disturbance are

$$u^* \triangleq -B_u^T Rx \tag{5.37}$$

$$w^* \triangleq B_w^T Rx\,. \tag{5.38}$$

We wish to examine the dissipation inequality when both the control and disturbance are set to their extremal values. In other words, we examine the case where

$$\begin{aligned} x^T R(Ax + B_w w^* + B_u u^*) + (Ax + B_w w^* + B_u u^*)^T Rx \\ + (C_e x + D_{eu} u^*)^T (C_e x + D_{eu} u^*) + x^T C_z^T C_z x - w^{*T} w^* \leq 0\,. \end{aligned} \tag{5.39}$$

Substituting the definitions of u^* and w^* to inequality (5.39) and completing squares allows us to rewrite the extremal dissipation inequality as

$$\begin{aligned} x^T R(Ax + B_w w + B_u u) + (Ax + B_w w + B_u u)^T Rx + (x^T C_z^T C_z x - w^T w) \\ + x^T C_e^T C_e x + u^T u - (u - u^*)^T (u - u^*) + (w - w^*)^T (w - w^*) \leq 0\,. \end{aligned} \tag{5.40}$$

From here, we can see that

$$x^T R(Ax + B_w w + B_u u) + (Ax + B_w w + B_u u)^T R x \leq -(x^T C_z^T C_z x - w^T w) \\ - x^T C_e^T C_e x - u^T u + (u - u^*)^T (u - u^*) - (w - w^*)^T (w - w^*) . \quad (5.41)$$

Now, we return our attention to the general case, with non-extremal controls and disturbances. Equation (5.34) holds true in general. A comparison of this equation with inequality (5.41) reveals that

$$\lim_{t\to\infty} E\left\{-(x^T C_z^T C_z x - w^T w) - x^T C_e^T C_e x - u^T u + (u - u^*)^T (u - u^*) \right. \\ \left. - (w - w^*)^T (w - w^*)\right\} + \text{tr}\,(B_d^T R B_d) \geq 0 . \quad (5.42)$$

This last inequality can also be obtained by substituting for $x^T(RA + A^T R)x$ from the FSFB Riccati equation into equation (5.34).

For clarity, we make the following definitions

$$v \triangleq u - u^* = u + B_u^T R x \quad (5.43)$$

$$r \triangleq w - w^* = w - B_w^T R x , \quad (5.44)$$

such that v to represents the difference between the actual control and the extremal control, and r represents the difference between the actual disturbance, w and the extremal disturbance. Inequality (5.42) can now be written as

$$-\lim_{t\to\infty} E\left\{x^T C_e^T C_e x + u^T u\right\} + \lim_{t\to\infty} E\left\{w^T w - z^T z\right\} \\ + \lim_{t\to\infty} E\left\{v^T v - r^T r\right\} + \text{tr}\,(B_d^T R B_d) \geq 0 . \quad (5.45)$$

Next, recalling the definition of the cost function, we see that the performance metric that we would like to minimize can be written as

$$J \triangleq \lim_{t\to\infty} E\{e^T e\} = \lim_{t\to\infty} E\{x^T C_e^T C_e x + u^T u\} . \quad (5.46)$$

Furthermore, recall that $\|\Delta\|_\infty < 1$, so that

$$\lim_{t\to\infty} E\left\{w^T w - z^T z\right\} \leq 0\,. \tag{5.47}$$

Combining these facts with (5.45) reveals that

$$J \leq \text{tr}\,(B_d^T R B_d) + \lim_{t\to\infty} E\{v^T v - r^T r\}\,. \tag{5.48}$$

Thus, the cost is bounded by the sum of two terms. The first term is the FSFB cost bound. The second term will be shown to be related to an output estimation problem. To minimize the second term, we would like to push $v^T v$ as close to zero as possible. If we knew the state, x, then we could set $u = -B_u^T R x$ and $v^T v$ would be identically zero. Thus, this is an estimation problem. The $r^T r$ portion of the cost can be thought of as the amount by which the uncertainty differs from the worst–case uncertainty. Thus, this term helps decrease the cost.

To bound the cost function given by the right hand side of (5.48), we rewrite the system dynamics using the definitions for v and r. The original system, given in equation (3.15), is equivalent to

$$\begin{aligned} \dot{x} &= Ax + B_w B_w^T R - B_w B_w^T R + B_d d + B_w w + B_u u \\ &= A_R x + B_d d + B_r r + B_u u\,, \end{aligned} \tag{5.49}$$

where

$$A_R \triangleq A + B_w B_w^T R \tag{5.50}$$

$$B_r \triangleq B_w \tag{5.51}$$

$$C_v \triangleq B_u^T R\,. \tag{5.52}$$

Then, the open loop system is equivalent to

$$\begin{Bmatrix} \dot{x} \\ v \\ y \end{Bmatrix} = \begin{bmatrix} A_R & B_d & B_r & B_u \\ C_v & 0 & 0 & I \\ C_y & D_{yd} & 0 & 0 \end{bmatrix} \begin{Bmatrix} x \\ d \\ r \\ u \end{Bmatrix} . \tag{5.53}$$

Since we are interested in minimizing $(v^T v - r^T r)$, the original performance variable, e, is not relevant to this new, reduced problem. Minimizing $\mathcal{J}_{\text{OF}}$ for the system in (5.53) is clearly the situation that was examined in the OE problem, with A_R replacing A. Using Theorem 5.2, we see that to solve this problem, we must satisfy equivalent forms to (5.12) and (5.13). This leads to the desired, coupled Riccati equations (5.27) and (5.28). Furthermore, the desired controller, given in (5.29)–(5.31), is derived from the OE filter given in (5.15), by replacing A with A_R.

Lastly, we consider the cost function. Given that we recognize that part of the cost is bounded by the cost from an output estimation problem, the boundary given in (5.48) defines a bound on the cost as

$$J \leq \mathcal{J}_{\text{OF}} \triangleq \mathcal{J}_{\text{FS}} + \mathcal{J}_{\text{OE}} , \tag{5.54}$$

where $\mathcal{J}_{\text{FS}}$ and $\mathcal{J}_{\text{OE}}$ were defined in (5.4) and (5.14), respectively. Substituting in these definitions yields the desired bound (5.32). ■

According to this theorem, a solution to the $\mathcal{H}_2$/SN synthesis problem can be obtained by solving for the variables R, X, and Y, in equations (5.26)–(5.28). A close examination of the three equations reveals that equation (5.26), the FSFB equation, is written in terms of R alone. Equations (5.27) and (5.28) contain all three variables. This implies that equation (5.26) can be solved first, independently of the other two equations.

Assuming that the solution for R is known, equations (5.27) and (5.28) are then a pair of coupled, nonlinear, matrix equations for the unknowns X and Y. These last two equations are referred to as "coupled Riccati equations," because if Y is fixed, equation (5.27) is a Riccati equation for X. Likewise, for a fixed X, equation (5.28) is a Riccati equation for Y.

Note that we now have two seemingly different solutions to the $\mathcal{H}_2/SN$ controller synthesis problem — one in terms of coupled Riccati equations and one in terms of an LMI problem (from Chapter 4). We believe that they are, in fact, the same. Why this is felt to be the case is discussed later. The latest synthesis approach is interesting because it requires the solution of a smaller problem than is required from solving the LMI problem. The LMI problem posed in Theorem 4.1 is to minimize a linear cost functional subject to three LMIs of outer dimension $n + n_w$, $n + n_w + n_e$, and $3n$. In practice, an optimization routine can have great difficulty working with these LMIs as constraints. The Jacobians, and possibly Hessians, of the constraints must be evaluated, and the memory requirements for this operation can be prohibitive.

In contrast, the new derivation involves three matrix equations of size n. This solution does not require an explicit minimization of the cost function. If the solution to the three equations exists, then it is the minimal solution for $\mathcal{J}_{\mathrm{OF}}$. Suppose, for the sake of argument, that the coupled Riccati equations can be solved using standard Riccati equation solvers. In this case, if the equations can be solved via the successive solution of the three Riccati equations, say, n times each, then this should be at least as fast as the LMI-based solution for the problem. A Riccati equation can be solved more quickly than a comparably sized LMI problem. Current convex programming methods for LMI problems require computational times on the order of between n^4 and n^5. In contrast, a Riccati equation has approximately the complexity of the eigenvalue problem for its related Hamiltonian system, so it can be solved in times on the order of $(2n)^3$. Of course, for small systems, any speed advantage of the Riccati equations may be negligible.

It should be mentioned that the coupled Riccati equations in Theorem 5.3 are very similar to the coupled Riccati equations posed by Bernstein and Haddad for their full-order, mixed $\mathcal{H}_2/\mathcal{H}_\infty$ controller.[5] The matrices R, X and Y play similar roles to the variables used in Ref. 5, which are components of the closed loop covariance matrix. Dual forms for the Riccati equations can be developed if equations (5.26), (5.27) and (5.28) can be written in terms of R^{-1}, X^{-1}, and Y^{-1}. However, the problems are different in the sense that this formulation allows for the white noise disturbance to affect the system differently than the $\mathcal{L}_2$ disturbance, *i.e.*, $B_d \neq B_w$. In practice, if one were to formulate the problem without the ability to separate these inputs, then one would need to define a modified input matrix, $B = \begin{bmatrix} B_d \\ B_w \end{bmatrix}$, and replace all occurrences of B_d and B_w with this term. This would be conservative.

For pedagogical reasons, we can also point out how the output feedback controller encompasses the standard, non-robust, optimal, $\mathcal{H}_2$ controller, *i.e.*, an LQG controller. Recall that the LQG solution is found by solving equations (3.6) and (3.7). Equations (5.26)–(5.28) can be seen to be equivalent if the uncertainty terms, C_z and B_w, are set to zero. Note that D_{eu} has been normalized in the $\mathcal{H}_2/SN$ case. With the uncertainty terms neglected, equation (5.26) is seen to be identical to equation (3.6). Furthermore, equation (5.28) becomes decoupled from equation (5.27). It is seen to be equivalent to the Kalman filter Riccati equation (3.7) with $Y \longrightarrow Q$. What happens to the third equation in the robust case? With the uncertainty effectively eliminated, equation (5.27) becomes

$$X\left(A-(YC_y^T+B_dD_{yd}^T)E_d^{-1}C_y\right)+RB_uB_u^TR \\ +\left(A-(YC_y^T+B_dD_{yd}^T)E_d^{-1}C_y\right)^T X=0\,,$$

which is a Lyapunov equation for X. Note that the quantity $A-(YC_y^T+B_dD_{yd}^T)E_d^{-1}C_y$ is guaranteed to be stable because of the choice of Y (or Q) from the filter Riccati equation. This Lyapunov equation has a solution because of the assumption that (A, B_u) is stabilizable and that R is full rank. Thus, the equation is superfluous in the limiting case.

The $\mathcal{H}_2/SN$ formulation does not exactly capture the standard $\mathcal{H}_\infty$ central controller formulation,[23] because the system has been slightly altered from the $\mathcal{H}_\infty$ case. To have an equivalent problem, we would need to be able to directly measure part of the w disturbance, *i.e.*, we would need to add a D_{yw} feedthrough term.

We believe that an $\mathcal{H}_2/SN$ controller synthesized using LMIs, as discussed in Theorem 4.2, is equivalent to an $\mathcal{H}_2/SN$ controller derived using the separation principle given in Theorem 5.3. This fact is important because the LMI formulation was based on first principles (Lyapunov theory, dissipation, etc.). Therefore, the LMI formulation can be accepted as yielding a controller that will give the tightest possible quadratic bound on the $\mathcal{H}_2$ cost. In contrast, the separation principle formulation presumed a form for the controller. It is conceivable, therefore, that the formulation has accidentally excluded the desired controller from consideration.

Unfortunately, we are unable to prove that the two controllers solutions are the same. We can show that, given a solution to the LMI-based synthesis problem, this solution is also a valid solution of the separation-based synthesis problem. A proof of the converse is incomplete. The converse proof shows that, given a solution to the separation-based synthesis problem, this solution is equal to the optimal solution that would be obtained by solving the LMI-based synthesis problem over only a restricted set in the space of available Lyapunov matrices.

To see this, first consider that we have a solution to the LMI-based synthesis problem. We have a matrix $\tilde{P} \in \mathcal{S}^{2n}$ that was obtained by minimizing the cost function in equation (4.8), subject to LMIs (4.9)–(4.11). The controller in this case is given by equations (4.22)–(4.24). Note that, by definition, this controller and the accompanying Lyapunov matrix satisfy the closed loop Riccati equation (formed by substituting the closed loop realization in equations (4.2)–(4.4) into the Riccati equation of Theorem 2.8). We wish to transform the controller to have a realization that will fulfill the coupled Riccati equations of the separation principle. The effect of this on the Lyapunov matrix is a similarity transformation that leaves unchanged the upper left block of the matrix (the part corresponding to the states of the plant). Specifically, we need merely transform

$$\begin{bmatrix} \tilde{P}_{11} & \tilde{P}_{21}^T \\ \tilde{P}_{21} & \tilde{P}_{22} \end{bmatrix} \quad \text{to} \quad \begin{bmatrix} \tilde{P}_{11} & -P_{22} \\ -P_{22} & P_{22} \end{bmatrix}.$$

Because $\tilde{P}$ is positive definite, its lower right block, $\tilde{P}_{22}$, is also positive definite, so this transformation will always exist. Thus, we can obtain a matrix, say $P \in \mathcal{S}^{2n}$, that has the desired form.

To demonstrate that the transformed system satisfies the separation principle, we must demonstrate that matrices X, Y, and R exist that satisfy the FSFB Riccati equation and the OE coupled Riccati equations. Choosing $X = P_{22}$ and $R = \tilde{P}_{11} - P_{22} = P_{11} - P_{22}$ will demonstrate that equations (5.26) and (5.27) are satisfied. To see this, examine the lower right, size $n \times n$, block of the closed loop Riccati equation, which is

$$P_{22}A_c + A_c^T P_{22} + P_{21}B_u C_c + C_c^T B_u^T P_{21}^T + C_c^T D_{eu}^T D_{eu} C_c + P_{21} B_w B_w^T P_{21}^T = 0 \,. \quad (5.55)$$

We substitute into equation (5.55) for A_c and C_c from equations (4.22) and (4.24), respectively. Then, noting that $P_{21} = -P_{22}$, the matrix inversion lemma reveals that

$$\begin{aligned} Q_{11}^{-1} &= P_{11} - P_{21}^T P_{22}^{-1} P_{21} \\ &= P_{11} - P_{22} = (X + R) - X = R \end{aligned} \quad (5.56)$$

$$\begin{aligned} Q_{21}^{-1} &= -Q_{11}^{-1} P_{22}^{-1} P_{21} \\ &= Q_{11}^{-1} = R \,. \end{aligned} \quad (5.57)$$

Armed with these facts, after some algebra, equation (5.55) can be seen to be equivalent to the first OE coupled Riccati equation (5.27), with B_c left unspecified. Similarly, the upper right block of the closed loop Riccati equation can be shown to equal the sum of equations (5.26) and (5.27). Thus, the FSFB Riccati equation (5.26 is also satisfied. Finally, since equations (5.26 and (5.27) are satisfied, a solution for Y is guaranteed to exist from equation (5.28) (because the coefficient of the quadratic term is negative and the system is detectable). Furthermore, the cost function that is minimized subject to the LMI constraints can be examined. It is

$$\text{tr}\left(\bar{B}_d^T P \bar{B}_d\right) = \text{tr}\left(B_d^T P_{11} B_d + B_d^T P_{21} B_c D_{yd} + D_{yd}^T B_c^T P_{21} B_d + D_{yd}^T \bar{B}_c^T P_{22} \bar{B}_c D_{yd}\right)$$

$$
\begin{aligned}
&= \operatorname{tr}\left(B_d^T(X+R)B_d - B_d^T X B_c D_{yd} - D_{yd}^T B_c^T X B_d + D_{yd}^T \bar{B}_c^T X \bar{B}_c D_{yd}\right) \\
&= \operatorname{tr}\left(B_d^T R B_d + [Bd - \bar{B}_c D_{yd}]^T X [Bd - \bar{B}_c D_{yd}]\right) \\
&= \mathcal{J}_{\text{FS}} + \mathcal{J}_{\text{OE}} \,. \qquad (5.58)
\end{aligned}
$$

Thus, the solution found by solving the LMI synthesis problem satisfies the separation principle.

We now examine the converse statement. Assume that a solution exists for matrices R, X, and Y from equations (5.26), (5.27), and (5.28). We can define a size $2n \times 2n$ matrix to satisfy the closed loop Riccati equation. Reversing the procedure just discussed, we choose

$$
\begin{aligned}
P_{11} &\triangleq X + R \qquad (5.59) \\
P_{21} &\triangleq -X \qquad (5.60) \\
P_{22} &\triangleq X \,. \qquad (5.61)
\end{aligned}
$$

Using the separation-based controller realization of Theorem 5.3, and this definition for a candidate Lyapunov matrix, P, it is straightforward to show that the closed loop Riccati equation is satisfied. For instance, as already mentioned, summing equations (5.27), and (5.26) yields the upper left block of the closed loop Riccati equation. The other blocks are obtained similarly. Therefore, we have demonstrated that the separation-based controller satisfies the desired robustness constraints. However, this has not demonstrated that the matrix P is equal to the Lyapunov matrix that would be obtained by minimizing cost function (4.8) subject to the three LMI constraints. Therefore, the converse is not proved.

However, it is the case that both the FSFB and OE problems have implicitly minimized the individual cost functions related to those problems. Referring back to the LMI cost function, written in equation (5.58), it is clear that the separation-based solution does yield a minimal value for the cost function if the space of Lyapunov matrices is restricted to the form

$$\begin{bmatrix} A & -B \\ -B & B \end{bmatrix} > 0 \tag{5.62}$$

with $A = A^T$, and $B = B^T$. Thus, the converse holds in a limited sense. If the restricted set in the space of available Lyapunov matrices could be shown to contain the optimal Lyapunov matrix for certain realizations of $\bar{B}_c$, then the converse would be proven in full.

These facts lead us to propose the following conjecture:

Conjecture 5.1 *Consider the system of equation (3.15). With $\sigma = 1$ and $\gamma = 1$, examine the controller of equations (4.22)–(4.24), found by minimizing (4.8), subject to (4.9)–(4.11). This controller is equivalent to the controller given by (5.29)–(5.31), found by solving equations (5.26)–(5.28). Furthermore, the minimal value of the bound on the cost found from each minimization is the same.*

This conjecture has not been proven. It will, however, be shown in Chapter 6 that the conjecture seems to hold in practice.

Turning aside from the question of whether or not the two controller solutions are equivalent, we can also give an interpretation for one of the LMI constraints used in the LMI-based solution. For the solution to exist, there must exist a matrix Q_{11} which satisfies LMIs (4.10) and (4.11). In fact, if the $\bar{B}_c$ matrix is set to have the realization of the B_c matrix in the separation-based solution, then the matrix Q_{11} will satisfy the following equality (based on the Schur complement of LMI (4.10)),

$$AQ_{11} + Q_{11}A^T - B_uB_u^T + B_wB_w^T + Q_{11}C_z^TC_zQ_{11} + Q_{11}C_e^TC_eQ_{11} = 0\,. \tag{5.63}$$

We can show that the separation-based solution satisfies this condition as well. Multiplying equation (5.63) through on the left and right by Q_{11}^{-1} reveals that it has the the same form as the FSFB Riccati equation (5.26). Thus, the second LMI in Theorem 4.1 can be thought of as a constraint on the controller gain, much as the FSFB Riccati equation yields a controller gain.

The motivation for the definitions of P_{ij} blocks in the previous discussion comes from a careful examination of the closed loop version of the cost function in Theorem 2.7 (see also equation (4.5)). That cost function is (again)

$$\text{tr}\ \left(\bar{B}_d^T P \bar{B}_d\right) = \text{tr}\ \left(B_d^T P_{11} B_d + B_d^T P_{21} B_c D_{yd} + D_{yd}^T B_c^T P_{21} B_d + D_{yd}^T \bar{B}_c^T P_{22} \bar{B}_c D_{yd}\right) \ .$$

It is useful to think of the trace($B_d^T P_{11} B_d$) term in this expression as the "state cost" in the bound, *i.e.*, that portion of the cost that is due to the system states being nonzero. The other terms of the cost relate to the compensator states or some combination of compensator and plant states. This state cost term has the same form as the $\mathcal{H}_2$ cost one could derive for the open loop system (assuming the nominal plant is stable). Furthermore, the state cost term also has the same form as the bound on the cost of a system with a state feedback controller, tr $(B_d^T R B_d)$.

Intuitively, it seems that the state cost (not to mention the entire cost function) for the output feedback controller must be greater than the cost for the system with a state feedback controller, because the former can only estimate the system states. The additional cost is accounted for in the matrix X. This provides an understanding for the definition that $P_{11} = X + R$. The definitions for the other two blocks of P follow from this initial observation.

These definitions also provide insight into the last LMI constraint in Theorem 4.1, which states that

$$\begin{bmatrix} P_{11} & P_{21}^T & I \\ P_{21} & P_{22} & 0 \\ I & 0 & Q_{11} \end{bmatrix} \geq 0 \, .$$

In Theorem 4.1, assume that we have chosen $\bar{B}_c$ to equal the controller input matrix found from separation-based solution. This implies that the P matrix satisfies the closed loop Riccati equation (substitute the closed loop system to the Riccati equation in Theorem 2.8). Using Schur complements, we can see that this LMI constraint is equivalent to (non-trivial part)

$$\begin{bmatrix} P_{11} - Q_{11}^{-1} & P_{21}^T \\ P_{21} & P_{22} \end{bmatrix} \geq 0 \,.$$

However, as discussed previously, in this case, Q_{11}^{-1} is equal to R, the solution of the FSFB Riccati equation. Thus, since the upper left block of the above LMI must be greater than or equal to zero, this last constraint can be interpreted as requiring that the state cost of an output feedback controller be greater than the associated cost for a system with a state feedback controller.

The fact that the LMI-based synthesis solution satisfies a separation principle is significant. First, this clarifies the relationship between the LMI-based solution to the $\mathcal{H}_2/SN$ problem and the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ controller of Bernstein and Haddad.[5] Using equations (5.59)–(5.61), we see that

$$\tilde{P} = \begin{bmatrix} X + R & -X \\ -X & X \end{bmatrix} \,. \tag{5.64}$$

As shown in Ref. 5, this is the same as the form that a closed loop covariance matrix should have for full-order controllers. Furthermore, the matrices X and R appear in the bound on the cost of this system in a similar manner to the way that the covariance terms appear in the bound on the cost of the mixed $\mathcal{H}_2/\mathcal{H}_\infty$ controller. Of course, P is a Lyapunov matrix, rather than the system covariance, but it seems that the inverse of P should be closely related to the covariance found in Ref. 5. They are dual problems. This duality, however, is limited in the sense that there is no guarantee that the dual problems will yield the same cost bound for a given problem.

If Conjecture 5.1 could be shown to hold, then this has implications for broader class of problems as well. We would effectively show that the full-order, mixed $\mathcal{H}_2/\mathcal{H}_\infty$ controller synthesis problem of Ref. 5 could be reformulated as an LMI, *i.e.*, it could be formulated in a convex fashion. Other control problems are similar, involving necessary conditions which contain coupled Lyapunov and Riccati equations. One example of this is the *nominal* $\mathcal{H}_2/\mathcal{H}_\infty$ problem posed by Ridgely *et al.*[70] This problem is typically solved using homotopy-type methods.[77] Conjecture 5.1 would imply that the necessary conditions found by Ridgely could be replaced with a convex problem written in terms of LMIs. For the nominal $\mathcal{H}_2/\mathcal{H}_\infty$ problem, other researchers have actually found this convexity result by other means.[12,76]

5.2 A Separation Principle for $\mathcal{H}_2$/Popov Controllers

This section expands on the previous work in the chapter to show that a robust $\mathcal{H}_2$ controller with Popov multipliers can be synthesized by solving a robust estimation problem and a full information control problem. The control signal will equal the optimal state estimate multiplied by the gain from the full information problem.

The separation-based cost function derived in this section will be shown to overbound the cost function that is minimized in the LMI-based synthesis method of Theorem 4.3. Nevertheless, despite the fact that the two cost functions do not match, it will be shown in Chapter 6 that controllers obtained from the two synthesis methods seem to match in practice.

As before, we will present the two subsidiary problems before presenting the problem of greatest interest, the output feedback problem. Throughout this section, all weights will be left in explicitly, to leave the development as general as possible. Of course, because we are dealing with just the synthesis problem, and not the full design problem, the Popov stability multipliers, H and N, will be considered fixed throughout this section.

5.2.1 $\mathcal{H}_2$/Popov Full Information Control

The goal of the full information (FI) control problem is to find the static feedback gain that robustly minimizes the system performance metric. In general, a full information control problem differs from a full-state feedback problem because the system allows independent measurements of both its states and a disturbance signal. Like the original problem, this problem has two disturbance signals, an exogenous broadband disturbance, d, and a disturbance due to the uncertainty in the system, w. However, we are now interested in a system that can measure w. The FI system is the same as the original system (3.15) except for modifications necessitated by the measurement change. In particular, we have that $C_y = [\ I \quad 0\ ]^T$, $D_{yw} = [\ 0 \quad I\ ]^T$, and $C_y^T D_{yw} = 0$, *i.e.*,

$$\begin{Bmatrix} \dot{x} \\ e \\ z \\ y \end{Bmatrix} = \begin{bmatrix} A & B_d & B_w & B_u \\ C_e & 0 & 0 & D_{eu} \\ C_z & 0 & 0 & 0 \\ \begin{bmatrix} I \\ 0 \end{bmatrix} & 0 & \begin{bmatrix} 0 \\ I \end{bmatrix} & 0 \end{bmatrix} \begin{Bmatrix} x \\ d \\ w \\ u \end{Bmatrix} . \tag{5.65}$$

A method to determine the optimal FI control gain is the subject of the following theorem.

Theorem 5.4 *For the system in (5.65), given fixed multipliers H and N, assume that the matrix*

$$Z \triangleq M_d H + H M_d - N C_z B_w - B_w^T C_z^T N + N C_z B_u E_u^{-1} B_u^T C_z^T N \tag{5.66}$$

is positive definite. In this case, if there exists a matrix R that is the unique, positive semidefinite, stabilizing solution to the Riccati equation,

$$\begin{aligned} &R\hat{A} + \hat{A}^T R + C_e^T C_e + \left\{ R(B_w - B_u E_u^{-1} B_u^T C_z^T N) + (C_z^T H + \hat{A}^T C_z^T N) \right\} \\ &\quad \cdot Z^{-1} \left\{ (H C_z + N C_z \hat{A}) + (B_w^T - N C_z B_u E_u^{-1} B_u^T) R \right\} - R B_u E_u^{-1} B_u^T R = 0 \,, \end{aligned} \tag{5.67}$$

where $\hat{A} \triangleq A + B_w M_1 C_z$ and $E_u \triangleq D_{eu}^T D_{eu}$, then, for all valid uncertainty blocks, Δ, the optimal control that minimizes the cost is given by

$$u = E_u^{-1} B_u^T \left(-Rx + C_z^T N M_1 C_z x - C_z^T N w \right) , \tag{5.68}$$

and an upper bound on the cost, J is given by

$$\mathcal{J}_{\mathrm{FI}} = \mathrm{tr}\ \left[B_d^T(R + C_z^T(M_2 - M_1)NC_z)B_d\right]\ . \tag{5.69}$$

Proof. This proof is based on dissipation theory. The proof differs from the $\mathcal{H}_2$/*SN* full-state feedback proof because the supply rates are specialized for the case of Popov stability multipliers. To achieve robust stability and performance, dissipation theory tells us that we must require that

$$\dot{V}_G + r_\Delta + r_P \leq 0\,, \tag{5.70}$$

where

$$\begin{aligned} V_G &= x^T R x\,, \quad \text{for some positive definite matrix } R \in \mathcal{R}^n \\ r_\Delta &= 2\left[(w - M_1 z)^T N\dot{z} + (w - M_1 z)^T M_d H(M_2 z - w)\right] \\ r_p &= e^T e\,, \end{aligned}$$

and $M_d \triangleq (M_2 - M_1)$. Writing out the dissipation inequality using the system dynamics and definitions of the supply rates and storage function yields,

$$\begin{aligned} x^T R(Ax + B_u u + B_w w) + (Ax + B_u u + B_w w)^T R x + (x^T C_e^T + u^T D_{eu}^T)(C_e x + D_{eu} u) \\ +2(w - M_1 C_z x)^T\left[NC_z(Ax + B_u u + B_w w) + M_d H(M_2 C_z x - w)\right] \leq 0 \end{aligned} \tag{5.71}$$

Interestingly, because of the form of the uncertainty supply rate, r_Δ, inequality (5.71) has terms involving products of the control, u, and the uncertainty input, w.

To find the optimal control gain, we must determine where the left hand side of (5.71) is extremal with respect to u. Differentiating with respect to u and setting this quantity to zero reveals that the extremal control is given by equation (5.68).

To obtain the desired Riccati equation, we substitute the expression for the optimal control back to inequality (5.71). After simplifying terms, we note that the resulting inequality must hold for all quantities $\begin{bmatrix} x & w \end{bmatrix}$. This implies that a related matrix is negative semidefinite, in particular, that

$$\left[\begin{array}{c|c} \begin{array}{c} RA + A^T R - C_z^T M_1 N C_z A - A^T C_z^T N M_1 C_z \\ + C_z^T M_1 N C_z B_u E_u^{-1} B_u^T R - R B_u E_u^{-1} B_u^T R \\ + R B_u E_u^{-1} B_u^T C_z^T N M_1 C_z + C_e^T C_e \\ - C_z^T M_1 N C_z B_u E_u^{-1} B_u^T C_z^T N M_1 C_z \\ - C_z^T M_1 M_d H M_2 C_z - C_z^T M_2 H M_d M_1 C_z \end{array} & (\cdots)^T \\ \hline \begin{array}{c} N C_z (A + B_u E_u^{-1} B_u^T C_z^T N M_1 C_z) \\ + (B_w^T - N C_z B_u E_u^{-1} B_u^T) R - B_w^T C_z^T N M_1 C_z \\ + M_d H M_2 C_z + H M_d M_1 C_z \end{array} & \begin{array}{c} - M_d H - H M_d \\ + N C_z B_w + B_w^T C_z^T N \\ - N C_z B_u E_u^{-1} B_u^T C_z^T N \end{array} \end{array}\right] \leq 0 \,, \tag{5.72}$$

(we have added lines to visually separate the matrix blocks for clarity). At this stage, we could form a Riccati equation by taking Schur complements of the above inequality and examining its extremal solutions. However, the above expression is rather complicated. We can simplify it using a congruence transformation. A congruence transform preserves the signature of a matrix, thus leaving the inequality valid. The transform is to multiply on the left and right by

$$\mathcal{T} = \begin{bmatrix} I & C_z^T M_1 \\ 0 & I \end{bmatrix} \tag{5.73}$$

and $\mathcal{T}^T$, respectively. This leads to the following matrix inequality

$$\begin{bmatrix} \begin{array}{c} R(A + B_w M_1 C_z) + (A + B_w M_1 C_z)^T R \\ + C_e^T C_e - R B_u E_u^{-1} B_u^T R \end{array} & (\cdots)^T \\ \begin{array}{c} N C_z (A + B_w M_1 C_z) + H C_z \\ (B_w^T - N C_z B_u E_u^{-1} B_u^T) R \end{array} & \begin{array}{c} - M_d H - H M_d \\ + N C_z B_w + B_w^T C_z^T N \\ - N C_z B_u E_u^{-1} B_u^T C_z^T N \end{array} \end{bmatrix} \leq 0 \,. \tag{5.74}$$

Taking Schur complements of (5.74) leads to the following

$$R\hat{A} + \hat{A}^T R + C_e^T C_e + \left\{R(B_w - B_u E_u^{-1} B_u^T C_z^T N) + (C_z^T H + \hat{A}^T C_z^T N)\right\}$$
$$\cdot Z^{-1} \left\{(HC_z + NC_z\hat{A}) + (B_w^T - NC_z B_u E_u^{-1} B_u^T)R\right\} - RB_u E_u^{-1} B_u^T \leq 0\,, \tag{5.75}$$

and

$$-Z \leq 0\,, \tag{5.76}$$

where Z is seen to be equal to the lower right $(2,2)$ block of (5.74). The fact that we require that $-Z$ be negative semidefinite ensures that the solution to the Riccati equation, R, will be positive semidefinite (see Lemma 3 of Ref. 23).

It should be noted that the value of R that is extremal in expression (5.71) will also be extremal with respect to inequality (5.75). Thus, we need only examine the extremal points of (5.75). This leads to the desired Riccati equation (5.67). The expression for the bound on the cost is obtained by appealing to Theorems 2.5 or 2.6.

■

The most interesting difference between the full information $\mathcal{H}_2$/Popov problem the the full-state feedback $\mathcal{H}_2/SN$ problem is the form of the optimal control. For the $\mathcal{H}_2/SN$ problem, the control is a linear function of the states alone. This is just what one would expect from an LQR problem or an $\mathcal{H}_\infty$ full information problem. In contrast, the control for the $\mathcal{H}_2$/Popov problem is a linear combination of both a state feedback term as well as a term involving the uncertainty input, w. This second, unexpected term arises because of the Popov stability multiplier, N. The uncertainty supply rate contains a term that is the product of N and the *derivative* of the state. This leads to terms involving products of w and u. It should be noted that this is not due to the fact that the FI system measures w. Rather, it is entirely due to the form of the Popov multipliers. Were we to examine a FI problem for the $\mathcal{H}_2/SN$ case, we would find the same result as was found in Theorem 5.1 for the FSFB problem.

Of course, because practical systems have noise in their measurements, it is not expected that the FI control can be achieved. We shall see later how an output feedback controller attempts to approximate this control.

5.2.2 $\mathcal{H}_2$/Popov Output Estimation

The motivation for the Output Estimation (OE) problem will become clear after the derivation of the output feedback controller. However, this sub-problem is interesting in its own right. Consider the system

$$\begin{Bmatrix} \dot{x} \\ v \\ y \end{Bmatrix} = \begin{bmatrix} A & B_d & B_r & B_u \\ C_v & 0 & D_{vr} & D_{vu} \\ C_y & D_{yd} & 0 & 0 \end{bmatrix} \begin{Bmatrix} x \\ d \\ r \\ u \end{Bmatrix}, \tag{5.77}$$

where D_{vu} is known to be full rank. The system has only two outputs, a performance variable, v, and a measurement, y. The system is subject to a white noise disturbance, d, and a bounded power disturbance, r. The performance signal contains feed-through contributions from both the r disturbance as well as the control, u. This problem differs from the OE problem for the $\mathcal{H}_2/SN$ system because of the effect of r on the performance metric.

The goal in this problem is to find a dynamic filter that minimizes the expected value of the quantity $(v^T E_u v - \zeta r^T r)$, where E_u is a known, positive definite matrix and ζ is a known, positive scalar. It is assumed that there is not enough knowledge regarding any feedback between v and r to apply a Popov multiplier to the system. The desired filter can be found by applying the following theorem.

Theorem 5.5 *For the system of equation (5.77), which is subject to a zero-mean, unit covariance, white-noise signal, d, and signal $r \in \mathcal{L}_2$, suppose the matrix*

$$\xi \triangleq \zeta I - D_{vr}^T D_{vr} \tag{5.78}$$

is positive definite. If there exist positive semidefinite matrices $X \in \mathcal{S}^n$ and $Y \in \mathcal{S}^n$ that are the stabilizing solutions to the following coupled Riccati equations

$$X\left(A-(YC_y^T+B_dD_{yd}^T\right)E_d^{-1}C_y)+\left(A-(YC_y^T+B_dD_{yd}^T)E_d^{-1}C_y\right)^T X$$
$$+(XB_r+C_v^TE_uD_{vr})\xi^{-1}(B_r^TX+D_{vr}^TE_uC_v)+C_v^TE_uC_v=0 \quad (5.79)$$

and

$$(A+B_r\xi^{-1}(B_r^TX+D_{vr}^TE_uC_v))Y+Y(A+B_r\xi^{-1}(B_r^TX+D_{vr}^TE_uC_v))^T$$
$$-(YC_y^T+B_dD_{yd}^T)E_d^{-1}(C_yY+D_{yd}B_d^T)+B_dB_d^T=0\,, \quad (5.80)$$

where $E_d \triangleq D_{yd}D_{yd}^T$*, then a filter exists such that the expected value of* $(v^TE_uv-\zeta r^Tr)$ *is bounded by*

$$\mathcal{J}_{\mathrm{OE}}=\mathrm{tr}\,(B_d-HD_{yd})^TX(B_d-HD_{yd})\,, \quad (5.81)$$

where $H=(YC_y^T+B_dD_{yd}^T)(E_d)^{-1}$*. Lastly, a realization for the filter which achieves this is given by*

$$\left[\begin{array}{c|c} A-HC_y-B_uD_{vu}^{-1}C_v & H \\ \hline -D_{vu}^{-1}C_v & 0 \end{array}\right]. \quad (5.82)$$

Proof. This proof will proceed exactly as the proof of Theorem 5.2 proceeded. There is essentially no difference between the two because we do not include a Popov multiplier in the description of the feedback loop from v to r; it is only described by bounded gain information.

As before, we will assume that the filter has a model-based observer structure. However, unlike before, there is now a feedthrough term from r to v. We assume the filter to have the form

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x}+B_uu+H(y-C_y\hat{x}) \\ u &= -D_{vu}^{-1}C_v\hat{x}\,, \end{aligned} \quad (5.83)$$

where $\hat{x}$ is the state estimate and H is an unknown filter gain. The observer does not account for the presence of the D_{vr} term explicitly. This information cannot be included because there is no information about the exogenous r signal other than the fact that it has bounded power. Defining the error coordinates, $\tilde{x} \triangleq x-\hat{x}$, the closed loop system with the filter can be written as

$$
\begin{aligned}
\dot{\tilde{x}} &= (A - HC_y)\tilde{x} + (B_d - HD_{yd})d + B_r r \\
v &= C_v\tilde{x} + D_{vr}r \, .
\end{aligned}
\tag{5.84}
$$

Following exactly the same procedure used in the proof of Theorem 5.2, we can use dissipation theory to derive a Riccati equation that allows us to bound the cost. If $X \in \mathcal{R}^n$ is a positive definite matrix satisfying

$$
(A-HC_y)^T X + X(A-HC_y) + (XB_r + C_v^T E_u D_{vr})\xi^{-1}(B_r^T X + D_{vr}^T E_u C_v) + C_v^T E_u C_v = 0,
\tag{5.85}
$$

then a bound on the cost is

$$
\text{tr } (B_d - HD_{yd})^T X (B_d - HD_{yd}) \, .
\tag{5.86}
$$

Equation (5.85) happens to be a standard $\mathcal{H}_\infty$ Riccati equation for the closed loop system. Thus, if $r \in \mathcal{L}_2$, the Riccati constraint would guarantee that the closed loop system achieved $\|v\|_2^2/\|r\|_2^2 < \zeta$. Note also that the cost function of equation (5.86) has the desired form of equation (5.81).

Next, we wish to minimize the bound on the cost, subject to the Riccati constraint. Thus, we form an augmented cost functional

$$
\begin{aligned}
\mathcal{J}_{\text{aug}} = \text{tr } (B_d - HD_{yd})^T X (B_d - HD_{yd}) + \text{tr } Y \Big[(A - HC_y)^T X + X(A - HC_y) \\
+ (XB_r + C_v^T E_u D_{vr})\xi^{-1}(B_r^T X + D_{vr}^T E_u C_v) + C_v^T E_u C_v \Big] \, ,
\end{aligned}
\tag{5.87}
$$

where the Lagrange multiplier, $Y \in \mathcal{R}^n$, is a symmetric matrix. Since this is a steady state bound and we are interested in time-invariant controllers, $\mathcal{J}_{\text{aug}}$ should be stationary with respect to the optimal gain, H^*. This leads us to two necessary conditions

$$
(A + B_r\xi^{-1}(B_r^T X + D_{vr}^T E_u C_v) - H^* C_y)Y + B_d B_d^T + H^* D_{yd} D_{yd}^T H^{*T}
$$

$$+Y(A+B_r\xi^{-1}(B_r^TX+D_{vr}^TE_uC_v)-H^*C_y)^T-H^*D_{yd}B_d^T-B_dD_{yd}^TH^{*T}=0 \quad (5.88)$$

and

$$H^*=(YC_y^T+B_dD_{yd}^T)(E_d)^{-1}, \qquad (5.89)$$

where $E_d \triangleq (D_{yd}D_{yd}^T)$. We note that E_d is always invertible since D_{yd} was assumed to be full row rank. Substituting the form for H^* into (5.85) and (5.88) leads to the desired conditions (5.79) and (5.80).

Finally, the equations describing the optimal filter can be obtained by substituting the definition for H^* into the assumed filter form in equation (5.83). ■

The OE problem we have just examined is essentially the same as the OE problem for the $\mathcal{H}_2$/*SN* system, since the implicit feedback loop from v to r has not been characterized using Popov multipliers. As a result, the solutions essentially have the same form. As before, the description of the optimal filter gain has been written in the same form as that which would typically be found for a Kalman filter. However, neither the filter nor the filter gain are the same as would be found in a Kalman filter, because the conditions that the matrix Y must fulfill are not those of a Kalman filter. The matrix Y must fulfill a pair of nonlinear equations, equations (5.79) and (5.80). These equations are referred to as coupled Riccati equations. For a fixed Y, equation (5.79) is a Riccati equation in terms of X. Similarly, for a fixed X, equation (5.80) is a Riccati equation in terms of Y. The coupled Riccati equations constitute a set of conditions that are sufficient to guarantee the existence of a solution to the OE problem.

5.2.3 $\mathcal{H}_2$/Popov Output Feedback

Now that we have discussed the two subsidiary problems that will be used to simplify the Output Feedback (OF) problem, we can give the main results of this section.

Theorem 5.6 *Consider the system of equation (3.15). Assume that we are given fixed stability multipliers, H and N. Define the following constant matrices*

$$\zeta \triangleq M_d H + H M_d - N C_z B_w - B_w^T C_z^T N \tag{5.90}$$

$$Z \triangleq \zeta + N C_z B_u E_u^{-1} B_u^T C_z^T N \tag{5.91}$$

$$\xi \triangleq \zeta - N C_z B_u E_u^{-1} B_u^T C_z^T N\,, \tag{5.92}$$

where $E_u \triangleq D_{eu}^T D_{eu}$. Furthermore, assume that both Z and ξ are positive definite. If a matrix $R \in \mathcal{S}^n$ exists such that it is the unique, positive semidefinite, stabilizing solution to the related full-state feedback Riccati equation

$$R\hat{A} + \hat{A}^T R + C_e^T C_e + \left\{R(B_w - B_u E_u^{-1} B_u^T C_z^T N) + (C_z^T H + \hat{A}^T C_z^T N)\right\}$$
$$\cdot Z^{-1}\left\{(HC_z + NC_z\hat{A}) + (B_w^T - NC_z B_u E_u^{-1} B_u^T)R\right\} - RB_u E_u^{-1} B_u^T R = 0\,, \tag{5.93}$$

where $\hat{A} \triangleq A + B_w M_1 C_z$, and if positive semidefinite matrices $X \in \mathcal{S}^n$ and $Y \in \mathcal{S}^n$ exist that are the stabilizing solutions to the related output estimation coupled Riccati equations

$$X\left(A_{\text{temp}} - (YC_y^T + B_d D_{yd}^T\right) E_d^{-1} C_y) + \left(A_{\text{temp}} - (YC_y^T + B_d D_{yd}^T) E_d^{-1} C_y\right)^T X$$
$$+(XB_w + C_v^T E_u D_{vr})\xi^{-1}(B_w^T X + D_{vr}^T E_u C_v) + C_v^T E_u C_v = 0 \tag{5.94}$$

and

$$(A_{\text{temp}} + B_w \xi^{-1}(B_w^T X + D_{vr}^T E_u C_v))Y + Y(A_{\text{temp}} + B_w \xi^{-1}(B_w^T X + D_{vr}^T E_u C_v))^T$$
$$-(YC_y^T + B_d D_{yd}^T) E_d^{-1}(C_y Y + D_{yd} B_d^T) + B_d B_d^T = 0\,, \tag{5.95}$$

where $E_d = (D_{yd} D_{yd}^T)$, and

$$A_{\text{temp}} = A + B_w \zeta^{-1} \mathcal{G} \tag{5.96}$$

$$C_v = E_u^{-1} B_u^T (R - C_z^T N M_1 C_z + C_z^T N \zeta^{-1} \mathcal{G}) \tag{5.97}$$

$$D_{vr} = E_u^{-1} B_u^T C_z^T N \tag{5.98}$$

$$\mathcal{G} = B_w^T R - B_w^T C_z^T N M_1 C_z + N C_z A + M_d H (M_1 + M_2) C_z \tag{5.99}$$

then, a controller given by

$$A_c = A_{\text{temp}} - B_c C_y + B_u C_c \tag{5.100}$$

$$B_c = (Y C_y^T + B_d D_{yd}^T)(D_{yd} D_{yd}^T)^{-1} \tag{5.101}$$

$$C_c = -(I + E_u^{-1} B_u^T C_z^T N \zeta^{-1} N C_z B_u)^{-1} C_v \tag{5.102}$$

will stabilize the system for all allowable Δ. *Furthermore, for all allowable* Δ, *the closed-loop performance of the system,* J, *will be bounded by*

$$\mathcal{J}_{\text{OF}} = \text{tr}\ \left\{ B_d^T R B_d + [B_d - B_c D_{yd}]^T X [B_d - B_c D_{yd}] \right\}. \tag{5.103}$$

Proof. The proof of this theorem uses the results of the FI and OE problems for the $\mathcal{H}_2$/Popov case. The proof will follow the same course as was developed for the $\mathcal{H}_2$/*SN* case. However, it differs because the Popov multipliers (H and N) are incorporated into the system dynamics differently than the manner in which the multiplier (σI) was incorporated in the $\mathcal{H}_2$/*SN* case.

First, as was done in the proof of Theorem 5.3, consider a candidate Lyapunov function for the $\mathcal{H}_2$/Popov system. At steady state, the expected value of the Lyapunov function is a constant, such that

$$\lim_{t\to\infty} \frac{d}{dt} E\{x^T R x\} = \frac{d}{dt} \lim_{t\to\infty} E\{x^T R x\} = 0 \tag{5.104}$$

$$= \lim_{t\to\infty} E\left\{ x^T (RA + A^T R) x + x^T R B_w w + w^T B_w^T R x + x^T R B_u u + u^T B_u^T R x \right\} + \text{tr}\ (B_d^T R B_d), \tag{5.105}$$

where the last equality comes from substituting in the system dynamics, and the trace terms arise due to the differentiation of the white noise signal, d. As an aside, note that equation (5.105) has the same form as equation (5.34).

We will now discuss a dissipation inequality for the system. The dissipation inequality involves terms that ensure that the closed loop system will be stable for any uncertainty that can be described using Popov multipliers. Also, the inequality contains terms that allow us to bound the $\mathcal{H}_2$ cost of the closed loop system. This dissipation inequality has already been derived in the proof of the FI theorem. It is repeated here for clarity:

$$x^T R(Ax + B_u u + B_w w) + (Ax + B_u u + B_w w)^T Rx + (x^T C_e^T + u^T D_{eu}^T)(C_e x + D_{eu} u)$$
$$+2(w - M_1 C_z x)^T \left[NC_z(Ax + B_u u + B_w w) + M_d H(M_2 C_z x - w)\right] \leq 0\,. \quad (5.106)$$

As was done in the proof for the output feedback problem in the $\mathcal{H}_2/SN$ case, we wish to examine this inequality under extremal conditions, when both the control and disturbance are extremal. Then, we wish to complete the square on the control and disturbance terms in the general dissipation inequality. This will reveal that the performance metric is bounded by the sum of the cost functions from the FI and OE problems.

We have already derived the form for the extremal control in the proof of the FI theorem. It is

$$u^* = E_u^{-1} B_u^T \left(-Rx + C_z^T N M_1 C_z x - C_z^T N w\right)\,. \quad (5.107)$$

This expression could have been found by differentiating equation (5.106) with respect to u. The expression for this extremal control can be simplified if we make the following definitions:

$$K_1 \triangleq B_u^T R \quad (5.108)$$
$$K_2 \triangleq B_u^T C_z^T N \quad (5.109)$$
$$K_3 \triangleq B_u^T C_z^T N M_1 C_z\,. \quad (5.110)$$

Therefore the extremal control can be written as

$$u^* = -E_u^{-1}\left((K_1 - K_3)x + K_2 w\right)\,. \quad (5.111)$$

At this point, the form for the worst-case (extremal) disturbance is not known. It will be needed to complete a square in the dissipation inequality. In the $\mathcal{H}_2/SN$ case, the worst-case disturbance was found to have essentially the same definition as the optimal control, with B_u replaced by B_w. (This is the same as is found in a pure $\mathcal{H}_\infty$ problem.) This will not be the case for the $\mathcal{H}_2$/Popov system. To find where the inequality is extremal with respect to the disturbance, w, we take the derivative of (5.106). Setting the resulting expression equal to zero reveals that

$$B_w^T R x + N C_z(Ax + B_u u + B_w w^*) + B_w^T C_z^T N(w^* - M_1 C_z x) \\ + M_d H(M_2 C_z x - w^*) + M_d H(M_1 C_z x - w^*) = 0 . \qquad (5.112)$$

Noting that H, N, M_1, M_2, and M_d are all diagonal matrices, we can rewrite this last expression to find that the worst-case disturbance is

$$\begin{aligned} w^* &= \zeta^{-1}\left[B_w^T R x + N C_z(Ax + B_u u) - B_w^T C_z^T N M_1 C_z x \right.\\ &\qquad \left. + M_d H(M_1 + M_2) C_z x\right] && (5.113)\\ &= \zeta^{-1}\left[\mathcal{G} x + K_2^T u\right] , && (5.114) \end{aligned}$$

where

$$\begin{aligned} \zeta &\triangleq M_d H + H M_d - N C_z B_w - B_w^T C_z^T N && (5.115)\\ \mathcal{G} &\triangleq B_w^T R - B_w^T C_z^T N M_1 C_z + N C_z A + M_d H(M_1 + M_2) C_z . && (5.116) \end{aligned}$$

We note that the definitions for ζ and $\mathcal{G}$ agree with equations (5.90) and (5.99), respectively. Here, in the $\mathcal{H}_2$/Popov case, just as the extremal control required knowledge of any disturbance, the worst case control makes use of the control input (not necessarily optimal). This occurs because the uncertainty supply rate in the dissipation analysis contains the product of the Popov multiplier N with the *derivative* of the state variable.

Having derived explicit expressions for both the extremal control and disturbance, we wish to examine inequality (5.106) with the control, u, and the disturbance, w, specified to be these functions. Replacing each occurrence of u and w by u^* and w^*, respectively, and substituting definitions (5.107) and (5.113) to the inequality yields a new, complicated inequality. After some algebra, this can be simplified to

$$\begin{aligned} x^T(RA + A^TR)x - 2x^TC_z^TM_1\left[NC_zAx + M_dHM_2C_zx\right] \\ +x^TC_e^TC_ex - u^{*T}E_uu^* + w^{*T}\zeta w^* \leq 0\,, \end{aligned} \tag{5.117}$$

where we have explicitly made use of the fact that $C_e^TD_{eu} = 0$. In order to relate the performance of the system with extremal inputs to a system with arbitrary inputs, we complete the square on the control and disturbance terms in equation (5.117) to obtain

$$\begin{aligned} &x^TR(Ax + B_uu + B_ww) + (Ax + B_uu + B_ww)^TRx + x^TC_e^TC_ex \\ &\quad +u^TE_uu + (w - w^*)^T\zeta(w - w^*) - (u - u^*)^TE_u(u - u^*) \\ &\quad +2(w - M_1C_zx)^T\left[NC_z(Ax + B_uu + B_ww) + M_dH(M_2C_zx - w)\right] \leq 0\,. \end{aligned} \tag{5.118}$$

This last inequality can be rearranged to reveal that

$$\begin{aligned} x^TR(Ax + B_uu + B_ww) + (Ax + B_uu + B_ww)^TRx \leq -x^TC_e^TC_ex - u^TE_uu \\ -2(w - M_1C_zx)^T\left[NC_z(Ax + B_uu + B_ww) + M_dH(M_2C_zx - w)\right] \\ -(w - w^*)^T\zeta(w - w^*) + (u - u^*)^TE_u(u - u^*)\,. \end{aligned} \tag{5.119}$$

At this point of the proof, we can return our attention to the general case, with non-extremal controls and disturbances. Equation (5.105) holds true in general. A comparison of this equation with inequality (5.119) demonstrates that

$$\begin{aligned} \lim_{t\to\infty} E\Big\{-x^TC_e^TC_ex - u^TE_uu - (w - w^*)^T\zeta(w - w^*) + (u - u^*)^TE_u(u - u^*) \\ -2(w - M_1C_zx)^T\left[NC_z(Ax + B_uu + B_ww) + M_dH(M_2C_zx - w)\right]\Big\} \end{aligned}$$

$$+\operatorname{tr} B_d^T R B_d \geq 0\,. \qquad (5.120)$$

This latest inequality can be simplified if we make the following definitions

$$v \triangleq u - u^* = u - E_u^{-1} B_u^T \left(-Rx + C_z^T N M_1 C_z x - C_z^T N w\right) \qquad (5.121)$$

$$r \triangleq w - w^* = w - \zeta^{-1}\left[B_w^T R x + N C_z (Ax + B_u u) - B_w^T C_z^T N M_1 C_z x \right.$$
$$\left. + M_d H (M_1 + M_2) C_z x\right]\,, \qquad (5.122)$$

such that v has been defined to be the difference between the actual control and the desired, optimal control. Likewise, r has been defined to be the difference between the actual disturbance, w, and the worst-case disturbance. This allows us to rewrite inequality (5.120) as

$$-\lim_{t\to\infty} E\left\{2(w - M_1 C_z x)^T \left[N C_z (Ax + B_u u + B_w w) + M_d H (M_2 C_z x - w)\right]\right\}$$
$$-\lim_{t\to\infty} E\left\{x^T C_e^T C_e x + u^T E_u u\right\} + \lim_{t\to\infty} E\left\{v^T E_u v - r^T \zeta r\right\} + \operatorname{tr} B_d^T R B_d \geq 0\,. \qquad (5.123)$$

Next, recalling the definition of the cost function, we see that the performance metric that we would like to minimize can be written as

$$J = \lim_{t\to\infty} E\{e^T e\} = \lim_{t\to\infty} E\{x^T C_e^T C_e x + u^T E_u u\}\,. \qquad (5.124)$$

Furthermore, we recall that the uncertainty is described by Popov multipliers, so that

$$\lim_{t\to\infty} E\left\{2(w - M_1 C_z x)^T \left[N C_z (Ax + B_u u + B_w w) + M_d H (M_2 C_z x - w)\right]\right\} \geq 0\,. \qquad (5.125)$$

Combining these facts with (5.123) reveals that

$$J \leq \operatorname{tr} B_d^T R B_d + \lim_{t\to\infty} E\{v^T E_u v - r^T \zeta r\}\,. \qquad (5.126)$$

Thus, we see that the performance of the closed loop system will be bounded by the sum of two expressions. The first expression is equal to the bound on the cost for the related robust full information (FI) system without the additional quantity

$$\text{tr}\ \left[B_d^T C_z^T (M_2 - M_1) N C_z B_d\right]\ ,$$

which is always positive (see Theorem 5.4). The second expression is equal to the cost for a related output estimation (OE) problem. This estimation problem is, essentially, to estimate the system states and the incoming disturbance, w. If we knew both x and w, then we could set the control to be equal to the optimal control, and then $v^T E_u v$ would be identically zero. Thus, it is an estimation problem. The $(-r^T \zeta r)$ term actually helps to decrease the cost, because it accounts for the amount by which the uncertainty differs from the worst-case uncertainty.

We wish to bound the second term on the right hand side of (5.126). To do this, we shall use the results from the OE problem discussed in Theorem 5.5. First, we transform the system to have the form specified for an OE problem. The original system dynamics are given in equation (3.15). They can be rewritten as

$$\begin{aligned} \dot{x} &= Ax + B_d d + B_w w + B_u u + B_w w^* - B_w w^* \\ &= Ax + B_d d + B_w w + B_u u + B_w \zeta^{-1}(\mathcal{G}x + K_2^T u) - B_w \zeta^{-1}(\mathcal{G}x + K_2^T u) \\ &= A_{\text{temp}} x + B_d d + B_r r + B_{u_{\text{temp}}} u \end{aligned} \tag{5.127}$$

where

$$A_{\text{temp}} \triangleq A + B_w \zeta^{-1} \mathcal{G} \tag{5.128}$$

$$B_r \triangleq B_w \tag{5.129}$$

$$B_{u_{\text{temp}}} \triangleq B_u + B_w \zeta^{-1} N C_z B_u \tag{5.130}$$

$$= B_u + B_w \zeta^{-1} K_2^T\ . \tag{5.131}$$

Furthermore, the new output of interest, v, can be rewritten as

$$v = u + E_u^{-1} B_u^T \left(-Rx + C_z^T N M_1 C_z x - C_z^T N w\right)$$

$$+E_u^{-1}B_u^TC_z^TNw^* - E_u^{-1}B_u^TC_z^TNw^*$$

$$= C_vx + D_{vr}r + D_{vu}u\,, \tag{5.132}$$

where

$$C_v = E_u^{-1}B_u^T(R - C_z^TNM_1C_z + C_z^TN\zeta^{-1}\mathcal{G}) \tag{5.133}$$

$$= E_u^{-1}(K_1 - K_3 + K_2\zeta^{-1}\mathcal{G}) \tag{5.134}$$

$$D_{vr} = E_u^{-1}B_u^TC_z^TN = E_u^{-1}K_2 \tag{5.135}$$

$$D_{vu} = I + E_u^{-1}B_u^TC_z^TN\zeta^{-1}NC_zB_u \tag{5.136}$$

$$= I + E_u^{-1}K_2\zeta^{-1}K_2^T\,. \tag{5.137}$$

Together, these definitions allow us to rewrite the open loop system as

$$\begin{Bmatrix} \dot{x} \\ v \\ y \end{Bmatrix} = \begin{bmatrix} A_{\text{temp}} & B_d & B_r & B_{u_{\text{temp}}} \\ C_v & 0 & D_{vr} & D_{vu} \\ C_y & D_{yd} & 0 & 0 \end{bmatrix} \begin{Bmatrix} x \\ d \\ r \\ u \end{Bmatrix}. \tag{5.138}$$

Comparing this system to the OE system of Theorem 5.5, we see that they have exactly the same form. Also, comparing this system to the original system of equation (3.15), we see the new system retains three inputs, but has only two outputs. The original performance variable is no longer needed. Since the goal is to minimize $E\{v^TE_uv - r^T\zeta r\}$, the e output does not need to be considered. Lastly, we note that D_{vu} is invertible, as was required by the OE problem. The matrix is, in fact, positive definite, since it is equal to the sum of a positive definite matrix, I, with a positive semidefinite matrix, $E_u^{-1}K_2\zeta^{-1}K_2^T$. Clearly, this is now an output estimation problem with performance variable v.

Since this is an OE problem, Theorem 5.5 is used to derive a filter which will minimize the bound on the cost. Equations (5.94) and (5.95) are found from equations (5.79) and (5.80) by replacing occurrences of A with A_{temp}. Note that definitions for the C_v, D_{vr}, and D_{vu} matrices, which are needed in Theorem 5.5, have also been specified. Furthermore, the definitions for the controller parameters proposed in equations (5.100)– (5.102) can be derived by replacing A by A_{temp} and substituting in the appropriate definitions to the filter of equation (5.82).

Finally, we are in a position to derive the separation-based cost function of equation (5.103). As already noted, equation (5.126) demonstrates that the $\mathcal{H}_2$ cost can be bounded by the sum of two terms. The first term can be overbounded by solving a full information control problem. The second term can be bounded by solving an output estimation problem. Thus a bound for the closed loop system cost can be given as

$$
\begin{aligned}
J \leq \mathcal{J}_{\text{OF}} &\triangleq \operatorname{tr} B_d^T R B_d + \mathcal{J}_{\text{OE}} && (5.139)\\
&= \operatorname{tr} B_d^T R B_d + \operatorname{tr} \left[B_d - B_c D_{yd}\right]^T X \left[B_d - B_c D_{yd}\right] && (5.140)\\
&\leq \mathcal{J}_{\text{FI}} + \mathcal{J}_{\text{OE}} \, ,
\end{aligned}
$$

where $\mathcal{J}_{\text{FI}}$ and $\mathcal{J}_{\text{OE}}$ were defined in equations (5.69) and (5.81), respectively. The controller synthesis problem has effectively been separated into two smaller problems: one problem to find the appropriate feedback gain and one problem to find an optimal filter. ■

Theorem 5.6 provides a set of conditions that are sufficient, but not necessary, to guarantee the existence of an $\mathcal{H}_2$/Popov controller. Given the assumptions mentioned in the theorem, existence is guaranteed if we can solve equations (5.93)–(5.95). Equation (5.93) is seen to be a Riccati equation in terms of R alone. In contrast, equations (5.94) and (5.95) are functions of three variables, R, X, and Y (R appears in the definitions of A_{temp} and C_v). In practice, then, one can solve equation (5.93) for R first, without regard to the other two equations. Then, given R, one can solve equations (5.94) and (5.95) for X and Y. These last two equations are coupled Riccati equations. Given fixed R and Y, equation (5.94) is a Riccati equation in terms of X. Likewise, given fixed R and X, equation (5.95) is a Riccati equation for Y.

The proof of Theorem 5.6 follows the same course as the proof of Theorem 5.3 for the $\mathcal{H}_2/SN$ system. No new mathematical machinery is developed for the $\mathcal{H}_2$/Popov case. The proofs are fundamentally the same, but differ because of the way that the stability multipliers affect the dynamics. If the separation principle for the system with Popov multipliers had not been demonstrated, one might mistakenly believe that the separation principle was restricted to systems with constant multipliers (as in the $\mathcal{H}_2/SN$ case). In fact, the exact form of the multipliers should not be a limiting factor. The choice of multipliers certainly does affect the controller dynamics and associated sufficiency conditions. However, it is not fundamental to the existence of a separation principle for a quadratic bound on a robust $\mathcal{H}_2$ cost. It should be possible to derive similar results for other finite dimensional multipliers. Strictly proper multipliers should allow one to avoid some of the anomalies that appear in the $\mathcal{H}_2$/Popov case, such as an optimal control that depends on knowing the disturbance.

To understand why the exact form of the separation principle should hold independently of the type of multiplier, consider that a sector transformation can be used to change the explicit form of the multiplier. Sector transformations do not affect the input/output characteristics of the system intact; thus, they should not affect whether or not this separation principle will hold. In the case of a system with Popov multipliers, a sector transformation will make the transformed system appear to only have a constant stability multiplier. However, the Popov multiplier's unusual non-causal dynamics also makes the transformed system dynamics fail to have the form required by the $\mathcal{H}_2/SN$ problem of Theorem 5.3. The transformed system contains a direct feedthrough term from w to z. One should be able to rederive the separation principle for this system with a constant multiplier.

As in the case of the $\mathcal{H}_2/SN$ synthesis, what makes the separation principle interesting for the $\mathcal{H}_2$/Popov synthesis is the size and form of the new synthesis problem. After solving for R with a standard Riccati equation solver, we are left with two size $n \times n$ variables, X and Y. In contrast, the LMI solution requires one to solve for a size $2n \times 2n$ and a size $n \times n$ matrix. Furthermore, the separation-based formulation uses coupled Riccati equations. Note that standard Riccati equations of size n require solution times proportional to $(2n)^3$, while our experience with convex optimization codes indicate that these perform on the order of between n^4 and n^5. Therefore, if one can solve the coupled Riccati equations by, say, iterating on the solutions to standard Riccati equations, then this could be faster than solving the corresponding LMI problem.

In addition, it should be mentioned that a separation-based synthesis routine has the potential to require much less memory than a comparable LMI-based routine. The LMI-based synthesis technique required us to minimize a linear cost function subject to the three LMI constraints. However, our experience with LMILab indicates that current codes are not usable for higher-order systems. The difficulty is that a solution technique must determine the Jacobians and, possibly, Hessians of the constraints. For a practical problem, this requires too much memory to be evaluated. In contrast, the sufficiency conditions in the separation principle (see Theorem 5.6) do not require the explicit minimization of any cost function. A solution of the coupled Riccati equations automatically yields the optimal controller.

Before leaving our discussion of Theorem 5.6, we should mention how it enhances our understanding of the LMIs in Theorem 4.3. Specifically, we wish to give an interpretation for LMI (4.33). Examining the Schur complements of this LMI leads to two new LMIs

$$\begin{aligned} Q_{11}\hat{A}^T + \hat{A}Q_{11} + \left[Q_{11}(C_z^T H + \hat{A}^T C_z^T N) + B_w - B_u E_u^{-1} B_u^T C_z^T N\right] \\ \cdot Z^{-1}\left[(HC_z + NC_z\hat{A})Q_{11} + B_w^T - NC_z B_u E_u^{-1} B_u^T\right] \\ -B_u^T (D_{eu}^T D_{eu})^{-1} B_u + Q_{11} C_e^T C_e Q_{11} \;\leq\; 0 \end{aligned} \tag{5.141}$$

and

$$-Z \leq 0 \tag{5.142}$$

where Z is as defined in equation (5.91). The second of these LMIs requires that $-Z$ be negative semidefinite. This is actually less strict than was assumed in the separation principle of Theorem 5.6, but has much the same effect. Of greater interest is the first of these LMIs. If we multiply this LMI on the left and right by Q_{11}^{-1}, we obtain

$$\begin{aligned} Q_{11}^{-1}\hat{A} + \hat{A}^T Q_{11}^{-1} + \left\{Q_{11}^{-1}(B_w - B_u E_u^{-1} B_u^T C_z^T N) + (C_z^T H + \hat{A}^T C_z^T N)\right\} \\ \cdot Z^{-1}\left\{(HC_z + NC_z\hat{A}) + (B_w^T - NC_z B_u E_u^{-1} B_u^T)Q_{11}^{-1}\right\} \\ + C_e^T C_e - Q_{11}^{-1} B_u E_u^{-1} B_u^T Q_{11}^{-1} \leq 0\,. \end{aligned} \tag{5.143}$$

After rearranging this inequality, it can be seen that this is an inequality form of the FI Riccati equation for R in Theorem 5.6. Thus, the second LMI constraint of Theorem 4.3 can be thought of as a constraint on the feedback gain of the controller. Of course, for any given solution of the LMI-based control problem, it need not be the case that Q_{11}^{-1} is equal to the solution of the FI Riccati equation, R. This would only happen in the case where the correct realization was chosen for $\bar{B}_c$ in the solution of the LMI problem.

5.3 Suboptimal Controllers

It has not been demonstrated that solving the FI (or FSFB) problem and the OE problem will yield the same controller that would have been obtained using the LMI-based synthesis technique. However, the controllers derived using the separation principle are optimal in the sense that they minimize a bound on the $\mathcal{H}_2$ cost of the system, $\mathcal{J}_{\mathrm{OF}}$. Theorems 5.3 and 5.6 provide a set of conditions that are sufficient to guarantee that a robust controller exists. Note that in the separation-based synthesis method, an explicit minimization is not performed on $\mathcal{J}_{\mathrm{OF}}$. We shall now discuss how the previous results point to a means to synthesize suboptimal controllers, *i.e.*, controllers that satisfy the robustness requirements of the system but do not necessarily minimize the bound $\mathcal{J}_{\mathrm{OF}}$. These results will be discussed in terms of the $\mathcal{H}_2$/Popov controller, but they apply to the case with the $\mathcal{H}_2/SN$ controller as well.

Assume that we can satisfy the assumptions of Theorem 5.6 and that R is a stabilizing solution to the FI Riccati equation. Then, according to the theorem, finding solutions for the two coupled Riccati equations will yield the optimal controller. What would we obtain if we satisfied conditions which were less stringent than the coupled Riccati equations? In particular, what would we obtain if equation (5.94) were a matrix inequality rather than an equality?

Recall that the derivation of the $\mathcal{H}_2$/Popov separation principle relied upon prior knowledge of the OE filter discussed in Theorem 5.5. It can be shown that a Riccati inequality with the left hand side of equation (5.94) would arise if one were to derive equation (5.85) using dissipation theory, rather than simply appealing to standard $\mathcal{H}_\infty$ theory. A dissipation derivation would quickly show that, because ξ is positive definite, finding a matrix that satisfies the inequality form of (5.94) will guarantee that a controller exists that can satisfy the closed loop robustness requirements.

Note that certain properties of Riccati equations are will established. In particular, say that P that is the unique stabilizing solution to a Riccati equation, $\text{Ric}(P) = 0$. Given a matrix $\acute{P}$ that satisfies the related Riccati inequality, $\text{Ric}(\acute{P}) \leq 0$, one can show that $P \leq \acute{P}$.[81]

Consider the Riccati inequality formed by requiring that the left hand side of equation (5.94) be negative definite. Suppose that positive definite matrices X and Y are such that this inequality is satisfied. Furthermore, suppose that equation (5.95) is satisfied; this implies that Y is a valid Lagrange multiplier. Then the controller formed by substituting X and Y to the definitions in equations (5.100)–(5.102) will suffice to satisfy the necessary robustness requirements. Furthermore, the closed loop cost can be bounded by substituting the given matrices to equation (5.103). The controller is considered suboptimal because controllers that achieve tighter bounds on the $\mathcal{H}_2$ cost must exist.

Having a set of conditions that are sufficient to guarantee the existence of a suboptimal controller could be useful. It is possible that the suboptimal controller could be easier to solve for than the optimal controller. Given a fixed Y matrix, the Riccati inequality for X can be converted into an LMI — it is a convex constraint. Thus, for a fixed Y, the set of feasible X is a convex set. It is likely that a solution technique could take advantage of this fact, even if only indirectly. For instance, even if the

LMI constraint requires too much memory to use in an optimization routine, it is conceivable that one might be able find a member of the feasible set more easily than one can solve the Riccati equation. If this is the case, then, after finding a feasible X, one can evaluate the bound on the $\mathcal{H}_2$ cost. If the cost is deemed acceptable, then it is unnecessary to go on to find the exact solution that minimizes $\mathcal{J}_{\mathrm{OF}}$.

Chapter 6

Design of Robust $\mathcal{H}_2$ Controllers

Previously, we have examined several existing methods to analyze the $\mathcal{H}_2$ performance of an uncertain closed loop system, and we have derived new methods to synthesize robust $\mathcal{H}_2$ controllers for an uncertain plant. This chapter combines the available analysis and synthesis tools to form a complete robust $\mathcal{H}_2$ controller design methodology. The design technique will rely on what is commonly called a "$D-K$ iteration." In the D, or analysis, step of the iteration, the controller is held fixed and the performance of the system is analyzed (stability multipliers are determined). Conversely, in the K, or synthesis, step of the iteration, a new controller is synthesized for a given set of stability multipliers. The great attraction of using a $D - K$ iteration is its relative ease of implementation and use. By dividing the design process into separate analysis and synthesis steps, the $D - K$ iteration allows us to directly employ our already developed analysis and synthesis tools in a full design process. Furthermore, as we shall demonstrate, it is easier to initialize a $D-K$ iteration than a methodology that allows both the controller and the stability multipliers to vary simultaneously.

The theoretical foundations behind two different controller synthesis methods were presented in Chapters 4 and 5. In contrast, the next section of this chapter will focus on practical issues related to synthesis algorithms, particularly memory usage and time of solution. A synthesis algorithm based on the separation principle of Chapter 5 will be shown to be effective and attractive for use in a $D - K$ iteration.

The second section of the chapter focuses on the implementation of a robust analysis routine. A practical analysis routine is proposed for use with higher-order systems.

Finally, in the last section of the chapter, the algorithms developed for analysis and synthesis are combined to form a practical controller design routine. The design routine uses a $D - K$ iteration and appears to be useful for high order systems. The design technique is demonstrated on two structural control design examples.

6.1 Implementation of a Synthesis Routine

Two competing methods have been proposed to synthesize robust $\mathcal{H}_2$ controllers. In Chapter 4, it was shown that the controllers could be synthesized by minimizing a linear cost function subject to LMI constraints. This is known as an eigenvalue problem (EVP) problem. EVPs are convex and can be solved in polynomial time. Alternatively, in Chapter 5, a synthesis method was derived based on a separation principle. This synthesis method required the solution of a full information Riccati equation followed by the solution of an output estimation (OE) problem. The OE problem was solved by finding the solution to a pair of coupled Riccati equations.

As discussed in Chapter 5, the reason for investigating synthesis via the separation principle is that it potentially offers several advantages over an LMI-based synthesis. The first of these is decreased memory usage. The outer dimensions of the LMI constraints are greater than the dimensions of the Riccati equations in the separation principle. Furthermore, the number of variables in the LMI-based formulation is greater than the number of variables in a separation-based formulation. The second potential advantage of a separation-based technique is speed. If the OE problem can be solved by successively solving standard Riccati equations, then the separation-based technique should approach solution times proportional to $(2n)^3$ (for a problem with n states) rather than the n^4 or n^5 times typically found with EVP solvers. Both of these advantages become more and more significant as problem size increases. The final reason to investigate the separation-based synthesis is that it gives us more insight into the makeup of the robust controllers.

This section concentrates on the implementation of the separation-based synthesis technique. It should be noted that because the LMI-based methodology is an EVP, its implementation is straightforward. Off-the-shelf codes, such as the LMI Control Toolbox (LMILab) for Matlab,[32,30,31] exist that will solve these problems. Using LMILab, we have been able to implement synthesis routines based on Theorems 4.1 and 4.3 for plants with up to approximately 16 states on a Sun SPARC-20 workstation. Synthesis for larger systems has not proven possible because the stated software requires excessive amounts of computer memory. Some of the memory overhead in LMILab is due to the fact that LMILab is a general-purpose LMI solver. Thus, large amounts of memory are required to, for instance, merely describe the LMI constraints. However, it is believed that the major memory penalty occurs because the EVP algorithm must try to approximate the Jacobians and, possibly, Hessians of the cost function. It is believed that more efficient coding of an EVP algorithm (outside Matlab) would allow a marginal increase in the number of states that can be accommodated. This option was not pursued because writing specialized LMI code is not a practical option for a designer. Furthermore, even if the memory problem is overcome, any EVP algorithm is still expected to require solution times on the order of between n^4 and n^5, which will be too slow for large systems.

For greater details on the implementation of an LMI-based synthesis technique, the reader is referred to either Ref. 82 or the thesis by Livadas.[57] These references present a complete $D-K$ design using LMIs for two example systems with 4 and 8 states. Also, it should be noted that when a system is small enough such that one can perform an LMI-based synthesis, the resulting solution can be used to serve as a check for any controller derived using a separation-based technique.

We turn now to the topic of finding a means to efficiently synthesize a controller using the separation principle. For a system with gain-bounded uncertainties, the separation-based synthesis method is found in Theorem 5.3. To simplify the discussion, we will concentrate only on the $\mathcal{H}_2/SN$ case in this section. Results developed here will transfer to the $\mathcal{H}_2$/Popov case (and Theorem 5.6) by analogy.

To synthesize the controller, we must solve equations (5.26), (5.27), and (5.28).

As discussed in Chapter 5, equation (5.26) is a Riccati equation in terms of R alone. Therefore, it can be solved independently of the other two equations. We will presume throughout the rest of this section that we have found a solution for R, using equation (5.26). Now, we must find an effective method to solve the remaining equations, the coupled Riccati equations. These equations originate from an OE problem statement. We next present several inefficient or unsuccessful methods that have been investigated to solve the OE problem. Then, in Section 6.1.2, we introduce the Control Gain Iteration, which appears to be the most effective tool to solve the OE problem.

6.1.1 Preliminary Attempts to Solve the OE Problem

The coupled Riccati equations, equations (5.27) and (5.28), are a pair of nonlinear, coupled, matrix equations for the unknown, size $n \times n$, matrices X and Y. Because they possess bilinear coupling terms, equations (5.27) and (5.28) are *not* Riccati equations. However, for a fixed Y, equation (5.27) is a Riccati equation for X, and the converse is true for Y and equation (5.28).

In principle, because the OE coupled Riccati equations are only of size n, we expect that their solution should pose no serious memory problems, even for relatively large systems. The important question is whether or not these equations can be solved quickly.

The structure of the coupled Riccati equations influences whether or not they can be solved using standard Riccati equation solution techniques. In equation (5.28), the coefficient of the quadratic Y term is *negative* semi-definite. Given the fact that (A, C_y) was assumed to be detectable, then, for a fixed X, this equation always has a solution. It is akin to a typical $\mathcal{H}_2$ Riccati equation. In contrast, the coefficient for the quadratic X term in equation (5.27) is *positive* semi-definite. This implies that, given a fixed Y, there is no guarantee that the equation will have a stabilizing solution, even if (A, B_w) is controllable. Essentially, the equation has a form similar to an $\mathcal{H}_\infty$ Riccati equation.

The most straightforward method for trying to solve the coupled equations is to iterate between solutions of coupled Riccati equations, *i.e.*, to fix $X = X_k$ and solve equation (5.28) for Y_{k+1}, then to fix $Y = Y_{k+1}$ and solve equation (5.27) for X_{k+2}. The iteration is repeated until the solutions appear to converge. We refer to this procedure as the *standard iteration.* Note that a set of coupled Riccati equations that are similar to ours can be written for the solution to a mixed $\mathcal{H}_2/\mathcal{H}_\infty$ problem[5] or an optimal projection problem.[51] In Ref. 5, the standard iteration is proposed as a means to solve them.

The standard iteration was tested extensively on an 8-state example problem. This problem is detailed in Appendix A. Solutions for the problem were compared to the controllers derived from the LMI-based methodology. When controllers from a separation-based approach were obtained, they were identical to the controllers derived via the LMIs.

Unfortunately, testing reveals three difficulties with the standard iteration. First, it can be difficult to come up with a useful initial condition to start the iteration. While one can try $X_0 = 0$ and solve for Y_0, there is no guarantee that there will then be a stabilizing solution for X_1. This identifies the second problem with the iteration – that equation (5.27) does not necessarily have a stabilizing solution. Often, one may be able to start the iteration, and the procedure will iterate for some time, but eventually Y_k enters a region where there is no solution for X_{k+1}. These problems are due to the sign of the quadratic term in the equation, as mentioned above.

The two previous problems might seem to be manageable if one could start the iteration near to the solution. However, experimental evidence indicates that even this is not adequate. The third problem is that at low levels of γ (at high levels of uncertainty, *i.e.*, for γ near $\gamma_{\min}$) the standard iteration is divergent. For high levels of γ the iteration will converge reliably, even with poor initial conditions. However, at low levels of γ, the iteration does not converge. Significantly, even if one starts the iteration at a point obtained by solving the LMI-based synthesis problem, the iteration diverges. The finite precision of any Riccati equation solver ensures that the initial point will be different from the true stationary point. The iteration typically displays an exponential divergence in these cases.

Because these are matrix, rather than scalar equations, there does not seem to be a way to determine whether or not an iteration is stable at a given point. Therefore, we are unable to predict whether any iterative method will work at a given level of γ.

Leaving aside the standard iteration, we can examine a second method that is commonly proposed for use with coupled Riccati equations, namely a homotopic continuation method.[14,16,15,69] The central idea behind a homotopy algorithm is to first solve a degenerate form of the required problem. The degenerate problem is chosen because it has a known solution. The problem is then continuously deformed (by changing whatever parameters made the problem degenerate) until it becomes the required problem. By following the solution as it evolves along the deformation trajectory, one can arrive at the desired solution. In particular, for the OE coupled Riccati equations, we might use an initial degenerate problem consisting of the original system but with $C_z = 0$ and $B_w = 0$. We know from Chapter 5 that the solution to this problem is the LQG controller. Then, the homotopy parameters C_z and B_w can be incrementally increased in size until their desired values are reached. Another choice for the homotopy parameter might be γ^2.

The problem with a homotopy method is that at each increment, the gradient and then the trajectory with respect to the homotopy parameter(s) must be calculated for the unknown variables. In our case the unknowns are X and Y. If γ^2 is chosen as the homotopy parameter, calculating these gradients requires the solution of four linear matrix equations, each with approximately n^2 unknowns. If C_z and B_w are chosen to vary independently, then the gradient calculation becomes even more complicated. Solving the equations is theoretically possible, but this method is not computationally attractive for larger systems. Thus, this method seems unlikely to offer any of the desired benefits over an LMI-based synthesis routine. Furthermore, our experience indicates that the solution can vary nonlinearly, or even discontinuously, with the homotopy parameter. This complicates the algorithm by requiring it to make judicious choices in step size and other practical matters at each step. For these reasons, homotopy is not pursued for the OE problem.

It is also possible that a solution to the OE problem could be found by developing some sort of Hamiltonian system that captures the dynamics implied by equations (5.27) and (5.28). This is, of course, how steady state Riccati equations are typically solved — by forming the associated Hamiltonian matrix and performing an eigen-decomposition of it. Were the OE equations not coupled, then we could form a large (size $4n$ rather than $2n$) Hamiltonian system for them. Of course, it is the coupling that makes this problem interesting, and it is the coupling that prevents us from forming a Hamiltonian system. No similarity transform has been found that can decouple the problem such that it can be solved using an eigen-decomposition.

A fourth, relatively successful, method to try and solve the OE problem comes from using a Shanks transformation to modify the standard iteration. The Shanks transformation is numerical procedure that is designed to find approximate solutions to slowly converging or diverging series.[3] The transform (which is nonlinear) is performed after every third iteration. The procedure for the transformation is

$$X_n \Longleftarrow S(X_n) = \frac{X_{n+1}X_{n-1} - X_n^2}{X_{n+1} + X_{n-1} - 2X_n} \, .$$

When it works, the procedure removes the effects of the dominant eigenvalue from the convergence of the iteration. The main difficulty with the transformation is that, as one gets close to the solution, the differences between each iteration become negligible, and the transformation becomes numerically unreliable.

For the coupled Riccati equation iteration, the Shanks transformation was performed element-by-element on the X variable. It was tested on the previously mentioned problem with 8 states and demonstrated some very positive effects. For some levels of γ that were previously shown to be divergent using the standard iteration, if the iteration was started near the solution, then the Shanks transformation stabilized the iteration and drove it toward the solution. At high levels of γ, cases run with the Shanks transformation converged *faster* than the standard iteration alone. The shortcoming of the method was that it did not affect one's ability to start the iteration. At low levels of γ, initial conditions which seemed reasonable did not take three steps such that one could perform the transformation. Thus, the Shanks transformation was deemed too unreliable for general usage.

A fifth avenue of research focused on trying to make the method of solving for X more robust. One method that was investigated was to fix Y and then solve not equation (5.27), but, essentially, the sum of equations (5.28) and (5.27) for X. (The substitute equation was actually $(5.27) + \alpha Y^{-1}(5.28)Y^{-1} = 0$, for some scalar parameter α). This affected the stability of the coefficient of the linear term in the equation. While in certain circumstances this allowed the routine to find solutions for X when there would be none for the original iteration, this method did not work reliably. However, this approach led to the development of the control gain iteration, which proved to be the most reliable method available to solve the OE problem.

6.1.2 The Control Gain Iteration for Coupled Riccati Equations

Three problems have been encountered when using the standard iteration to solve the OE problem: it is difficult to find valid initial conditions for the iteration, the iteration stops when equation (5.27) fails to have a stabilizing solution, and the iteration becomes more divergent as the uncertainty size increases. The control gain (CG) iteration is a modified version of the standard iteration. It has been designed to eliminate the first two of the stated problems. It also seems to be effective at alleviating the third problem. The control gain iteration and a modified version of the CG iteration are presented below. An example synthesis problem is worked to demonstrate the effectiveness of the CG iteration versus the standard iteration and an LMI-based synthesis.

The CG iteration is essentially a perturbation method, and can also be thought of as a continuation method for X. At iteration k, the procedure is to solve equation (5.28) for Y_k, at a fixed X_k. Then, X_{k+1} is taken to be the stabilizing solution of the following modified Riccati equation

$$
\begin{aligned}
X_{k+1} \left(A + B_w B_w^T R - (Y_k C_y^T + B_d D_{yd}^T) E_d^{-1} C_y + B_w B_w^T X_k\right) \\
+ \left(A + B_w B_w^T R - (Y_k C_y^T + B_d D_{yd}^T) E_d^{-1} C_y + B_w B_w^T X_k\right)^T X_{k+1} \\
- X_{k+1} B_w B_w^T X_{k+1} + R B_u B_u^T R = 0\,. \qquad (6.1)
\end{aligned}
$$

Assuming the iteration has not converged, the procedure loops back and finds Y_{k+1}, and so on. Note that if it is evaluated at a stationary point, *i.e.*, if $X_{k+1} = X_k$, then equation (6.1) is exactly the same as the original equation (5.27).

The CG iteration is effective because the modified Riccati equation has a negative coefficient for the quadratic X_{k+1} term. Thus, this equation has the form of an $\mathcal{H}_2$, rather than an $\mathcal{H}_\infty$, equation. Then, according to standard Riccati equation theory,[23] as long as the linear coefficient of X_{k+1} is stabilizable with respect to B_w, at *any* iteration there will be a stabilizing X_{k+1} to solve equation (6.1). This eliminates the second problem found for the standard iteration. In fact, note that the modified dynamics matrix

$$
A + B_w B_w^T R - (Y_k C_y^T + B_d D_{yd}^T) E_d^{-1} C_y + B_w B_w^T X_k \qquad (6.2)
$$

can be assumed to be stable at every step. This is because the matrix Y_k, which was selected at the beginning of the step as the solution to equation (5.28), was chosen to stabilize the above quantity.

The key point that makes this iteration akin to a perturbation method is that the iteration *always* has a solution at the initial step. A valid initial condition is $X_{k=0} = X = 0$. Even at this point, the iteration will find a $\delta X_{k=1}$ such that $X_{k=1} = X_{k=0} + \delta X_{k=1}$ is a stabilizing solution of equation (6.1). This eliminates the first problem found for the standard iteration. The control gain iteration takes its name from the fact that the resulting X effectively yields the optimal stabilizing control gain for the dynamics matrix given in equation (6.2).

Before presenting some results obtained with the CG iteration, we present a further enhancement to the iteration. This iteration is referred to as the "Control Gain Iteration with Relaxation," or CGR iteration. It is motivated by the fact that, in practice, the standard iteration will head toward values of X that are unbounded. Equations (5.27) and (5.28) can be viewed as antagonists in a game. Equation (5.28) seeks to find the optimal estimator for the given X. In contrast, equation (5.27) seeks to find the worst-case disturbance that will damage the performance for a given Y. The CGR iteration seeks to allow the estimator solution (Y) to "catch up" with the uncertainty solution (X) by effectively giving the estimator forewarning of which way the X solution might go. Consider a slightly modified version of equation (6.1),

$$\begin{aligned} X_{k+1}\left(A + B_w B_w^T R - (Y_k C_y^T + B_d D_{yd}^T) E_d^{-1} C_y + \alpha B_w B_w^T X_k\right) \\ + \left(A + B_w B_w^T R - (Y_k C_y^T + B_d D_{yd}^T) E_d^{-1} C_y + \alpha B_w B_w^T X_k\right)^T X_{k+1} \\ + (1-2\alpha) X_{k+1} B_w B_w^T X_{k+1} + R B_u B_u^T R = 0 \,, \end{aligned} \tag{6.3}$$

where α is a known scalar. Given an $\alpha > \frac{1}{2}$, equation (6.3) will always have a stabilizing solution at every step. Choosing a large value of α forces X_{k+1} to stay near X_k. Choosing α closer to $\frac{1}{2}$ allows X_{k+1} to change as much as possible while still ensuring that a stabilizing solution will exist.

The CGR iteration is performed as follows. Preselect two scalar relaxation parameters, α_1 and α_2. Then, at iteration k,

1. Set $\alpha = \alpha_1$. Solve eqn. (6.3) for a stabilizing X_{k+1}, given X_k.
2. Solve eqn. (5.28) for $Y \to Y_k$, given $X = X_{k+1}$.
3. Increment counter: $k \to k+1$.
4. Set $\alpha = \alpha_2$, $X_k = X_{k-1}$. Resolve eqn. (6.3) for X_{k+1}.
5. Resolve eqn. (5.28) for $Y \to Y_k$, given $X = X_{k+1}$.
6. Check stopping conditions. If not done: $k \to k+1$, then go back to step 1.

Choosing a relatively high choice for α_1 and a relatively low choice for α_2 in the CGR iteration often has following desirable effects: The solution for X from step 1 of the iteration stays near the previous X solution. Then, in step 2, Y is selected in response to this value of X. Step 4, then, allows the solution to move toward the (presumably pathological) solution that the original Riccati equation (5.27), would have yielded. In step 5, a solution for Y is found that balances out the effects of the nearly pathological solution. Finally, returning to step 1, the solution for X does not head in a divergent direction because of the choice for Y.

Again, because these iterative routines involve matrix equations, there appears to be no way to *a priori* determine whether or not an iteration will converge. However, we can demonstrate the effectiveness of the CG and CGR iterations on the 8-state example problem described in Appendix A.

Controllers were synthesized for the example problem using four methods: the standard iteration; the CG iteration; the CGR iteration; and, serving as a benchmark, an LMI-based minimization using LMILab. All controller synthesis was performed with a single, fixed value of the stability multiplier ($\sigma = 1$). The computer used was a Sun Sparc-20 workstation. For the iterative procedures, the stopping condition was based on the Frobenius norm of the change in Y. At iteration k, we calculate

$$Y_k \,, \quad \text{and} \quad dY_k \triangleq Y_k - Y_{k-1} \,.$$

If

$$\frac{\text{tr } (dY_k \, dY_k^T)}{\text{tr } (Y_k \, Y_k^T)} \leq 1 \times 10^{-14} \,,$$

then the iteration is deemed converged. It should be noted that, for convenience, the stopping condition is only checked after every even numbered step in the CGR routine.

In practice, it was found that a value of (10^{-14}) for the stopping condition was more than enough to ensure that the resulting controllers were identical to the controllers from the LMI-based solution. Even if the stopping condition was varied up or down by a few orders of magnitude, the Bode plots of the separation-based and LMI-based controllers matched. In addition, the separation-based cost function, equation (5.32), was found to equal the cost function that was minimized in the LMI-based synthesis technique, equation (4.8). For this problem, then, Conjecture 5.1 was found to hold.

The stated design point of the problem was for an uncertainty level of $\gamma = 2.0$. At this level of γ, the LMI problem solved in approximately 48 seconds. In contrast, the standard iteration was divergent. Even when it was initialized on the solution point, the iteration diverged toward infinity. However, the CG iteration converged nicely from a starting point of $X = 0$. It needed only one-fourth of the time required by the LMI solver. The convergence of the CG iteration did not appear to be sensitive to the choice of initial condition for X. The iteration converged from every initial point tested (varying X_0 between 0 and I), although an extensive test was not done to find a point which would cause the iteration to fail.

Use of the CGR iteration requires us to specify two parameters, α_1 and α_2. Some choices for the parameters cause the iteration to converge, while others can make it diverge. After testing the CGR iteration on this example problem and the problems that will be presented later in the chapter, several rules of thumb for selecting α_1 and α_2 were developed. One should always choose α_1 to be greater than α_2. The reasons for this were discussed previously. Also, it is useful to plot the OE cost, tr $(B_d - BcD_{yd})^T X_k (B_d - BcD_{yd})$, versus iteration to judge convergence. It is desirable to smooth any oscillations in the plot to speed convergence. The cost can oscillate in both a short period mode (with period 1, *i.e.*, between steps 2 and 5 of the iteration) and a longer period mode. Raising α_1 tends to decrease the overall rate of change of the iterate. When plotting the solution after each iteration, this can make the plot appear to be smoother over long periods. Changing α_2 tends to affect the short period character.

For the 8-state example problem, a series of tests was done at an uncertainty level of $\gamma = 1.0$, to see what values of α should be used. A wide range of values were tried for α_1 and α_2. Three representative plots of the OE cost at different values of α are shown in Figures 6-1, 6-2, and 6-3. By comparing the $(\alpha_1, \alpha_2) = (3, 1)$ case and the $(2, 1)$ case it can be seen that, raising α_1 caused the cost trajectory to appear smoother, though more oscillatory than before. Meanwhile, lowering α_2, seemed to successfully allow the estimator to "catch up" with the disturbance. This can be seen

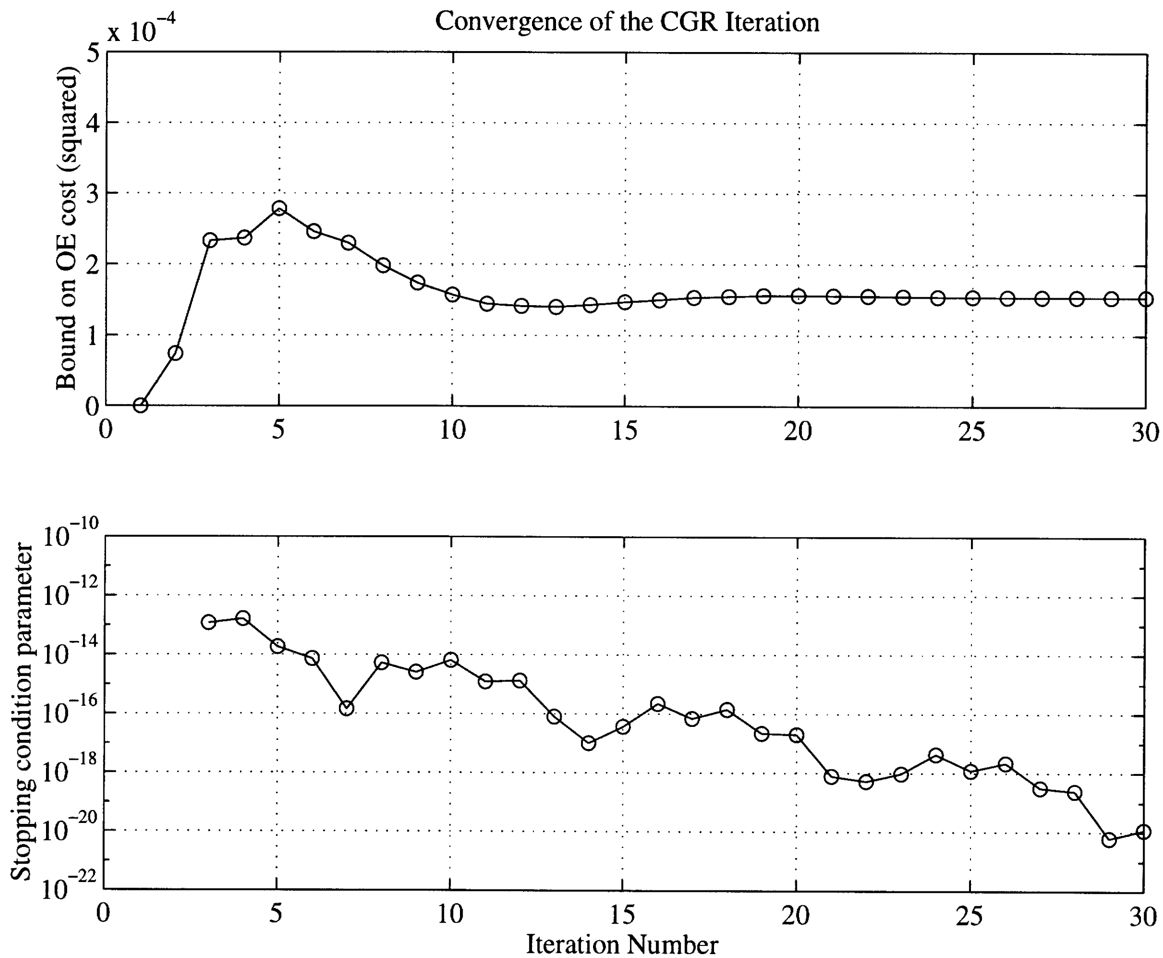

Figure 6-1: Convergence of the CGR iteration, with (α_1,α_2)=(2,1). The iteration is initialized at the solution from the LMI routine. Stopping condition tolerance set to 10^{-19}.

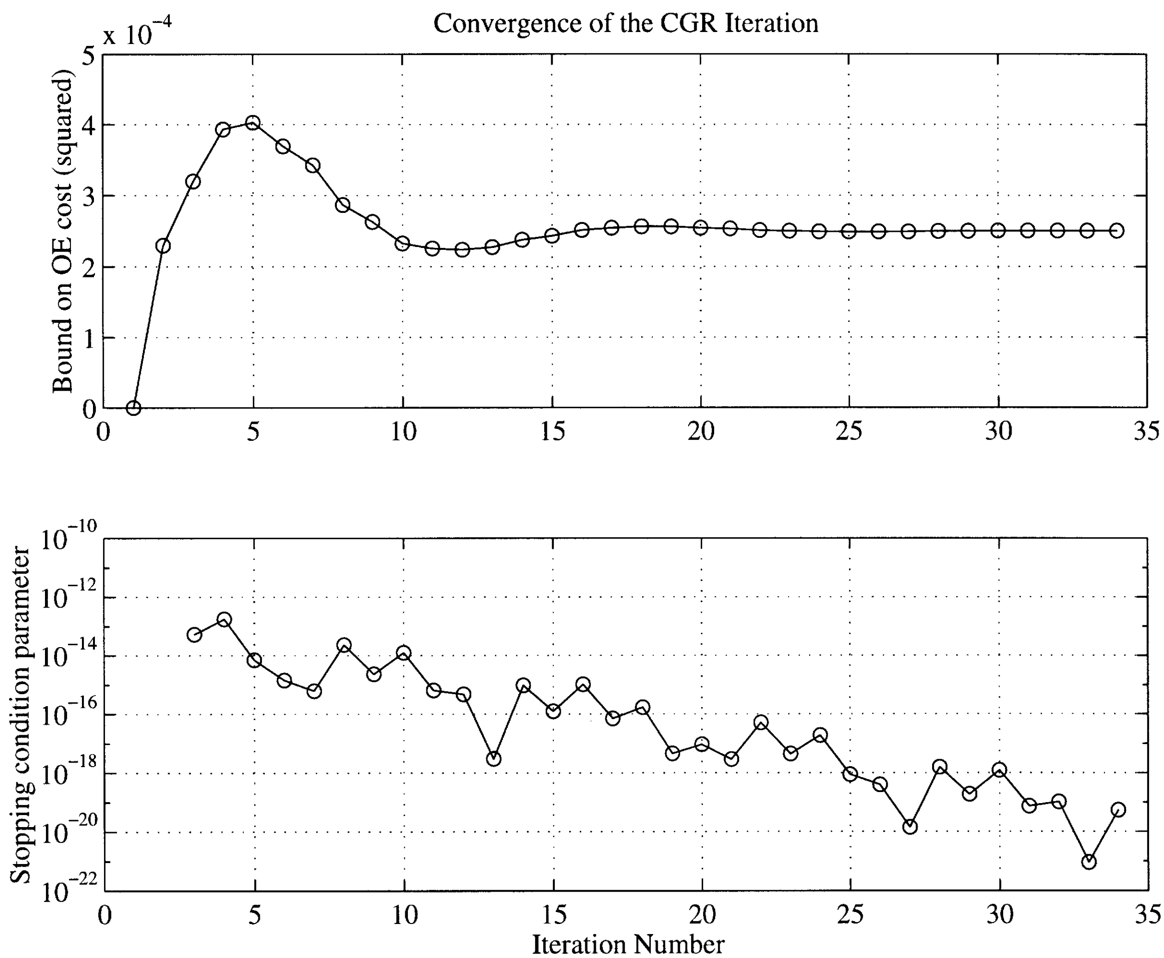

Figure 6-2: Convergence of the CGR iteration, with (α_1,α_2)=(3,1). The iteration is initialized at the solution from the LMI routine. Stopping condition tolerance set to 10^{-19}.

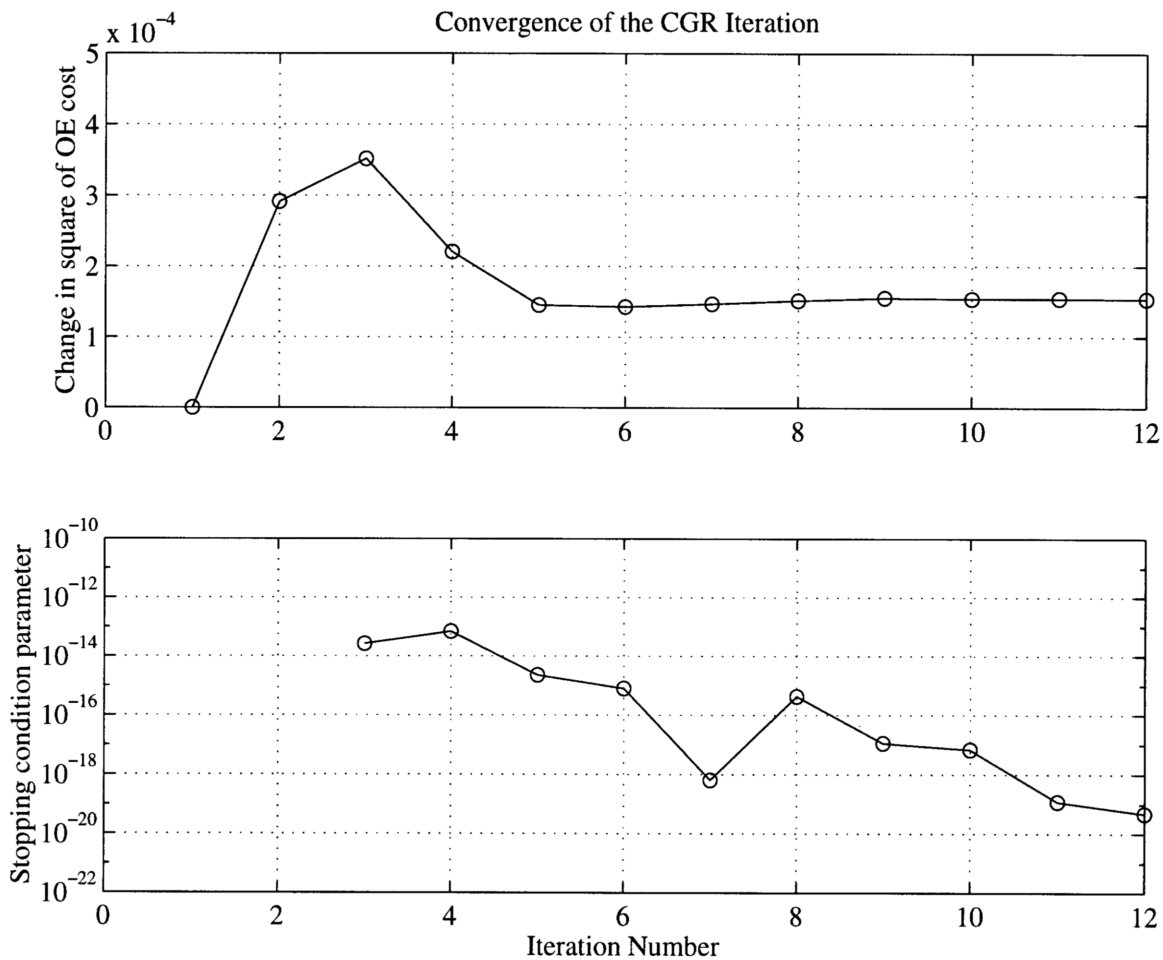

Figure 6-3: Convergence of the CGR iteration, with (α_1,α_2)=(2,0.55). The iteration is initialized at the solution from the LMI routine. Stopping condition tolerance set to 10^{-19}.

from the fact that the $(2, 0.55)$ case appears to be a version of the $(2, 1)$ case with approximately every other iteration point removed. Choosing $\alpha_1 = 2.0$ and $\alpha_2 = 0.55$ was found to make the iteration converge most rapidly, although not most smoothly. Furthermore, the choice of $\alpha_1 = 2.0$ and $\alpha_2 = 0.55$ was found to be approximately as effective at a wide range of values for γ. All data regarding the CGR iteration and the 8-state example was obtained using this choice for the parameters.

Finally, the four synthesis techniques were compared for an entire range of uncertainty levels. Of course, the range of uncertainties for which controllers exist is only semi-infinite. Recall that $\gamma = \infty$ would correspond to a case with no uncertainty, leading to an LQG controller. At the other extreme, as γ approaches the minimum level that can be obtained by an $\mathcal{H}_\infty$-optimal controller, $\gamma_{\min}$, the $\mathcal{H}_2/SN$ problem becomes intractable (by any means) because the $\mathcal{H}_2$ cost approaches infinity.

The time required for solution and the number of steps required for the iterative schemes to converge is listed in Table 6.1. In all cases, the iterative schemes were initialized at $X_0 = 0$. As the level of uncertainty increased (as γ decreased), the iterative methods required a greater number of steps to converge. Finally, at some point, they diverged. The lowest level of γ achieved by the standard iteration was only 2.55. In contrast, the CG iteration was efficient for uncertainties that were greater than twice as large, *i.e.*, down to better than $\gamma = 1.0$.

Note that at any particular level of uncertainty, there is a unique, minimal level of $\mathcal{H}_2$ cost that can be guaranteed by an $\mathcal{H}_2/SN$ controller. Furthermore, as already noted, for this system the cost function from the LMI-based and separation-based synthesis methods were found to match. Therefore, when speaking of the time to solution versus the level of uncertainty, we could equally well speak of the time to solution versus the bound on the $\mathcal{H}_2$ cost. The time to solution data of Table 6.1 is shown graphically in Figure 6-4. It is clear from the figure that at reasonable levels of uncertainty, the iteration methods are superior to the LMI solver. However, the LMI solver is extremely robust and can find solutions at any valid level of γ. Unfortunately, the new iteration schemes are not convergent at the lower extremes of γ. However, for this example, the CG iteration is still faster than the LMI solver out to better

γ	Bound on the $\mathcal{H}_2$ Cost (sqrd)	Method Used	Time required (sec).	Iterations
10	5.2	LMI	25.8	NA
		Stnd	2.9	4
		CG	3.4	5
		CGR	4.7	8
5	5.4	LMI	41.5	NA
		Stnd	4.4	7
		CG	4.5	7
		CGR	7.5	12
2.55	6.3	LMI	25.7	NA
		Stnd	18.5	34
		CG	7.5	13
		CGR	10.7	20
2.0	8.2	LMI	47.9	NA
		Stnd	Div.	Div.
		CG	11.7	19
		CGR	14.5	24
1.5	18.4	LMI	34.0	NA
		Stnd	Div.	Div.
		CG	12.9	21
		CGR	14.2	24
1.0	287.7	LMI	44.6	NA
		Stnd	Div.	Div.
		CG	33.7	51
		CGR	31.9	50
0.9	935.5	LMI	55.7	NA
		Stnd	Div.	Div.
		CG	2164	2498
		CGR	63.9	88
0.75	17,482.9	LMI	56.8	NA
		Stnd	Div.	Div.
		CG	Div.	Div.
		CGR	483.5	638

Table 6.1: Synthesis Routine Performance for a Range of Uncertainties. "Div." means that the method diverged. All iterative solutions were started with initial condition $X_0 = 0$. The squared $\mathcal{H}_2$ cost for a system with an LQG controller ($\gamma \to \infty$) is 5.1.

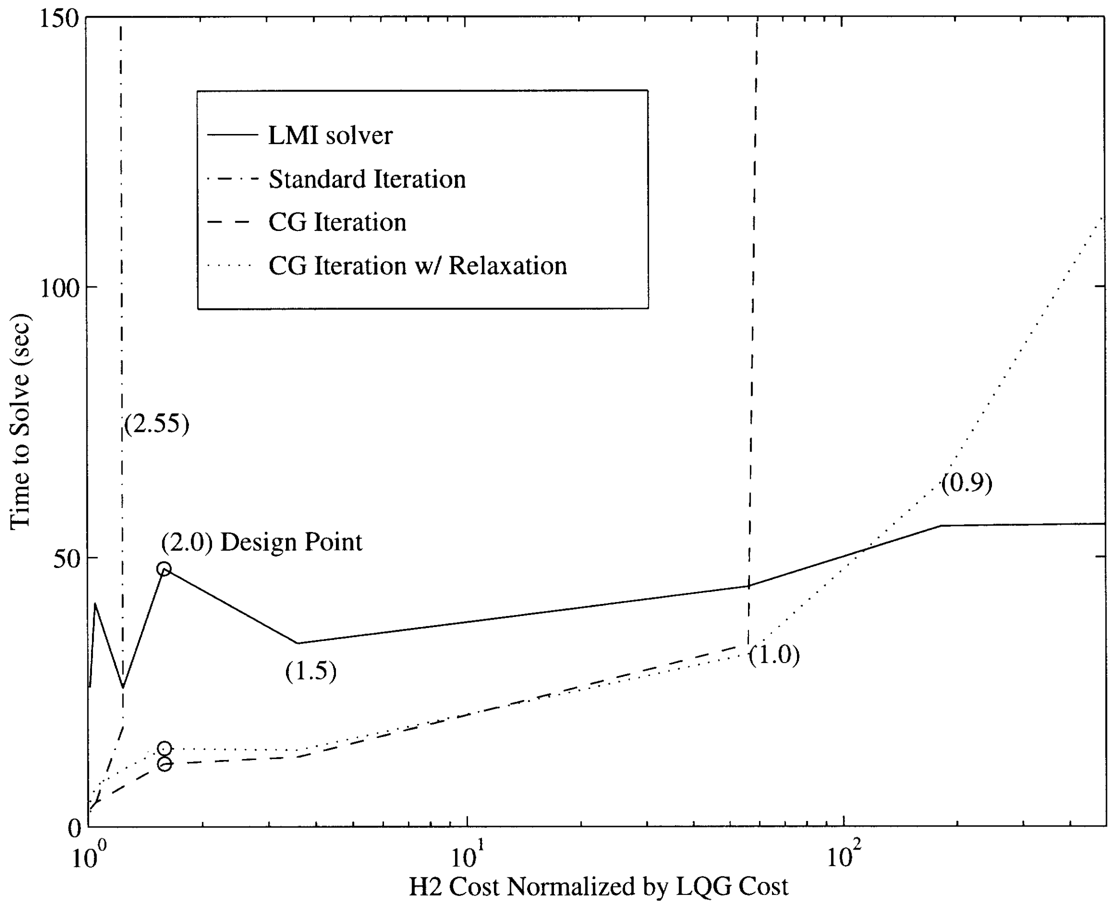

Figure 6-4: Time to solution vs. $\mathcal{H}_2$ cost (squared) for various synthesis techniques. Data is for a distinct set of 8 points only – lines plotted between points are not necessarily indicative of actual solution times. Levels of γ at five levels of $\mathcal{H}_2$ cost are indicated in parentheses. Notice that the standard iteration diverges for γ levels below 2.55. $\gamma_{\min}$ is approximately 0.66.

than 50 times the LQG cost and double the level of the design uncertainty. As γ is decreased past 1.0, the CG iteration takes considerably more time to converge. By a value of $\gamma = 0.9$, the CG iteration requires thousands of seconds to converge.

In practice, it should be the case that the iterative routines are even faster than indicated in the table and the figure. The runs used to generate the data for the table and figure were not refined to run at the highest possible speed. At each iteration, for instance, the iterative routines plotted figures which were similar to Figure 6-3. This and other computations were in place for diagnostic reasons. It was later found that, when these diagnostic tools were deactivated, the time required for the CGR iteration to solve the $\gamma = 1$ case dropped to 19.8 seconds.

It should also be mentioned that the behavior of the the CG and CGR iterations differs from that of the standard iteration in the cases where the iterations diverge. When the standard iteration diverges, it yields values for X that tend toward infinity. The iteration stops when either no stabilizing solution can be found for X or when the computer's numerical tolerances are exceeded. In contrast, the CG and CGR iterations tend to fall into limit cycles, which exist well within the machine's numerical limits, or into slowly expanding spirals. Because the modified iteration schemes can always find a stabilizing solution for X, these cycles can (theoretically) continue indefinitely. This, unfortunately, means that sophisticated criteria must be developed to detect if the iteration has fallen into an unstable spiral or limit cycle, so that the iteration can be stopped.

The CG and CGR iterations have been tested on several other small example problems, with similar results. One problem tested was the 8 state, single disturbance input $\mathcal{H}_2/\mathcal{H}_\infty$ synthesis problem of Ref. 5. In this case, the CGR routine was found to quickly converge to a solution for all desired levels of γ. In fact, the iteration yielded solutions at levels of γ which were, based on the reference, significantly below the level that was stated as being convergent when using a continuation method for γ wrapped around the standard iteration.

Controller synthesis problems were also run for systems with Popov multipliers. The problems were small enough such that LMI-based solutions could be obtained. It was found that controllers obtained by solving the conditions of Theorem 5.6 matched those obtained by performing the LMI-based minimization of Theorem 4.3. The controllers were identical for every example problem that was tested, despite the fact that the separation-based cost function was only guaranteed to overbound the LMI-based cost function. Unfortunately, it was also found that the separation-based cost function did not, in general match the LMI-based cost function. If the Popov multiplier, N, was set to zero, then the problem was essentially the same as the $\mathcal{H}_2$/*SN* case, and the cost functions were found to match. However, if N was nonzero and the quantity $(C_z B_u)$ was nonzero, then, even though the controllers matched, the cost functions did not. Why the quantity $(C_z B_u)$ should have an impact on whether or not the cost functions matched is partially explained by the fact that this term makes the OE system's realization different in the $\mathcal{H}_2$/*SN* and $\mathcal{H}_2$/Popov cases. Without $(C_z B_u)$, C_v has the same form as in the $\mathcal{H}_2$/*SN* case, and D_{vr} equals zero, as in the $\mathcal{H}_2$/*SN* case. Tests were also performed that seemed to demonstrate that the quantity $(C_z B_w)$ did not affect whether the cost functions matched.

The equivalence of the controllers, but lack of equivalence in the cost functions, for the $\mathcal{H}_2$/Popov case indicates that the separation-based cost function can probably be written in a less conservative form.

6.1.3 Implications for Controller Design

The results of the previous section indicated that, for the example problem, controller synthesis was possible using the CG and CGR iterations. The modified iterations were simple to initialize, and they could *not* enter a region where they could fail to yield a solution for X. Furthermore, compared to the standard iteration, the modified iterations were significantly more robust to the level of uncertainty. They were stable at higher levels of uncertainty. Furthermore, significant reductions to the time needed for solution were seen in comparison to the LMI-based synthesis routine.

Of course, one cannot draw general conclusions about how well the CG and CGR iterations will perform for a general plant based on the results of the sample 8-state problems. Nevertheless, if the results shown in Figure 6-4 are an indication, then it should perform well. For example, consider the Middeck Active Control Experiment (MACE), which flew on the space shuttle in 1995 (see Appendix B and Ref. 60). In that experiment, uncertainties in the structure's modal frequencies necessitated the use of robust controllers to achieve high-precision vibration control. Robust controllers were designed where the closed-loop cost of the perturbed system was less than 1.5 times the closed-loop LQG cost.[40,50] This makes the ability to design controllers in the 60-times-LQG-cost range ($\gamma = 1.0$), as demonstrated in the example problem, seem somewhat superfluous. Thus, it seems that for reasonable structural control experiments, where the models have been carefully developed, the CG iteration scheme should be in its usable range.

If controllers are desired to operate at the extreme range of γ, then, for small systems, the LMI synthesis technique can be employed. However, in these cases, it seems likely that the designer should consider using a pure $\mathcal{H}_\infty$ synthesis technique (*i.e.*, μ-synthesis), rather than an $\mathcal{H}_2/SN$ controller, since he is willing to sacrifice such high levels of performance.

Leaving aside the questions of convergence and uncertainty level, the CG and CGR iterations should be able to synthesize controllers more quickly than an LMI-based routine. The solution time of an LMI-based routine varies exponentially with the problem size. Assuming that a computer has enough memory to handle a given problem, the exponent is expected to be between 4 and 5. For the CG and CGR iterations, we expect solution times to vary proportionally to system size. To be efficient, we would would need the number of iterations in the CG and CGR routines to grow (at worst) linearly with system size. If this were the case, for a system with n states the solution time would scale according to the following formula: (close to, say, n iterations) $\times$ (2 Riccati equations per iteration) $\times$ ($(2n)^3$ solution time per equation) $=$ solution time proportional to $16n^4$. If this formula were correct, solution time would increase at an acceptable rate and the benefits of the separation-based routines over an LMI-based routine would become more and more significant as system size increased.

The results of the previous section also indicate that a controller derived by solving the FSFB (or FI) and OE problems can be identical to a controller derived by solving the LMI-based synthesis problem. Although there is no proof of this, it is expected that this will be the case for general systems. If this is found to not be the case, then a controller derived via the separation principle will still meet the desired robustness criteria, but will not deliver the lowest possible bound on the $\mathcal{H}_2$ cost.

The CG or CGR iteration should be used in the synthesis portion of a $D-K$ iteration controller design. This should produce designs more quickly than a $D-K$ iteration that employs LMIs for controller synthesis.

6.2 Implementation of an Analysis Routine

We have analyzed the $\mathcal{H}_2$ performance of an LTI system with static, real, parametric uncertainties using dissipation theory and Popov stability multipliers. For the closed loop system expressed in equations (4.2)–(4.4), the $\mathcal{H}_2$ performance is bounded by equation (4.29), where P is a size $2n \times 2n$ Lyapunov matrix satisfying the matrix inequality given in (4.30). Similarly, for a system with time-varying, gain bounded uncertainties, a simplified dissipation analysis can be performed using a constant stability multiplier. In this case, the $\mathcal{H}_2$ performance is bounded by equation (4.5), where P is a size $2n \times 2n$ Lyapunov matrix satisfying the matrix inequality given in (4.6).

With either uncertainty set, if the controller is held fixed, than the matrix inequality (or the Schur complement of it) becomes a linear matrix inequality. Also, the bound on the cost becomes linear, making the minimization of this bound a convex problem (recall the EVP discussed in Chapter 2). This means that, theoretically, the analysis problem can be solved using off-the-shelf LMI codes.

Unfortunately, just as the LMI-based synthesis approach of Chapter 4 was found to be unusable for higher-order systems, an analysis routine based on LMIs and currently available codes has been found to be unusable for such systems. On a Sun Sparc-20 workstation using LMILab,[32,30,31] we have only been able to analyze systems with up to approximately 16 states. The problem is that the LMIs routines required too much memory. Note that the dimension of the analysis LMI is even larger than the synthesis LMIs of Chapter 4. An analysis LMI constrains the closed loop system, so its outer dimension is size $2n + n_w$.

Therefore, we seek to develop an analysis routine that does not rely upon LMIs. Fortunately, we note that the problem is convex, so the new routine will not encounter local minima. In fact, because it is convex, there are quite a few different algorithms that could be chosen to solve the problem. Even algorithms that are not specifically designed for convex problems, such as a gradient search routine, should converge to the unique minimum. Regardless of the solution technique, however, we need to make the routine as efficient as possible, both in terms of speed as well as in terms of memory requirements, so that it can be used for problems with greater than, say, twenty states. We will concentrate our discussion on an analysis using Popov stability multipliers. The case of the single, scalar stability multiplier is a subset of the Popov multiplier case, so our results will carry over to this simpler case.

For the new analysis routine, we will continue to use equation (4.29) to define the cost function. Also, the solution needs to satisfy a constraint that is equivalent to the Popov LMI constraint (4.30) to achieve stability and performance robustness. The solution is to use Theorem 2.6. Substituting the closed loop system matrices to the Riccati equation of Theorem 2.6 and recalling the definition of ζ in equation (5.90), allows us to write an equivalent set of robustness constraints. They are

$$\tilde{P}(\hat{\tilde{A}} + \tilde{B}_w\zeta^{-1}\bar{C}_z) + (\hat{\tilde{A}}^T + \bar{C}_z^T\zeta^{-1}\tilde{B}_w^T)\tilde{P} + \tilde{C}_e^T\tilde{C}_e + \bar{C}_z^T\zeta^{-1}\bar{C}_z + \tilde{P}\tilde{B}_w\zeta^{-1}\tilde{B}_w^T\tilde{P} = 0 \quad (6.4)$$

and

$$\zeta > 0\,, \quad (6.5)$$

in which we have used the following two definitions,

$$\hat{\tilde{A}} \triangleq \tilde{A} + \tilde{B}_w M_1 \tilde{C}_z \quad (6.6)$$

$$\bar{C}_z \triangleq H\tilde{C}_z + N\tilde{C}_z\hat{\tilde{A}}\,. \quad (6.7)$$

Thus, the closed loop Riccati equation can be used to find the minimum value of the bound on the $\mathcal{H}_2$ cost. Also, note that, by definition, H and N are diagonal. We require that $H > 0$ and $N \geq 0$. As noted in Chapter 3, these sign-definiteness constraints are included so that the results can be compared to the designs in Refs. 46, 49, 50, 57.

To solve this optimization, we could minimize an augmented Lagrangian, consisting of our bound on the cost, equation (4.29), plus the sum of the trace of the two constraints, (6.4) and (6.5), each of which must be multiplied by a Lagrange multiplier matrix. This is similar to the optimization performed to obtain the Optimal Projection necessary conditions.[51] However, in the Optimal Projection case the object was to find the controller parameters for a fixed scaling. In our case, we are searching for the Lyapunov matrix, $\tilde{P}$, and the optimal scalings, H and N. Our optimization is also similar to the optimization examined by How *et al.*[46,48] In Chapter 5 of Ref. 46, How proposed to optimize equation (4.29), plus the trace of the Riccati equation (6.4) multiplied by a Lagrange multiplier matrix. In How's case, the controller parameters, Lyapunov matrices, scalings, and Lagrange multiplier were all unknown. The difference here is that we have made explicit the constraint on ζ and that, again, our controller parameters are fixed.

We desire to simplify the optimization so that it can be run on practical problems. How's approach leads us to conclude that we may be able to drop the second constraint, inequality (6.5), and still have a reasonable optimization. In the regions where ζ is not positive definite, we expect that no $\tilde{P}$ will exist to satisfy the Riccati inequality or that the cost will be high, so this constraint could be left out. Having only one explicit constraint simplifies the optimization considerably. Therefore, we will neglect this constraint. More will be said about this later.

Unfortunately, the proposed optimization is still too unwieldy for a large system. The unknown parameters in the above optimization are the $(2n)^2$ elements of $\tilde{P}$, the $(2n)^2$ elements of the Lagrange multiplier matrix, and the $2n_w$ nonzero elements in H and N. For a high order system, this is too many parameters to be able to expect to optimize quickly. Clearly, the greatest problem is the size of $\tilde{P}$. We will now demonstrate that we can perform the analysis optimization without explicitly optimizing of the $\tilde{P}$ variable and the Lagrange multiplier matrix, thus greatly reducing the problem size. This is the key feature of this optimization routine.

The optimization will be performed explicitly over only the H and N variables. The $\tilde{P}$ matrix will implicitly be allowed to vary with H and N. We achieve this by only examining the gradient directions that keep the optimization trajectory in the space specified by the Riccati constraint, equation (6.4). For clarity, we re-display the bound on the cost given in equation (4.29). It is

$$\mathcal{J} \triangleq \operatorname{tr}\left[\tilde{B}_d^T\left(\tilde{P} + \tilde{C}_z^T(M_2 - M_1)N\tilde{C}_z\right)\tilde{B}_d\right] .$$

We wish to find the gradient of $\mathcal{J}$ with respect to the diagonal parameters of H and N. The gradient of $\mathcal{J}$, with respect to H alone, is a size n_w vector,

$$\frac{\partial \mathcal{J}}{\partial H} \triangleq \left[\cdots \ \frac{\partial \mathcal{J}}{\partial h_i} \ \cdots\right]^T , \quad i \in [1, n_w] , \tag{6.8}$$

where h_i is the ith diagonal element in the H matrix. From the definition of $\mathcal{J}$, the derivative with respect to a single parameter is

$$\frac{\partial \mathcal{J}}{\partial h_i} = \operatorname{tr}\left(\tilde{B}_d\tilde{B}_d^T\frac{\partial \tilde{P}}{\partial h_i}\right) , \tag{6.9}$$

where $\frac{\partial \tilde{P}}{\partial h_i}$ is implicitly evaluated along a direction dictated by the Riccati constraint. To find this constrained gradient of $\tilde{P}$, we take the first variation of the Riccati constraint (6.4). Keeping only the first order terms, we obtain

$$\begin{aligned}\delta\tilde{P}(\hat{\tilde{A}} + \tilde{B}_w\zeta^{-1}\bar{C}_z) + (\hat{\tilde{A}} + \tilde{B}_w\zeta^{-1}\bar{C}_z)^T\delta\tilde{P} + \bar{C}_z^T\delta(\zeta^{-1})\bar{C}_z + \tilde{P}\tilde{B}_w\zeta^{-1}\delta\bar{C}_z \\ +\delta\bar{C}_z^T\zeta^{-1}\tilde{B}_w^T\tilde{P} + \tilde{P}\tilde{B}_w\zeta^{-1}\tilde{B}_w^T\delta\tilde{P} + \delta\tilde{P}\tilde{B}_w\zeta^{-1}\tilde{B}_w^T\tilde{P} + \tilde{P}\tilde{B}_w\delta(\zeta^{-1})\tilde{B}_w^T\tilde{P} = 0\end{aligned} \tag{6.10}$$

To evaluate this expression we write out the variation of each of the terms. Two of these are

$$\delta\bar{C}_z = \delta H\tilde{C}_z + \delta N\tilde{C}_z\hat{\tilde{A}} \tag{6.11}$$

$$\delta(\zeta^{-1}) = -\zeta^{-1}\left(\delta H M_d + M_d\delta H - \tilde{B}_w^T\tilde{C}_z^T\delta N - \delta N\tilde{C}_z\tilde{B}_w\right)\zeta^{-1}\,. \tag{6.12}$$

Substituting the above expressions into equation (6.10) yields a complete expression for the first variation of the Riccati constraint. To determine the derivative of $\tilde{P}$ with respect to h_i, we need only examine the variation of the Riccati constraint case with H varying, *i.e.*, where $\delta N = 0$. This is

$$\begin{aligned}\delta\tilde{P}(\hat{\tilde{A}} + \tilde{B}_w\zeta^{-1}C_P) + (\hat{\tilde{A}} + \tilde{B}_w\zeta^{-1}C_P)^T\delta\tilde{P} + C_P^T\zeta^{-1}\delta H\tilde{C}_z \\ +\tilde{C}_z^T\delta H\zeta^{-1}C_P - C_P^T\zeta^{-1}(\delta H M_d + M_d\delta H)\zeta^{-1}C_P = 0\,,\end{aligned} \tag{6.13}$$

where, for convenience, we have defined that

$$C_P \triangleq \bar{C}_z + \tilde{B}_w^T\tilde{P} = H\tilde{C}_z + N\tilde{C}_z\hat{\tilde{A}} + \tilde{B}_w^T\tilde{P}\,. \tag{6.14}$$

It can be seen that equation (6.13) is a Lyapunov matrix for $\delta\tilde{P}$ in terms of δH.

To use equation (6.13) to calculate the effects of changing element i of the H matrix, we can set

$$\delta H \longleftarrow \Delta H_i \triangleq \begin{bmatrix} 0 & & \cdots & & 0 \\ & \ddots & & & \\ \vdots & & \Delta h_i & & \vdots \\ & & & \ddots & \\ 0 & & \cdots & & 0 \end{bmatrix}, \tag{6.15}$$

i.e., δH is set to a matrix of zeros except for the ith element on the diagonal. To calculate the derivative of $\tilde{P}$ with respect to h_i, we choose $\Delta h_i = 1$ and define δH based on this. Then, equation (6.13) can be solved, yielding $\delta\tilde{P} \to \Delta\tilde{P}_i$. The derivative is then

$$\frac{\partial \tilde{P}}{\partial h_i} = \Delta\tilde{P}/\Delta h_i = \Delta\tilde{P}_i \,. \tag{6.16}$$

The only difficulty with this procedure is that we must solve a different Lyapunov equation for each of the n_w uncertain parameters. Fortunately, we can simplify this procedure by making use of a dual form of the Lyapunov equation.

To find the constrained gradient of the cost, $\mathcal{J}$, with respect to H, we must calculate the n_w scalar derivatives $\partial\mathcal{J}/\partial h_i$, $i \in [1 \ldots n_w]$. These were defined in equation (6.9). Given the Lyapunov equation of equation (6.13), our knowledge of linear systems tells us that these derivatives are numerically equal to the $\mathcal{H}_2$ norms of the following LTI systems

$$\left[\begin{array}{c|c} \hat{\tilde{A}} + \tilde{B}_w\zeta^{-1}C_P & \tilde{B}_d \\ \hline \begin{array}{l}\big(C_P^T\zeta^{-1}\Delta H_i\tilde{C}_z + \tilde{C}_z^T\Delta H_i\zeta^{-1}C_P \\ \quad -C_P^T\zeta^{-1}(\Delta H_i M_d + M_d\Delta H_i)\zeta^{-1}C_P\big)^{1/2}\end{array} & 0 \end{array}\right], \quad i \in [1 \ldots n_w]\,, \tag{6.17}$$

which are subject to exogenous white noise inputs with zero mean and unit covariance. However, the $\mathcal{H}_2$ norm of a system described by equation (6.17) can also be calculated by using a dual formulation. The norm, and, hence, $\partial\mathcal{J}/\partial h_i$, is also equal to

$$\begin{aligned} &\operatorname{tr} \Delta Q_H \left[C_P^T\zeta^{-1}\Delta H_i\tilde{C}_z + \tilde{C}_z^T\Delta H_i\zeta^{-1}C_P - C_P^T\zeta^{-1}(\Delta H_i M_d + M_d\Delta H_i)\zeta^{-1}C_P\right] \\ &= 2\operatorname{tr}\left[\Delta Q_H C_P^T\zeta^{-1}(\Delta H_i)(\tilde{C}_z - M_d\zeta^{-1}C_P)\right], \end{aligned} \tag{6.18}$$

where the ΔH_i are defined in equation (6.15), and ΔQ_H is the solution to the following Lyapunov equation

$$(\hat{\tilde{A}} + \tilde{B}_w\zeta^{-1}C_P)\Delta Q_H + \Delta Q_H(\hat{\tilde{A}} + \tilde{B}_w\zeta^{-1}C_P)^T + \tilde{B}_d\tilde{B}_d^T = 0\,. \tag{6.19}$$

Notice that equation (6.19) is not a function of ΔH_i. Therefore, ΔH_i is independent of which uncertainty signal, i, is being examined. To find the derivative, we need only solve equation (6.19) once. Given ΔQ_H, the n_w scalar derivatives are then found by evaluating the trace quantity in equation (6.18) n_w times, for each of the possible choices of ΔH_i. For each uncertainty $i \in [1 \ldots n_w]$, we define ΔH_i using $\Delta h_i = 1$.

The above derivation carries over in a straightforward fashion to the N scales. We calculate the constrained gradient of the cost component-wise. The n_w scalar derivatives of the cost are each defined with respect to a particular diagonal element of N, *i.e.*,

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial n_i} &= 2\,\mathrm{tr}\left[\Delta Q_N C_P^T \zeta^{-1} \Delta N_i (\tilde{C}_z \hat{\tilde{A}} + \tilde{C}_z \tilde{B}_w \zeta^{-1} C_P)\right] \\ &\quad + \mathrm{tr}\left(\tilde{B}_d^T \tilde{C}_z^T (M_2 - M_1) \Delta N_i \tilde{C}_z \tilde{B}_d\right), \quad i \in [1 \ldots n_w], \end{aligned} \tag{6.20}$$

where n_i is the ith diagonal element of N, ΔN_i is defined to be a matrix of zeros except for the (i,i) element which is set to unity, and where ΔQ_N is the solution to the following Lyapunov equation

$$(\hat{\tilde{A}} + \tilde{B}_w \zeta^{-1} C_P) \Delta Q_N + \Delta Q_N (\hat{\tilde{A}} + \tilde{B}_w \zeta^{-1} C_P)^T + \tilde{B}_d \tilde{B}_d^T = 0\,. \tag{6.21}$$

A careful examination of equations (6.19) and (6.21) reveals that they have exactly the same form. Therefore, for any problem, we will find that $\Delta Q_N = \Delta Q_H$.

As an aside, it should be noted that equations (6.19) and (6.21) are identical in form to a Lyapunov equation that was derived (but not used) by How.[46] In Ref. 46, the equation was derived by taking the derivative of the augmented cost function (mentioned earlier) with respect to the Lyapunov matrix, $\tilde{P}$. In that case, the variable that took the place of ΔQ_N was the Lagrange multiplier for the Riccati constraint. This makes intuitive sense because Lagrange multipliers give the sensitivity of a cost function to changes in its constraints. However, it should also be noted that the gradients obtained from equations (6.19) and (6.21) are much simpler than those found by taking the gradients of the augmented cost function. In Ref. 46, when using the augmented cost function, analytic forms for the gradients were necessarily taken with respect to changes in every term of the H and N matrices, even though these matrices were actually diagonal. In our case, we are able to simplify the formulation and restrict our attention to the effects of the diagonal elements.

In summary, we have shown that we can calculate the constrained gradient of the Lyapunov matrix with respect to the scalings H and N by merely solving a *single*, size $2n$ Lyapunov equation and then by evaluating $2n_w$ trace quantities. Additionally, the cost function, $\mathcal{J}$, for any H and N can be evaluated by using the closed loop Riccati equation, equation (6.4). Together, these operations form the basis for an efficient analysis routine.

6.2.1 Coding of the Analysis Routine

The analysis routine that we propose is a constrained optimization. There is a cost function, equation (4.29), together with certain constraints. There is the Riccati constraint, equation (6.4), which guarantees that the system fulfills the robustness requirements. This is implicitly enforced by constraining the gradient of the cost. However, there are two additional types of constraints that must be considered.

The next constraint that we will discuss is the requirement that H is positive definite and N is positive semidefinite. The constraint on H ensures that each term in the uncertainty block affects the system. The constraint on N enables the analysis routine to obtain information about the phase of a static real parametric uncertainty (or a static sector bounded nonlinearity). The constraints on H and N are, in fact, simple to enforce because these are diagonal matrices. Therefore, the constraints can be enforced on the individual diagonal elements of the matrices, h_i and n_i.

In practice, it has been found that the constraint $H > 0$ should be implemented as $H \geq \epsilon$, where ϵ is a small, arbitrary positive number. The important point to note, however, is that ϵ should be chosen to be significantly larger than the minimum precision of the optimization algorithm. This should help prevent the routine from running into precision problems — essentially, this should prevent the routine from considering a semi-definite matrix for H. Of course, the value for ϵ that should be chosen is problem specific. For the examples that will be discussed in the next section, ϵ was typically set to 10^{-4} or 10^{-6}. This was far from the expected precision of the algorithm (which was better than 10^{-10}).

One last constraint must be discussed. The quantity ζ should be constrained to be positive definite (recall inequality (6.5)). This condition is significantly more difficult to implement in a gradient type routine than were the definiteness constraints on H and N. Note that gradient routines, unlike LMI routines, optimize a scalar cost function subject to scalar constraints. The problem with a constraint on ζ is that it is not a diagonal matrix. It is a linear matrix function of the H and N variables. Clearly, we would like to implement this constraint in an LMI-type framework. Unfortunately, this is not a simple proposition using scalar constraints. It is possible that one could enforce the constraint on ζ by requiring that the n_w determinants of ζ and its sub-matrices are positive. These are difficult, nonlinear constraints to implement, particularly since the matrices $\tilde{C}_z$ and $\tilde{B}_w$ can be general, rank-deficient forms. The gradients of the constraint require a great deal of algebraic manipulation to evaluate efficiently. Therefore, as mentioned previously, we will follow the approach of How[46] and we will not include this as an explicit constraint in the optimization. It is felt that the Riccati equation will typically not have solutions in areas where the constraint on ζ would be violated, thus making the constraint superfluous.

However, depending on the coding of the gradient method, it is certainly possible that the effects of this constraint could be implemented. One *ad hoc* way to achieve some of the benefits of the constraint is to include it as an implicit penalty weighting when evaluating the cost. If ζ is found to have negative eigenvalues at a given point, then one can arbitrarily add a large value to the cost at this point.

To summarize the discussion so far, our optimization problem is to minimize $\mathcal{J}$, given by equation (4.29), with respect to the n_w parameters h_i and n_i, subject to the constraints that $h_i > 0$ and $n_i \geq 0$, for $i \in [1, n_w]$. The only variables in the optimization are the diagonal entries of H and N. The gradient of $\mathcal{J}$ is to be evaluated using the constrained gradient discussed in the previous section.

We will use a sequential quadratic programming (SQP) routine to solve this constrained optimization. The reader is referred to Refs. 37, 73 and the references contained therein for more detailed discussions of such a routine. Certainly, we could choose to spend the time to implement an optimization code that is geared for convex problems. An example of such a routine would be the method of centers.[8] However, implementation of such a routine would require a significant amount of development time. Another potentially useful optimization technique could be the successive approximation method presented in Refs. 73,21. This method is presented as a replacement for SQP routines and requires less memory to run. However, this type of routine would also require a considerable amount of development time. The use of an SQP algorithm offers several practical benefits. SQP codes are readily available, and they are easy to implement and debug. Furthermore, use of an SQP routine will demonstrate that the analysis procedure does not need to be solved by a state-of-the-art convex optimization routine.

An SQP routine implicitly forms an augmented Lagrangian consisting of the cost function and the constraints, together with Lagrange multipliers. This is why we have not discussed augmenting the cost function with the positive definiteness constraints. We need merely supply such a routine with a means to evaluate the cost function and constraints at any given point. In addition, we also supply the routine with the gradient of the cost function (the constrained gradient) and the gradient of the constraints. However, we do not need to provide gradients of the augmented Lagrangian. An SQP routine attempts to minimize this augmented Lagrangian by implicitly solving the Kuhn-Tucker optimality conditions and finding the Lagrange multipliers for the constraints.

We can briefly outline the major steps of an SQP routine. Given the cost, the gradient of the cost, and an estimate of the Hessian matrix for the current iteration, the routine sets up a quadratic programming (QP) problem. The purpose of the QP is to find the best direction in which to search for a point with a reduced cost. The QP minimizes a cost function that depends on the square of the step direction, as well as the Hessian matrix and the cost gradient. The QP can be solved using a variety of standard techniques, including linear programming.[37] After obtaining a direction from the QP, a line search routine is implemented. The line search routine searches in the indicated direction for a point which yields a lower value for the cost. Some sophisticated routines attempt to find the point which yields the *lowest* cost in the indicated direction (in the neighborhood of the current solution). The line search routine implemented in the Matlab Optimization Toolbox[37] merely searches for *any* lower cost point. The last major step of an SQP routine is to find and update the Hessian matrix for the current iteration. Since the Hessian is expected to be a large matrix, it is only estimated. A standard procedure used to update the Hessian is the BFGS (Broyden, Fletcher, Goldfarb, and Shanno) routine, which requires gradient information at the current and previous iterations.

Regardless of the specific routine that is chosen to solve the optimization, the main benefit of this analysis method is that it does not require large amounts of computer memory. Using the SQP routine in the Matlab Optimization Toolbox (known as *constrain.m*) with a slightly modified line search routine, the analysis routine has proven effective for systems with 8, 24, and 59 states. The Matlab line search routine was modified in certain examples to search over only half the originally programmed number of points. This reduced the search time. The results indicated that the proposed analysis routine should be useful for higher-order systems.

6.3 Testing of a $D - K$ Iteration

The synthesis and analysis algorithms developed in the previous two sections are combined to form a complete robust $\mathcal{H}_2$ controller design algorithm. The algorithm is a $D - K$ iteration. In this section, it is used to successfully design $\mathcal{H}_2$/Popov controllers for three structural control problems with real, parametric uncertainties.

As has been the case throughout this thesis, our primary interest is in the utility of our design algorithms. We wish to have a fast and easy-to-use design algorithm. While we will occasionally comment on the robustness characteristics of our controller designs, this is not our focus. In fact, each of the examples that we will examine has already been examined by either How *et al.*[46,50] or Livadas.[57] How's emphasis was on an examination of the characteristics of the $\mathcal{H}_2$/Popov controllers. Assuming that the stated results in the references are correct, then our designs should duplicate them.

The $D - K$ iteration will now be described in detail. To start the iteration, the designer must first pick the initial stability multipliers, H and N in the $\mathcal{H}_2$/Popov case, and σ in the $\mathcal{H}_2/SN$ case. The initial stability multiplier is then held fixed and the synthesis problem is solved. The controller that is derived in the synthesis step is then held fixed so that the analysis problem can be solved. The new set of stability multipliers that comes from the analysis step is then used to start the next iteration. A cost function is evaluated after each iteration (at the end of the synthesis step). The iteration is repeated until the function converges. The cost function is given by equation (4.29), and is calculated by solving the closed loop Riccati equation, given in (6.4). To be specific, after iteration $k+1$, if

$$\left|\frac{\mathcal{J}_{k+1} - \mathcal{J}_k}{\mathcal{J}_k}\right| \leq \ cost\_tol \tag{6.22}$$

where *cost_tol* is a prespecified parameter, then the $D-K$ iteration is deemed converged. It should also be mentioned that the $D-K$ iteration is also required to run a *minimum* number of iterations, denoted *min_iter*. This helps ensure that the routine will not indicate that it is converged when it has merely been initialized in a region where the cost function varies slowly with the multipliers.

It might seem that one could try to initialize the $D-K$ iteration by guessing an initial controller and using this to start the analysis algorithm. However, in practice, we have found this option to be considerably more difficult. The problem is that, in effect, one must solve a robust control design to obtain an initial controller guess. A guess such as the LQG controller may not work, because it may not allow the closed loop system to fulfill the stability robustness requirements. Therefore, the analysis algorithm would have no solution. Similarly, one could try to use the $\mathcal{H}_\infty$ central controller by disregarding the $\mathcal{H}_2$ performance measure and merely ensuring that the closed loop system is robustly stable. However, when attempting to synthesize this $\mathcal{H}_\infty$ controller, it should be understood that stability multipliers have implicitly already been picked in this synthesis. They are $H = I$ and $N = 0$ (or, equivalently, $\sigma = 1$). There is no guarantee that such a controller will exist, *i.e.*, the $\mathcal{H}_\infty$ central controller Riccati equations may not have a solution. Therefore, one should guess at stability multipliers until the initial synthesis step yields a valid controller.

It should also be noted that there may be cases in which the designer is unable to find a set of stability multipliers that can start the iteration. This can mean one of two things. First, it is possible that no LTI controller exists that can achieve the desired stability robustness bounds. A second, more pleasant, possibility is that controllers exist but the guesses for the stability multipliers have been inadequate. The problem in this case can usually be dealt with by wrapping a continuation method around the $D-K$ iteration. A controller design is first performed for a system with a lower level of uncertainty. This can then serve as an initial condition for a problem with a larger uncertainty. This process is repeated until the desired level of uncertainty is reached. A continuation method will be demonstrated on the third example in this section.

In addition to the parameters that specify when the overall $D-K$ iteration is deemed converged, both the analysis and synthesis steps each have their own convergence criteria. However, the convergence criteria of the inner steps are not as important to the design process as is the outer convergence parameter, *cost_tol*. While the individual steps of a $D-K$ iteration are themselves convex problems, the overall iteration is not. It may have local minima and, if so, which minimum is found can certainly be path dependent. The difficulty is that we have no way of knowing whether or not the iteration is following a path that will lead to the best solution. There is no evidence that optimizing each synthesis and analysis step to the finest possible precision will necessarily lead to the best path. Certainly, it leads to an acceptable path, in that the cost function is always guaranteed to decrease. However, it can also add unnecessary time to the design procedure. Instead, we can also follow a path found by allowing each analysis and synthesis step to be suboptimal. One need not require that high degrees of precision are obtained at each step, but only that the cost is moved downwards enough to make a significant difference in the overall iteration. This is the course that we adopt. The convergence criteria for the analysis and synthesis algorithms are discussed in light of this fact.

As discussed in Section 6.1.2, the synthesis step uses either a CG or a CGR iteration. For clarity, we note that in the CG iteration, the convergence criteria are checked after every solution for X. Meanwhile, in the CGR iteration, the convergence criteria are checked after only every other solution for X. Also, the convergence criteria for the CG-type methods are different than the criterion that was used in Section 6.1.2. Two criteria are used. The first criterion is based on how closely equation (5.94) or (5.27) is satisfied. Note that it is the solution for X given a value of Y that is only approximated in the CG-type methods. Meanwhile, the solution for Y always exactly satisfies the desired estimator equation ((5.95) or (5.28)) at each step. For simplicity, consider just the $\mathcal{H}_2$/Popov case. The $\mathcal{H}_2$/*SN* case follows by analogy. After iteration number $k+1$, we calculate the residual error by substituting the latest iterate into equation (5.94), *i.e.*,

$$X_{\text{res}} \triangleq X_{k+1}\left[A - (Y_k C_y^T + B_d D_{yd}^T)E_d^{-1}C_y\right] + \left[A - (Y_k C_y^T + B_d D_{yd}^T)E_d^{-1}C_y\right]^T X_{k+1}$$
$$+(X_{k+1}B_r + C_v^T E_u D_{vr})\xi^{-1}(B_r^T X_{k+1} + D_{vr}^T E_u C_v) + C_v^T E_u C_v \,. \quad (6.23)$$

Note that X_{res} will only equal the zero matrix when X_{k+1} is equal to the desired solution. An easily computable measure of the convergence is the trace of X_{res}, which also equals zero at the solution. The first stopping criterion for the synthesis algorithm is that

$$x_{\text{tol}} < \text{tr } X_{\text{res}} < 0\,, \quad (6.24)$$

where x_{tol} is a preselected *negative* parameter. The second stopping criterion is that the bound on the cost, as defined in equation (5.103) should be converged. When the absolute value of the fractional change in the cost is less than a prespecified parameter, *Jsynth_tol*, then the cost is deemed converged.

When both convergence criteria of the synthesis procedure are satisfied, then the procedure is stopped. These stopping criteria give us a great deal of flexibility. They give us the ability to allow distinctly suboptimal controllers as the output of the synthesis routine. As discussed in Section 5.3, if the matrix X_{res}, defined in equation (6.23), is negative semidefinite, then we are assured that a suboptimal controller exists. This controller will satisfy the required stability robustness constraints, but does not minimize the bound on the $\mathcal{H}_2$ cost. Of course, the trace of X_{res} cannot be the only stopping criterion, since the trace could be negative even when the matrix has some positive eigenvalues. However, in practice, it is found that when the trace constraint is combined with the restriction that the cost be convergent, there is no problem. As noted in Section 6.1.2, if the CG and CGR iterations are not convergent, they tend to fall into limit cycles, and these do not satisfy the cost convergence criterion. Therefore, setting x_{tol} closer to zero ensures that the resulting controller will be closer to optimal.

The analysis routine consists of the SQP algorithm discussed in the previous section. The SQP algorithm is deemed converged when the variables and the cost cannot be made to change significantly. This is specified by two parameters, *anvar_tol* and *ancost_tol* which correspond to the parameters known as *OPTIONS(2)* and *OPTIONS(3)* in Ref. 37. When weighted functions of the change in the variables and the cost fall below these parameters, the routine is considered to be converged. The reader is referred to Ref. 37 for a more complete discussion of these convergence issues.

In addition to converging to a solution, the SQP routine is also terminated if it has iterated for a period of time and brought the cost function down, but has not converged. Allowing the analysis routine to terminate early is particularly useful in cases where a superior solution can be obtained but only at great expense in terms of time. This is common in many optimization algorithms. As already discussed, it is not the case that we need every analysis algorithm to obtain the best stability multipliers. We only need better multipliers than the ones that we started with. In our analysis, after the routine has evaluated the cost at more than a prespecified number of points (even if some of these point proved to be infeasible), then the analysis is terminated. This parameter is denoted as *an_max*.

Therefore, in each of the following examples, we will specify seven convergence parameters: *cost_tol*, *min_iter*, x_{tol}, *Jsynth_tol*, *anvar_tol*, *ancost_tol*, and *an_max*.

For clarity, we note that in our terminology, an iteration refers to an analysis step followed by a synthesis step. Because we start with an initial controller synthesis, there is always one more synthesis step than there are iterations or analysis steps.

Unless otherwise noted, a discussion of solution time for a design will not include the time required for the initial synthesis step.

Each of the following examples was run on a Sun Microsystems Sparc-20 workstation. All programs were executed in Matlab. The solution of all Riccati equations was accomplished using the Schur decomposition method in the μ Synthesis Toolbox.[2] All operations involving LMIs were done using the LMI Control Toolbox.[32,30,31]

6.3.1 Four Mass System

In the first example, we will design a controller for the system with four masses and three springs detailed in Appendix A. The stiffness of two of the springs is uncertain, but a possible range of values is known for them. Thus, the system has real, parametric uncertainties. As an aside, we note that this is the same nominal system for which we synthesize $\mathcal{H}_2/SN$ controllers in Section 6.1.2. However, in the previous case, there is only a single uncertainty, an uncertain mass. We choose to do the design for the system with the uncertain springs so that we can compare the controller to the $\mathcal{H}_2$/Popov design done by Livadas.[57] Livadas also performs a $D-K$ iteration. However, for the synthesis step, he uses the LMI methodology presented in Chapter 4. Similarly, for the analysis step, Livadas minimizes the bound on the $\mathcal{H}_2$ cost given in equation (4.29), subject to LMI (4.30).

This design plant has 8 states. The controller is eighth order and SISO. There are two real parametric uncertainties. With a Popov multiplier for each uncertainty, this translates to 4 scalar parameters, $h_i, n_i, i \in [1 \dots 4]$, that must be optimized in each analysis step.

For this example, all synthesis steps were performed using the CG iteration. The CG iteration converged quickly to the desired controller without problem. The analysis steps were performed with the previously discussed SQP algorithm. The algorithm worked effectively and converged to an optimal set of multipliers in each iteration. It was found that at the start of the line search portion of the algorithm, the routine typically had to try two or three points before finding a point for which the cost function was lower than the current point. However, in the subsequent line searches,

Parameter	Value
cost_tol	1×10^{-4}
min_iter	3
x_{tol}	-1×10^{-6}
Jsynth_tol	1×10^{-6}
anvar_tol	1×10^{-4}
ancost_tol	1×10^{-4}
an_max	400

Table 6.2: Convergence criteria for the 4 mass problem

	Total Solution Time (min:sec)	# of Synth. Steps	Avg. Synth. Time (sec)	# of Anlys. Steps	Avg. Anlys. Time (sec)	Time per Full Iteration (sec)
Separation Based	3:37	11	8	10	14	22
LMI Based	na	na	na	na	na	129

Table 6.3: Solution Time for the 4 mass problem. na = not available or not applicable for comparison, because the criteria to determine convergence after each synthesis step of the $D-K$ iteration are not the same. The LMI-based results are from Reference 57, Table 4.14. Total solution time does include the initial controller synthesis.

the routine usually found a new point immediately.

Overall, the design algorithm works very well for this system. The $D-K$ iteration is started with an initial guess of $H = I$ and $N = 0$. The stopping parameters for this case are listed in Table 6.2. The algorithm completes the design in 10 iterations, *i.e.*, 11 synthesis steps and 10 analysis steps. Figure 6-5 shows how the cost changed after each iteration. In fact, the plot shows that the change in the cost is monotonically decreasing, indicating that the $D-K$ iteration has found a path that allows it to smoothly converge. We need not expect this to be the case in general, however. In fact, were we to run the iteration further, we would expect the second plot to level off or to sawtooth around a certain level, indicating the finite precision of the routine.

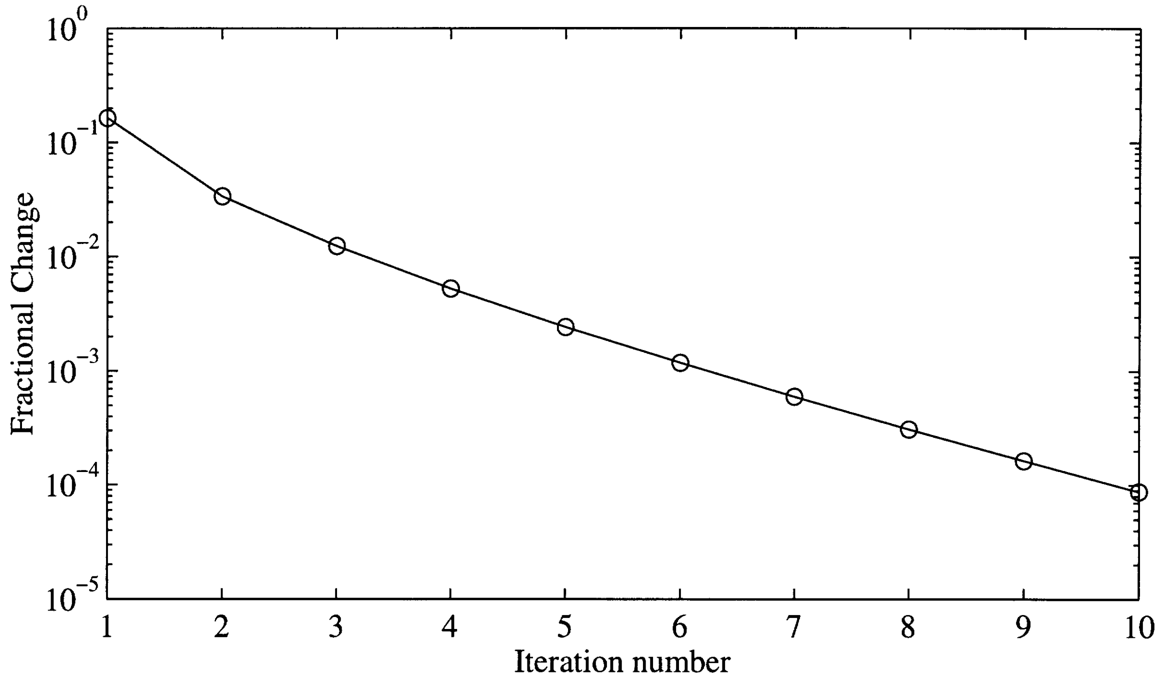

Figure 6-5: Convergence of the 4 mass problem. Fractional change in the cost function after each iteration.

The most important result for our purposes is the time that the algorithm required to solve the problem. These results are summarized in Table 6.3. The design required approximately three and a half minutes to solve. We can compare this to the results found using an LMI-based method. The best metric to examine for a comparison of the algorithms is the average time required for an iteration (an analysis step followed by a synthesis step). As shown in the table, the iteration that uses the separation principle is almost 6 times faster than the iteration that relies upon LMIs. This is in general agreement with the results for the synthesis step alone, discussed in Section 6.1.2.

Also shown clearly in Table 6.3 is the fact that the analysis problem takes longer to solve than the synthesis problem. Intuitively, one would probably expect that it takes less time to solve the analysis step than the synthesis step. The reasoning being that analysis step need only optimize over 4 parameters—the diagonal matrices H and N. In contrast, the synthesis step must optimize over the eight-state controller. In practice, however, the analysis algorithm is slower because it must solve larger Riccati equations. Analysis is done on the closed loop system, so the Riccati equation for

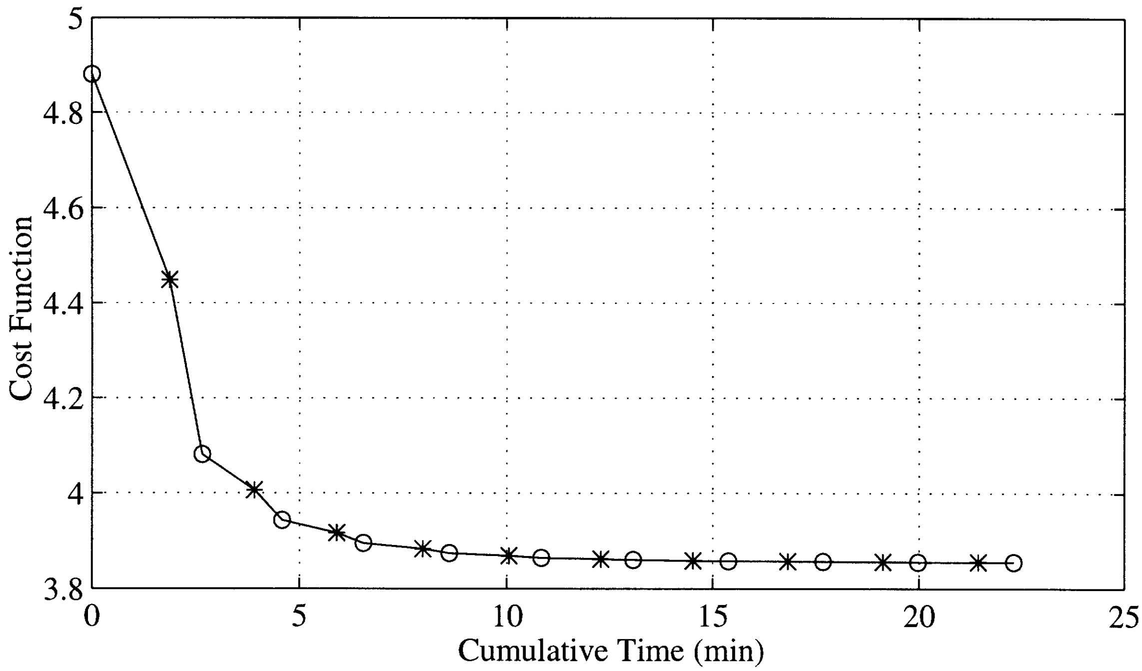

Figure 6-6: Cost function of the four mass problem vs. time. Cost after a synthesis step denoted by ∘; cost after an analysis step denoted by *.

the cost is size 16. The Riccati equations in the synthesis algorithm are only size 8. Because the time required to solve a Riccati equation is proportional to the cube of its size, the analysis Riccati equation solution is significantly slower.

The time required for each step to solve is also examined in Figure 6-6, which shows the cost function plotted after each step. From the plot, it is clear that the analysis steps require more time to solve. However, what the plot also shows is that the analysis and synthesis halves of the $D-K$ iteration are *both* critical to minimizing the cost function. The net effect of all the analysis steps on the cost is roughly equal to the net effect of all the synthesis steps. For this problem, at least, one cannot hope to find a global minimum for the cost function by performing synthesis for a fixed set of stability multipliers.

It should also be mentioned that the resulting $\mathcal{H}_2$/Popov controller design is essentially identical to the controller designed by Livadas. The poles of both controllers and the optimal stability multipliers are listed in Table 6.4. Most of the real and imaginary parts of the poles agree to better than ± 0.01. The Bode plots of the two

Parameter	Separation Based	LMI Based
Controller Poles	$-0.0375 \pm 1.3902j$	$-0.0382 \pm 1.3904j$
	$-0.5454 \pm 2.3348j$	$-0.5541 \pm 2.3227j$
	$-2.4347 \pm 1.5462j$	$-2.4445 \pm 1.5455j$
	$-2.4117 \pm 2.8750j$	$-2.4138 \pm 2.8774j$
h_i	1.34	1.36
	2.68	2.75
n_i	0.51	0.54
	1.20	1.22
bound on cost	3.8559	3.8555

Table 6.4: Solution for the 4 mass problem. The LMI-based results are from Reference 57, Table 4.13.

controllers are indistinguishable. The multiplier values, particularly in percentage terms, do not agree quite as closely. However, given the close agreement of the bound on the cost between the two solutions, this seems to indicate that the solution is relatively insensitive to the choice of the multipliers. The overall agreement between these solutions is evidence that, for this problem, there are no local minima. The results for this system at a lower level of uncertainty, as reported by Livadas in Ref,[57] Table 4.13 (who compared his results to that of How[46,48]) are in general agreement with this finding.

6.3.2 SISO MACE System

Our second example is taken from an actual structural control experiment. It provides us with the opportunity to do a robust controller design for a practical system. The system is the SISO model of the MACE system discussed in Appendix B. The plant has 24 states. The uncertainties come from 4 uncertain modal frequencies. Again, these are real, parametric uncertainties. Since we use Popov multipliers, this implies that there are 8 scalar parameters, $h_i, n_i, i \in [1 \ldots 4]$, that must be optimized in each analysis step.

Unfortunately, this problem is large enough such that the synthesis algorithm based on the LMI results of Chapter 4 cannot handle the system. Nor can the

Parameter	Value
$cost\_tol$	1×10^{-2}
min_iter	3
x_{tol}	-1×10^{-5}
$Jsynth\_tol$	1×10^{-5}
$anvar\_tol$	2.5×10^{-1}
$ancost\_tol$	1×10^{-2}
an_max	50

Table 6.5: Convergence criteria for the SISO MACE Problem

analysis routine based on LMIs run. The LMILab code requires too much memory. Therefore, we cannot give the type of comparative results that we presented for the 4 mass problem. However, $\mathcal{H}_2$/Popov controllers are designed for the SISO MACE system in the work by How *et al.* in Refs. 46 and 50. We attempt to replicate a design presented in these works. The level of uncertainty that we choose to concentrate upon, $M_2 = -M_1 = 0.02I$, is the most thoroughly discussed level of uncertainty in the these references.

From the results of the 4 mass problem, it can be seen that the cost of the analysis problem can be relatively insensitive to the choice of the multipliers. Therefore, requiring a high degree of precision in the choice of the multipliers is felt to be unnecessary. This is reflected in the choice of the convergence parameters, which are listed in Table 6.5.

All synthesis steps for the SISO MACE example were performed using a CG iteration. The CG iteration typically converged in under 15 steps. No convergence problems were encountered at this level of uncertainty. The analysis steps were performed using the SQP routine. For this problem, the line search routine was found to be excessively slow. Too many points were tried before the routine found one that lowered the cost function. Therefore, the line search routine was changed such that it evaluated the cost of only every second point that it would have originally considered. As a result, the modified SQP routine was found to be much more effective. The initial line search in each analysis step searched between 9 and 11 points, but

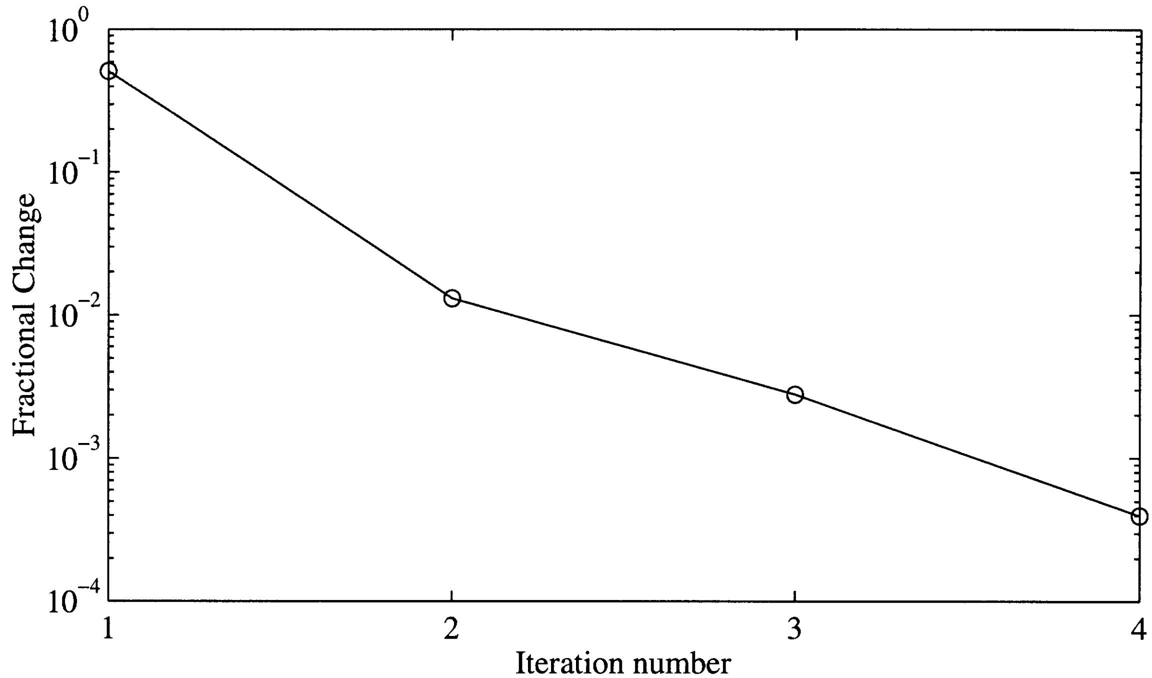

Figure 6-7: Convergence of the SISO MACE problem. Fractional change in the cost function after each iteration. The extra iteration that was run to check convergence is shown.

subsequent searches required three or fewer points. Furthermore, each analysis step converged before it had reached *an_max* evaluations of the cost function.

The $D - K$ iteration for this problem was initialized at $H = I$ and $N = 0$. The overall $D - K$ iteration converged in only three iterations. However, to double check that the solution had actually converged to the correct solution, the iteration was forced to restart from the solution point with $anvar\_tol = 1 \times 10^{-2}$ and $ancost\_tol = 5 \times 10^{-3}$. After completing the required extra iteration, the routine again exited, indicating that it had indeed converged. Further precision in the cost function was deemed unnecessary. The change in the cost function as a function of iteration is shown in Figure 6-7. Because the plot decreases monotonically, it seems clear that the $D - K$ iteration is converging to a solution.

The synthesis step of this design converged more rapidly than we might have predicted based on the size of the system. As shown in Table 6.6, the average synthesis time for the problem was only 44 seconds. The SISO MACE system has three times as many states as the previous, 4 mass problem. In section 6.1, we had predicted

Total Solution Time	# of Synthesis Steps	Avg. Synthesis Time	# of Analysis Steps	Avg. Analysis Time	Time per Full Iteration
12:00	4	0:44	3	3:16	4:00

Table 6.6: SISO MACE Problem: Solution time (min:sec) using the $D-K$ iteration with a separation-based synthesis step. Data does not include the final, extra iteration that was run as a check on convergence.

that the synthesis algorithm would have solution times proportional to n^4. However, the 4 mass problem required, on average, 8 seconds to converge. Clearly, an accurate calculation of convergence time for the synthesis algorithm must be based on much more than the size of the system. One of the variables affecting synthesis time is, as discussed in Section 6.1.2, the relative size of the uncertainty. Therefore, we do not actually learn anything about the workings of the synthesis method by trying to compare the results for the two examples.

Also shown in Table 6.6 is the total time of solution for the problem and the average time required to perform an analysis step. Again, we find that analysis takes up the majority of the design time. This is primarily due to the time required to solve the larger, closed loop Riccati equations. As in the previous example, we can plot the cost function as a function of time. This is shown in Figure 6-8. The plot graphically demonstrates how much longer the analysis steps require to solve than the synthesis steps. However, the plot also seems to indicate that the extra time needed by the analysis steps is worthwhile, since it is the first analysis step that seems to have the greatest effect at reducing the cost function. What the plot does not show is the change in the cost function achieved by the initial synthesis. The MACE system is open loop stable. The open loop system has a cost of approximately 17.1. The initial synthesis reduces this to approximately 3.7.

We would like to compare the design time that this methodology required to the time required by the routine of Ref. 50. Unfortunately, we do not know how much time the routine in the referenced work required to minimize the augmented Lagrangian.

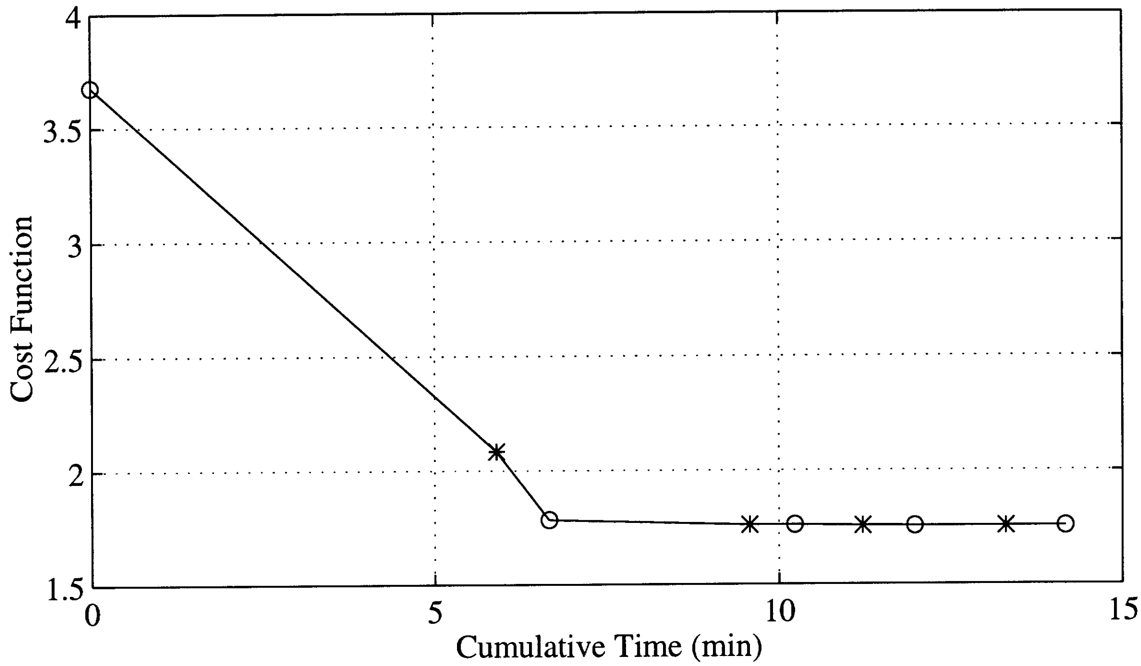

Figure 6-8: Cost function of the SISO MACE problem vs. time. Cost after a synthesis step denoted by ∘; cost after an analysis step denoted by *.

However, it is estimated[47] that, after accounting for differences in computational speed, the augmented Lagrangian approach required a *minimum* of between 30 and 45 minutes to reach its solution. This does not include the time required to find an initial condition to start the minimization. It is important to recognize, however, that in comparing solution times we have implicitly presumed that the two solutions are equivalent. To see whether the two solutions to the SISO MACE problem actually are equivalent, we can examine the stability multipliers, the bound on the $\mathcal{H}_2$ cost, and the controllers derived in each case.

The stability multipliers derived using both methodologies are listed in Table 6.7. Note that in the referenced solution, the last entry of the H matrix was *a priori* set to unity. This can be done without any loss of generality, and it has the benefit of reducing the size of the optimization by one variable. Therefore, if the optimal scales are the same, then, for comparison, our scales should be normalized such that the fourth entry of H is equal to unity as well. However, even after normalizing the scales, it is clear that the two solutions are different.

Parameter		Separation and $D-K$ Iter. Based	Aug. Lagrangian Based
multiplier	mode (Hz)		
h_i	8.83	9.32×10^{-2}	4.66
	9.40	4.10×10^{-2}	8.39
	13.30	1.39×10^{-1}	11.83
	13.88	4.51×10^{-2}	1.00
n_i	8.83	8.83×10^{-4}	0.050
	9.40	1.98×10^{-4}	0.043
	13.30	1.61×10^{-3}	0.026
	13.88	2.79×10^{-4}	0.082
cost/(open loop cost)		0.103	0.11

Table 6.7: Optimal multipliers for the SISO MACE problem. Results for the augmented Lagrangian case are shown to the same precision as given in Tables II and III for controller Gpc4 in Ref. 50. Open loop cost is 17.1.

The minimum value of the cost function, $\mathcal{J}$, attained via each design methodology is also listed in Table 6.7. Although Ref. 50 only lists the bound on the cost to two significant figures, it is clear that new methodology yields a lower bound on the cost. Because the goal of this design is to minimize the cost function, this implies that the controller of Ref. 50 is inferior. Having said this, since it is known that the cost function is a conservative estimate of the square of the $\mathcal{H}_2$ performance, this does not necessarily imply that the performance of the controllers will be different in practice. This is examined next.

The Bode plot for the robust $\mathcal{H}_2$ controller designed using the $D-K$ iteration is plotted in Figure 6-9. Also shown are the weighted Bode plots for the plant and the non-robust $\mathcal{H}_2$ controller for the nominal system (LQG). The robust controller appears to be very similar the LQG controller. The main difference is that some of the poles and zeros have different amounts of damping and that the DC gain of the robust controller is slightly decreased. This plot should be compared to Fig. 14 in Ref. 50, which shows the Bode plots for the same weighted plant, the same LQG controller, and the SISO robust controller designed in that reference. That robust

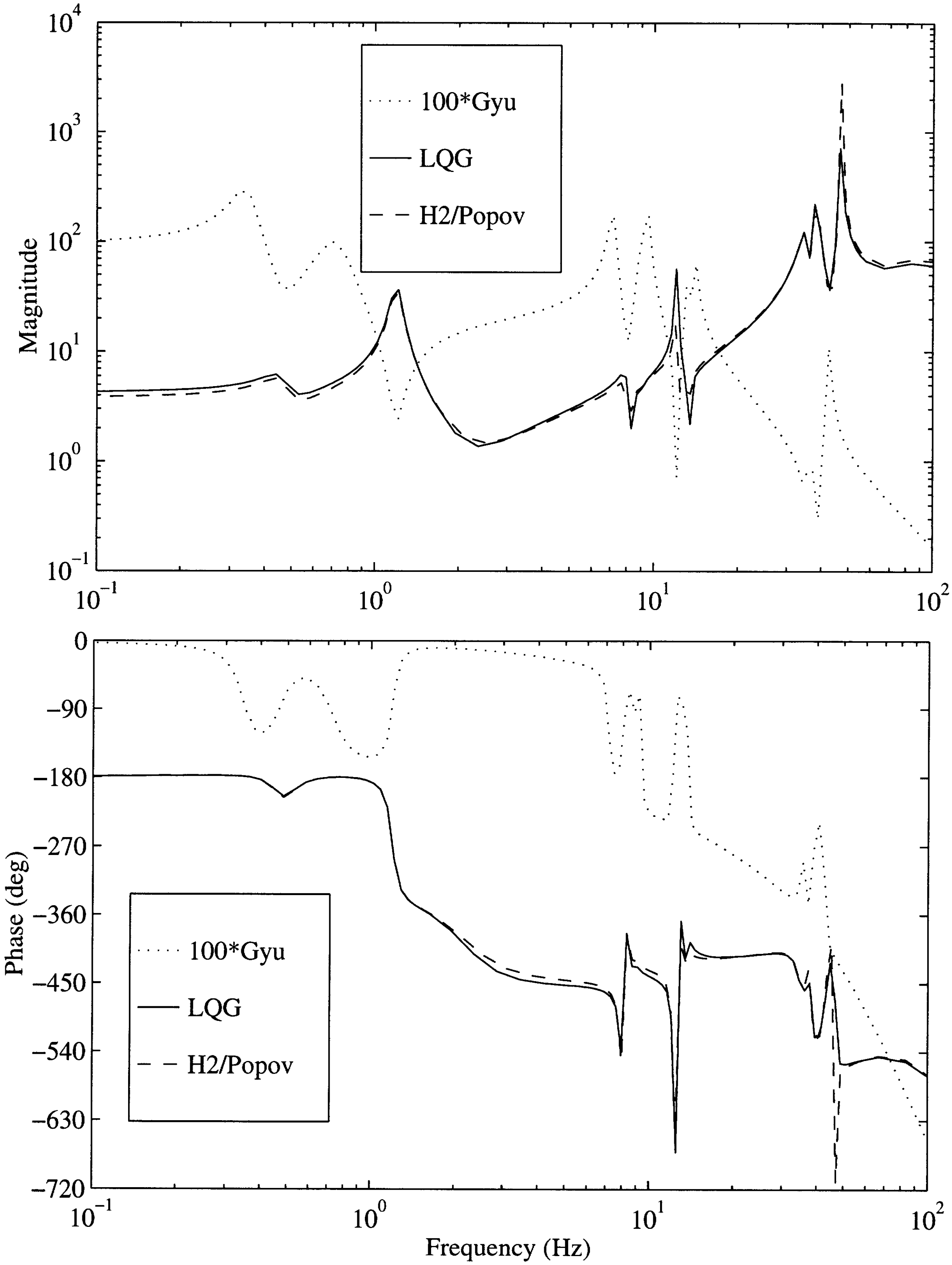

Figure 6-9: SISO MACE Problem: Bode Plots

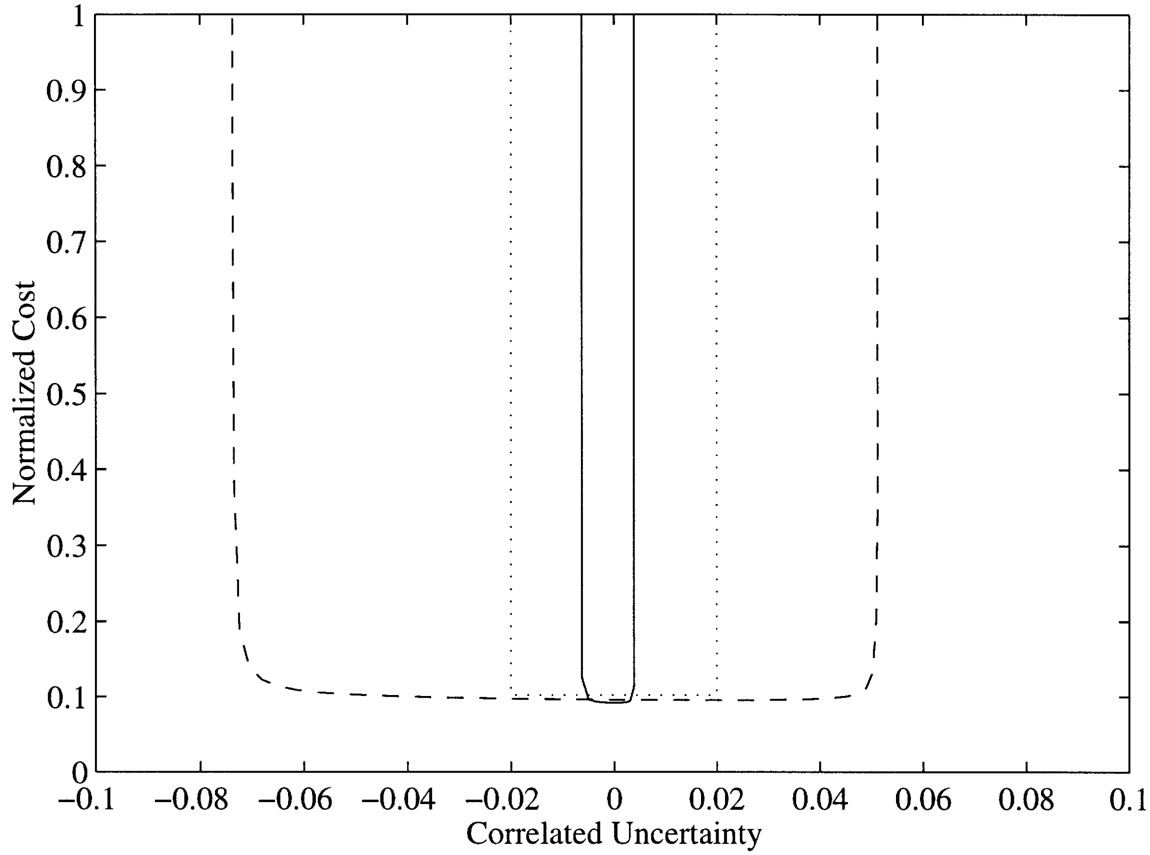

Figure 6-10: SISO MACE Problem: Performance at various levels of a correlated uncertainty. Robustness guaranteed to ± 0.02. Cost is normalized by the open loop cost. Solid line = system with LQG controller. Dashed line = system with $\mathcal{H}_2$/Popov controller. Dotted line = performance guaranteed by an $\mathcal{H}_2$/Popov controller. The optimal LQG cost is 0.0922. The actual $\mathcal{H}_2$/Popov controller performance for zero uncertainty is 0.0962.

controller is significantly different from the LQG controller. It, for instance, does not exhibit the peak at approximately 0.5 Hz, nor the same pole-zero pattern between 7 and 14 Hz. We conclude, then, that the two solutions for the SISO controller are significantly different.

The robustness of the controller designs can also be examined. Given a particular perturbation to the nominal system, we can calculate the exact $\mathcal{H}_2$ performance of the system, rather than just a bound on the performance. This gives an indication of how conservative our designs may be. We examine the case where the stiffness changes in the four modes are exactly correlated, *i.e.*, they are either all 1% high together, or 2% low together, etc.. Of course, this is not an inclusive test of all the worst case uncertainties, but it is a useful and practical measure of the robustness and conservatism. The exact performance of the perturbed system with the controller from the $D - K$ iteration is plotted in Figure 6-10. Also shown is the level of performance that was guaranteed by the $\mathcal{H}_2$/Popov design and the exact performance of the LQG controller. Notice that the range of stiffnesses for which the LQG controller can stabilize the system is very narrow. It is too highly tuned to the nominal system. For most perturbation sizes, the LQG controller significantly underperforms the guaranteed performance level of the robust controller. As expected, the $\mathcal{H}_2$/Popov controller achieves a significantly wider range of robustness. The controller was designed to be robust to $\pm 2\%$ variations in stiffness. In fact, the $\mathcal{H}_2$/Popov controller maintains a level of near-optimal performance over more than twice the required range. This is a measure of the conservatism of the controller for a correlated uncertainty. However, the achieved performance using the robust controller is close to the bound throughout the range of greatest interest, $[-0.02, 0.02]$. Also, at the nominal position, both the achieved performance of the robust controller and the bound on its performance are closer than 10% to the performance of the LQG controller. This means that the controller has not significantly sacrificed performance for the sake of robustness.

A similar plot was shown in Ref. 50. For perturbations in the $\pm 2\%$ range, the controller from Ref. 50 achieves inferior performance compared to the controller designed using our techniques. Furthermore, the controller from Ref. 50 maintains a flat level of performance for perturbations ranging in size from 0.1 down to below (-0.25). It is considerably more conservative than the new controller. The controller from the reference has achieved greater robustness bounds against a correlated uncertainty, but this has been at the expense of decreased performance. Furthermore, the designer typically will not have the ability to calculate such robustness margins, so this extra robustness is left unknown. The only information that is known is the bound, and the new controller design has achieved a superior bound.

As an aside, it should be noted that even though the design from Ref. 50 is different than the design performed using the $D - K$ iteration, this is not necessarily

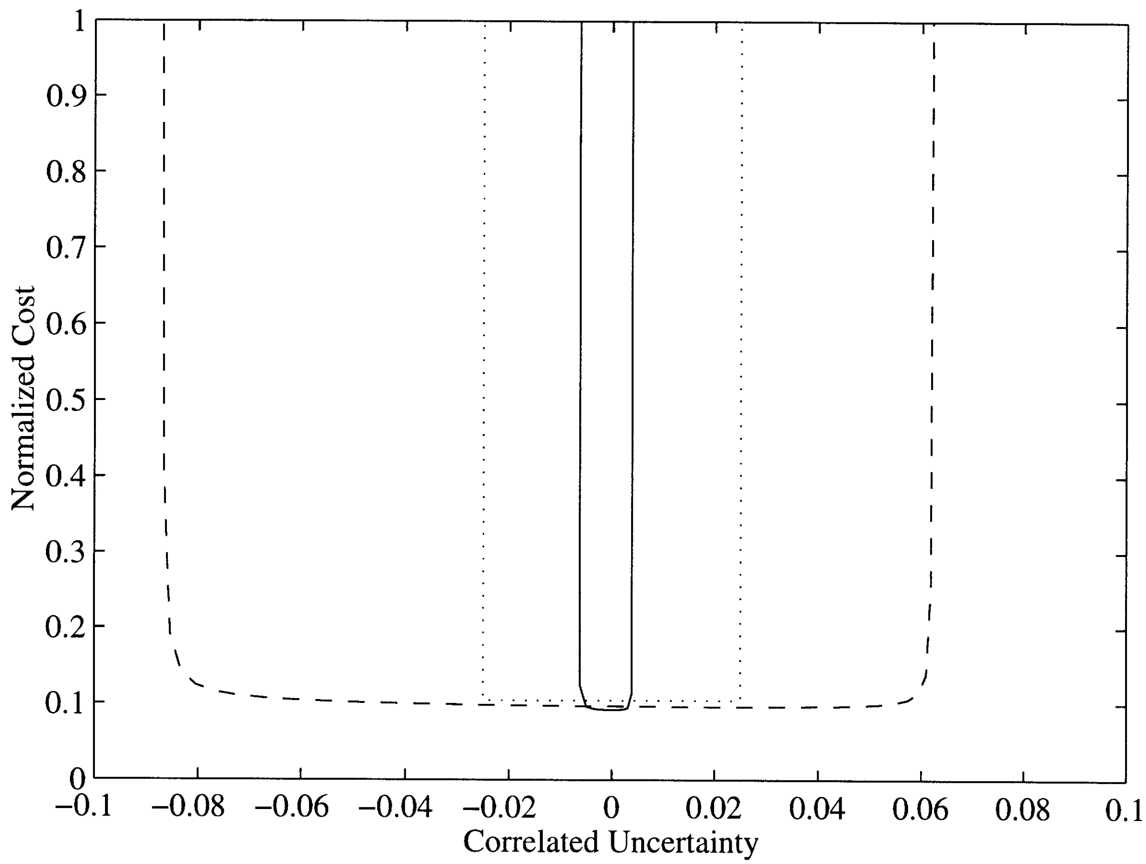

Figure 6-11: SISO MACE Problem: Performance at various levels of a correlated uncertainty. Robustness guaranteed to ± 0.025. Cost is normalized by the open loop cost. Solid line = system with LQG controller. Dashed line = system with $\mathcal{H}_2$/Popov controller. Dotted line = performance guaranteed by an $\mathcal{H}_2$/Popov controller. The optimal LQG cost is 0.0922. The actual $\mathcal{H}_2$/Popov controller performance for zero uncertainty is 0.0969.

an indication that the optimization problem has local minima. It could have local minima. However, it could also be the case that the augmented Lagrangian algorithm should have been run for a greater amount of time.

If desired, the robustness margins for this controller can be increased. Because our design for the $\mathcal{H}_2$/Popov controller at the $\pm 2\%$ level of uncertainty seems relatively non-conservative, the stability multipliers from this design can be used as an initial condition for a design at a larger level of uncertainty. A controller was designed for a system with a 10% larger uncertainty, *i.e.*, $M_2 = -M_1 = 0.022I$. This is the maximum level of uncertainty that appears in Ref 50. This design required less than 9 minutes to converge. To demonstrate that the level of uncertainty could be increased even further, a final design was performed for $M_2 = -M_1 = 0.025I$, using the solution from the previous case as the initial condition. This design required just over 7 minutes to converge. The Bode plot of this final controller was again almost identical to the LQG controller. The controller achieved a cost function of 0.105 times the open loop cost. The actual performance of the final controller on a perturbed system was also calculated. It is shown for the case of a correlated uncertainty in Figure 6-11. This latest controller guarantees a robustness margin that is 25% wider than the controller of Ref. 50 and is still less conservative.

For this system, it is clear then, that the $D - K$ iteration approach with a separation-based synthesis algorithm offers a significant benefits over a method that relies on a minimization of the augmented Lagrangian. The $D-K$ iteration converged more quickly and to a more optimal value of the cost function. The close similarity between the new $\mathcal{H}_2$/Popov controller and the LQG controller demonstrates that even subtle changes can have a dramatic effect on the robustness of a design.

6.3.3 MIMO MACE System

The third example system demonstrates that our control design methodology can be used for large systems. The system is a MIMO model of the same MACE system used in the previous example. The model is discussed in more detail in Appendix B. Unlike the SISO version of this system, however, the current controller has three control signals and three sensor inputs. The plant has 59 states (compared to 24 in the SISO model) and a commensurately larger set of uncertain modal frequencies. This time there are 11 uncertain frequencies (compared to 4 in the SISO model), which means that there are 22 scalar parameters, $h_i, n_i, i \in [1 \dots 11]$, that must be determined at each analysis step. Each uncertain frequency is known to vary within a certain percentage from its nominal value. The percentage of possible frequency shift varies between ± 2.2 and ± 6.7 percent, depending on the mode of interest. The exact uncertainty parameters are listed in the Appendix.

As was already found for the SISO model of this system, the MIMO model is too large for our LMI-based synthesis and analysis routines. However, we can compare our design to the $\mathcal{H}_2$/Popov controller discussed by How *et al.* in Refs. 50 and 46. Recall that the controller design discussed in these works was found by minimizing the augmented Lagrangian discussed in Section 6.1. To initialize the minimization, a controller was chosen from a multiple model controller design[38,39] for the system and combined with guesses for the optimal scalings. Ref. 50 presents the optimal Popov scalings (H and N) that were found using this and presents some experimental results found using the controller. It does not, unfortunately, present the bound on the cost that is guaranteed by the controller nor the time required by the algorithm to solve the problem.

To start our design algorithm, we needed to determine initial conditions for the Popov multipliers, H and N, that could enable the controller synthesis routine to converge. As was done for the previous examples, we wished to guess a simple set of initial conditions. This would highlight the relative ease with which one could start the $D-K$ iteration. Unfortunately, it was found that the problem could not be initialized starting from $H = I$ and $N = 0$. Furthermore, no scalar value, say k, was found such that $H = kI$ and $N = 0$ could initialize the algorithm. Therefore, it was decided, instead, to solve the problem with a continuation method wrapped around the $D-K$ iteration.

In the continuation method, an initial problem is considered that has a reduced level of uncertainty. A robust controller is designed for this degenerate system using our $D-K$ iteration. Then, the resulting multipliers are used to initialize a design at a higher level of uncertainty. This process is repeated until the required level of uncertainty is reached.

For the MIMO MACE plant, it was found that $H = I$ and $N = 0$ did not initialize the problem above approximately 50% of the required uncertainty level. However, a choice of $H = 2{\cdot}\mathrm{diag}\,(1, 2, \ldots, n_w)$ and $N = 0$ initialized a design at 65% of the required uncertainty, *i.e.*, using $M_2 \to 0.65M_2$ and $M_1 \to 0.65M_1$. The 65% case was used as the first of three separate designs in the continuation method. This was approximately the highest percentage of uncertainty for which the CG iteration con-

Parameter	Value
cost_tol	1×10^{-2}
min_iter	3
x_{tol}	-1×10^{-5}
Jsynth_tol	1×10^{-5}
anvar_tol	2.5×10^{-1}
ancost_tol	1×10^{-2}
an_max	50

Table 6.8: Convergence criteria for the MIMO MACE problem at 65% of the desired uncertainty

Parameter	Value
cost_tol	1×10^{-2}
min_iter	3
x_{tol}	-59 $(= -n)$
Jsynth_tol	7.5×10^{-3}
anvar_tol	2.5×10^{-1}
ancost_tol	1×10^{-2}
an_max	50

Table 6.9: Convergence criteria for the MIMO MACE problem at 82.5% of the desired uncertainty

Parameter	Value
cost_tol	1×10^{-2}
min_iter	3
x_{tol}	-45 $(\approx -0.75n)$
Jsynth_tol	4×10^{-4}
anvar_tol	2.5×10^{-1}
ancost_tol	1×10^{-2}
an_max	50

Table 6.10: Convergence criteria for the MIMO MACE problem at 100% of the desired uncertainty

verged using relatively simple choices for H and N. The convergence parameters for this design are listed in Table 6.8. After the initial design, the size of the uncertainty was increased by half of the required amount to 82.5% of the required uncertainty. The CG iteration was not convergent for the 82.5% case using the multipliers generated in the 65% case. Therefore, this design was run using a synthesis algorithm that employed the CGR iteration. A quick set of tests was run to find a set of parameters α_1 and α_2 that allowed the CGR iteration to converge quickly. Eventually, $\alpha_1 = 10$ and $\alpha_2 = 1$ were chosen. These values were used at each step of this design. The convergence parameters for the design at 82.5% are listed in Table 6.9. Finally, the optimal multipliers resulting from the 82.5% case were used to start a design for a system with 100% of the required uncertainty. Again, a CGR iteration was employed for the synthesis step. A quick check of a few values for α_1 and α_2 allowed us to settle upon using $\alpha_1 = 9$ and $\alpha_2 = 1$ for this design. These values were used at each step of the the design. The convergence parameters for the design at 100% are listed in Table 6.10.

The synthesis algorithms performed well at the three levels of uncertainty. At 65% of the required uncertainty, the CG iteration typically required less than 15 steps to converge. Similarly, at the two higher uncertainty levels, the CGR iteration was always found to finish in fewer than 17 steps.

It should be noted, however, that the stopping criteria used with the CGR iteration were less strict than those used with the CG iteration. The x_{tol} and *Jsynth_tol* parameters were chosen to allow the CGR synthesis algorithm to result in relatively suboptimal controllers. This change was motivated by the desire to speed up the overall $D - K$ iteration. It turns out, however, that this course of action was somewhat misguided. Speeding the synthesis of the controllers did not speed the overall $D - K$ iteration significantly since it was the analysis portion of the iteration that took up the majority of the design time. This is made clear in data that is presented later. Secondly, the stopping conditions for the CGR iteration can be relaxed too much. The CGR iteration synthesis was used for the design at 82.5% without incident. However, after this design was completed, x_{tol} and *Jsynth_tol* were tightened down to try and prevent overly suboptimal controllers from being synthesized. It turned out, as shall be seen in a plot presented later, that the parameters should have been tightened down further to ensure that the $D - K$ iteration yielded a cost that decreased monotonically with each step.

The analysis routine was less successful with the MIMO MACE designs than it was with the previous examples. The problem in the analysis routine was that the closed loop Riccati equation had 118 states. Certain Riccati equation routines exhibited numerical precision problems with this system. This is discussed later in the chapter. An obvious problem with Riccati equations of this size is that they are slow to solve. This makes the analysis routine significantly slower that the synthesis routine. The final problem encountered with these three MIMO designs is that the analysis routine's line search was inefficient. The analysis algorithm spent far more time in the line search portion of the routine than it did performing gradient and Hessian evaluations. Tests showed that, given a maximum of 50 cost function evaluations, an entire analysis algorithm typically found fewer than 5 points with successively lower costs. To alleviate this problem, the line search routine was modified to evaluate only every second point that it would have originally considered. This is the same modification that was made for the SISO MACE problem. Nevertheless, not once in any of the designs at the three uncertainty levels did the analysis routine exit because it had converged, rather than because it had reached the maximum number of function evaluations.

The convergence of the $D - K$ iteration is governed by the fractional change in the cost after each iteration. This is plotted for each of the three designs in Figures 6-12, 6-13, and 6-14. In the design for the 65% case, the design converges after 6 iterations. The fact that the plot of Figure 6-12 is *not* monotonic should not necessarily surprise us. The design problem is not convex, and time to solution can be highly path dependent. Turning to the design at 82.5%, Figure 6-13 shows that the initial condition based on the 65% solution allows the $D - K$ iteration to converge quickly to a solution. The algorithm exits after only 3 iterations. It would have exited sooner were it not for the requirement that it iterate at least 3 times.

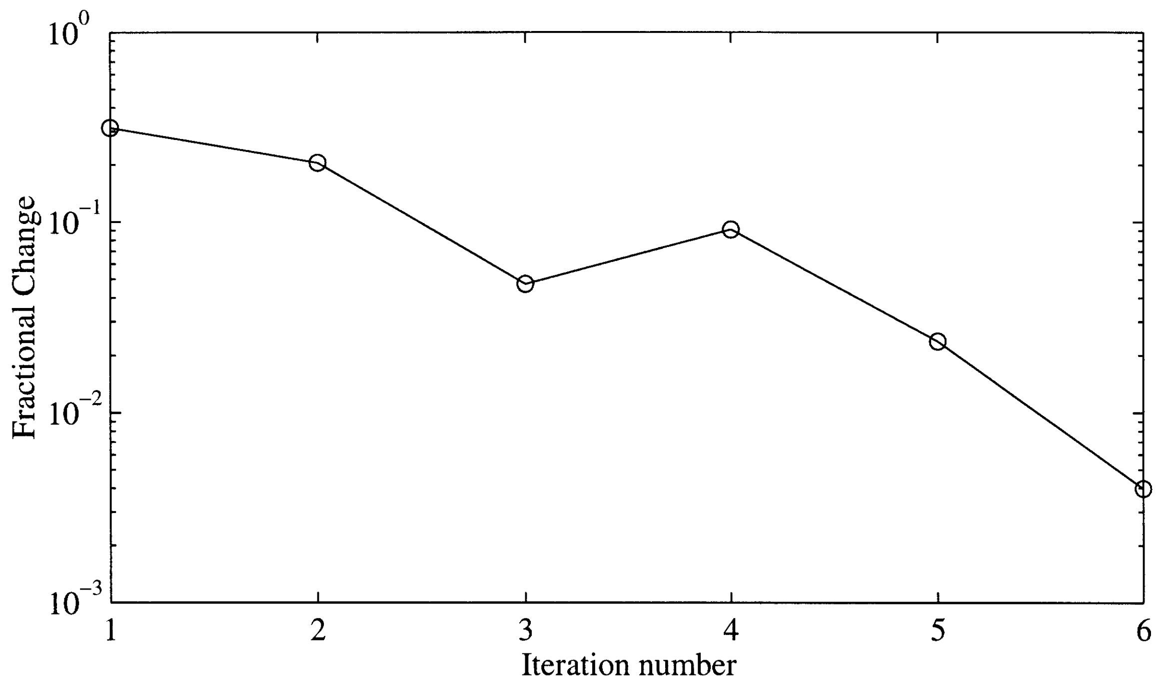

Figure 6-12: MIMO MACE Problem: Convergence at 65% of the required uncertainty. Fractional change in the cost function after each iteration.

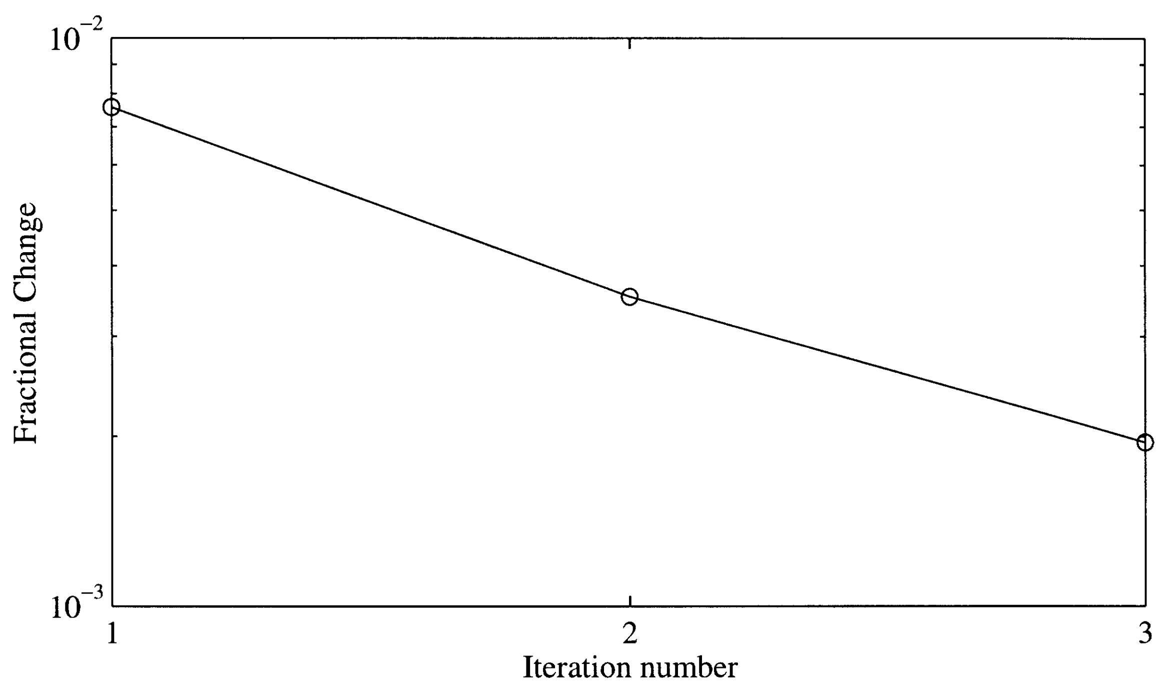

Figure 6-13: MIMO MACE Problem: Convergence at 82.5% of the required uncertainty. Fractional change in the cost function after each iteration.

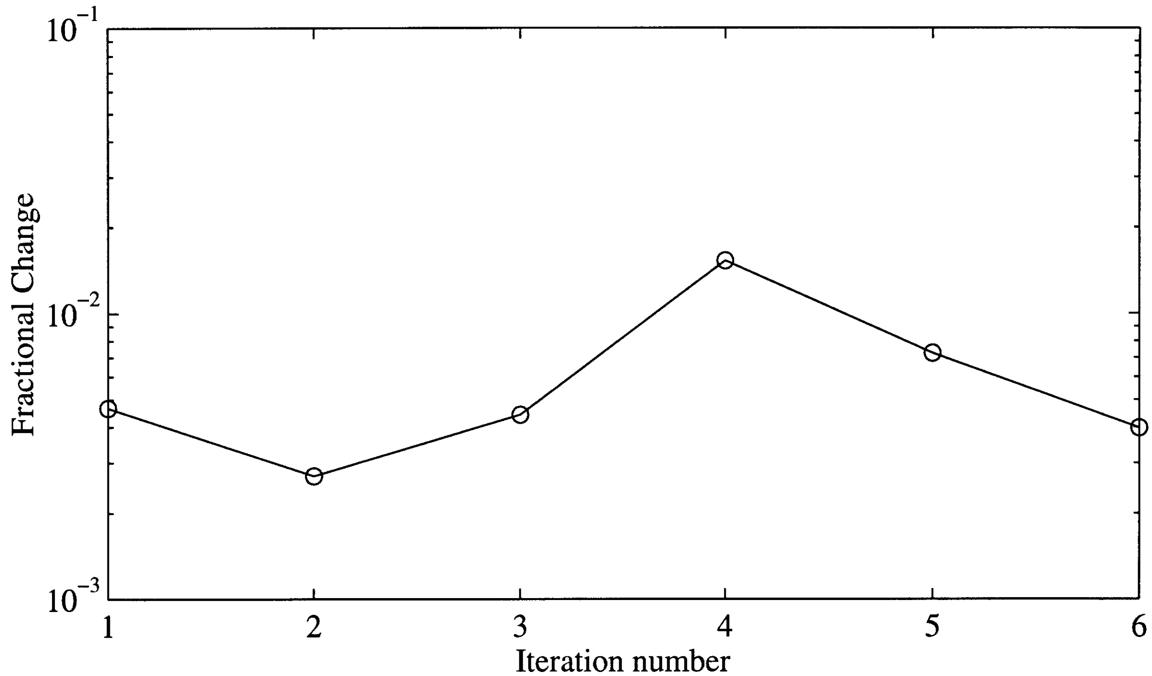

Figure 6-14: MIMO MACE Problem: Convergence at 100% of the required uncertainty. Fractional change in the cost function after each iteration.

Finally, we can discuss the design at 100% by examining Figure 6-14. Like the case at 65%, this design did not converge smoothly to its final value. It took its largest step in the fourth iteration and finally converged at the fifth iteration. Because the convergence was not smooth, to double check that the routine had found a minimum, it was restarted after the fifth iteration. However, it immediately exited again after one more iteration, indicating that it had found a minimum.

We would like to understand how design time using the $D-K$ iteration scales with problem size. It should be more useful to compare the design times of the two MACE models than it was to compare the SISO MACE model to the 4 mass model, because the two MACE models are of the same structure. Although the MIMO system has more control inputs and sensors, the underlying dynamics of the models are roughly the same (pole-zero patterns, major modal frequencies, etc.). We intuitively feel that if a problem was exhibited in the smaller model then it should be exhibited in the larger model. Therefore, we will compare the two models, but acknowledge that any correlations could be imaginary rather than real.

Percentage of Required Uncertainty	Total Solution Time	# of Synthesis Steps	Avg. Synthesis Time	# of Analysis Steps	Avg. Analysis Time
65	5:10:49	7	0:07:23	6	0:44:25
82.5	2:34:33	4	0:07:27	3	0:44:05
100	4:11:28	6	0:07:28	5	0:42:50
Overall	11:56:50	17	0:07:26	14	0:43:47

Table 6.11: MIMO MACE Problem: Solution time (hr:min:sec). Data for the design at 100% does not include the final, extra iteration that was run as a check on convergence.

The time required for controllers to be designed at each of the three levels of uncertainty is shown in Table 6.11. The overall MIMO design requires approximately 12 hours of computational time. This seems long compared to the 12 minute design time required for the 24 state SISO model, shown in Table 6.6. However, the long MIMO design time is due primarily to the fact that a continuation method was used. If the MIMO system could have been solved in one step, as the SISO problem was, then the design time would essentially be cut by two-thirds, down to perhaps 4 hours. Four hours is 20 times longer than was required for the SISO system. Note that if design time scales with n^3 for n states, then the MIMO case should take 15 times longer than the SISO problem. In contrast, if the design time scales with n^4, then the MIMO case should take 37 times longer than the SISO problem. Therefore, for this system, bringing a single $D-K$ iteration to convergence is a problem that seems to scale at a rate better than n^4.

It is also useful to discuss how size affects the solution time of the individual synthesis and analysis algorithms, rather than the $D-K$ iteration as a whole. The CG iteration required about 7.5 minutes to synthesize controllers for the MIMO MACE problem. This is only 10 times more than was required for the SISO problem. That is better than a rate of n^3. Unfortunately, a discussion of the time required to solve the analysis problem is probably not valid (using this example), since the analysis routines in the MIMO cases failed to converge.

The data in Table 6.11 reemphasize the fact that the analysis routine requires

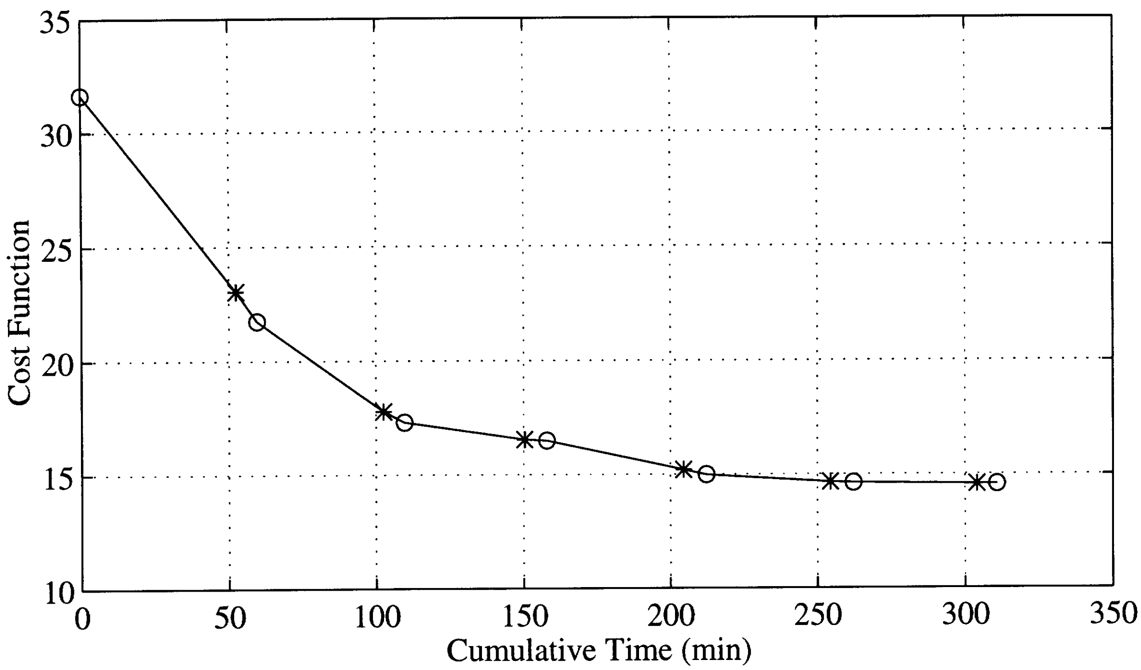

Figure 6-15: Cost function of the MIMO MACE problem vs. time at 65% of the required uncertainty. Cost after a synthesis step denoted by ∘; cost after an analysis step denoted by *.

a far greater amount of time to solve than does the synthesis routine. Any significant improvement to the analysis portion of the iteration would be manifested as a significant improvement in overall design time.

The information in Table 6.11 is augmented by the plots in Figures 6-15, 6-16, and 6-17, which show how the cost in each design evolved with each step and with time. From these plots it is clear that the majority of the optimization was accomplished in the first design. The designs at 82.5% and 100% were started very near to designs yielding their respective final costs.

The plot in Figure 6-17 exhibits one surprising anomaly. It is not monotonically decrescent. The last two synthesis steps actually worsen the cost. This is due to the fact that the stopping criteria for the synthesis algorithm were overly relaxed, as mentioned earlier. The controllers found in the last two synthesis steps were suboptimal to the extent that they were inferior to what had already been obtained (for a fixed set of stability multipliers). It seems likely, then, that it would be useful for the stopping criteria in the synthesis step (and probably the analysis step) to be

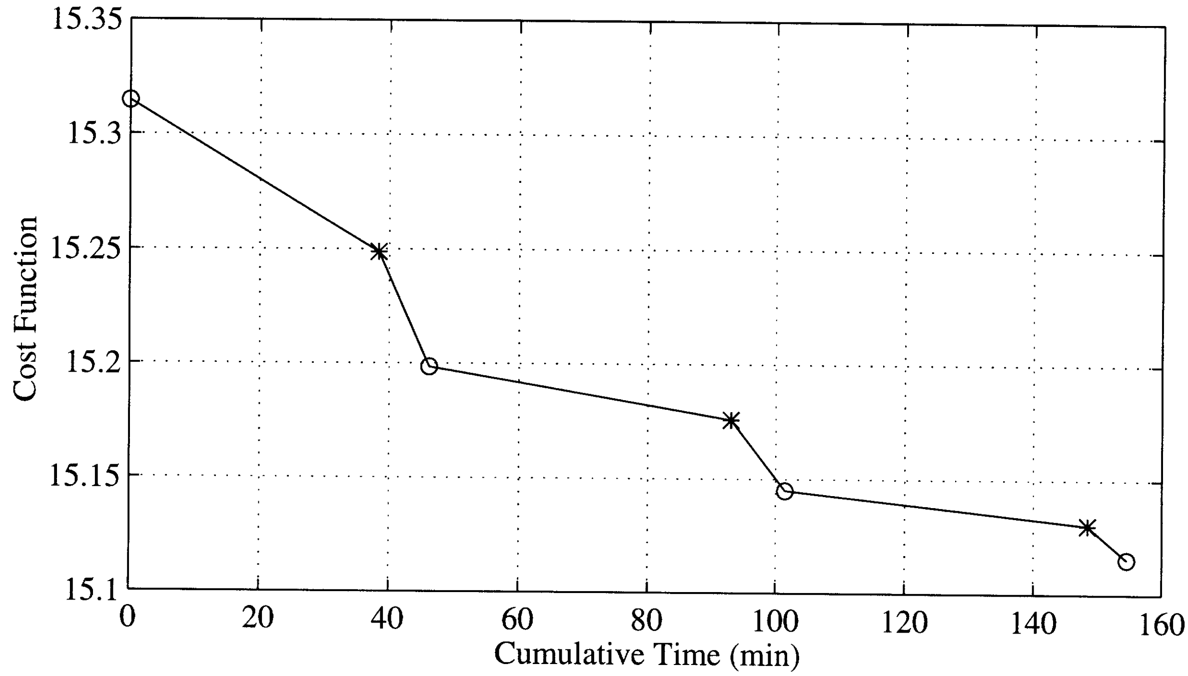

Figure 6-16: Cost function of the MIMO MACE problem vs. time at 82.5% of the required uncertainty. Cost after a synthesis step denoted by ∘; cost after an analysis step denoted by *.

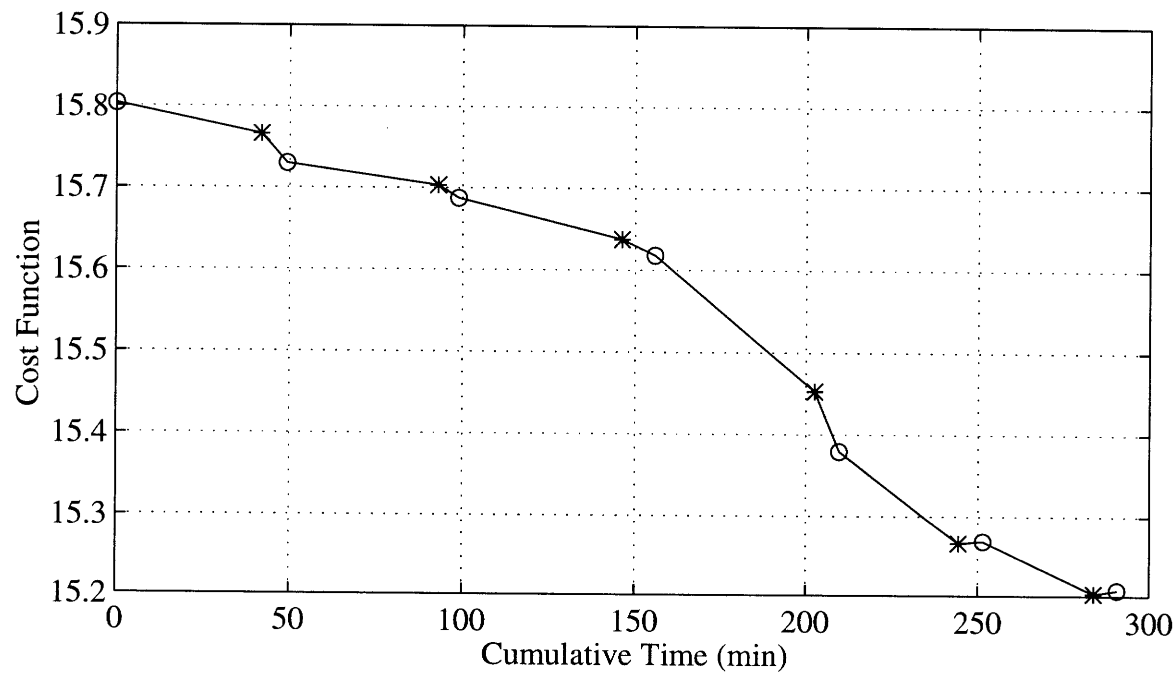

Figure 6-17: Cost function of the MIMO MACE problem vs. time at 100% of the required uncertainty. Cost after a synthesis step denoted by ∘; cost after an analysis step denoted by *.

made tighter as the $D - K$ iteration progresses. This would allow for more solutions to be found early on, but would not sacrifice the accuracy of the solution at the end.

The stability multipliers found for the three designs are listed in Table 6.12. From this data, it is clear that the multipliers do change significantly with the changing level of uncertainty. Furthermore, a close examination of the multipliers reveals one of the difficulties that would be encountered if we attempted to design the controller using a homotopy algorithm. The scalings do *not* vary linearly with the level of uncertainty. In fact, 6 of the 22 parameters increase between 65% and 82.5% and subsequently decrease between 82.5% and 100%, or vice versa. For a homotopy algorithm, this means that incremental increases in the uncertainty size must be conservative or any derivative information is useless. This leads to lengthy design times.

Table 6.12 also lists the multipliers from the design in Ref. 50 at 100% of the required uncertainty. In the reference, the last entry of the H matrix is fixed to be 20. After appropriately scaling our multipliers, we can see that the solutions for the multipliers are different. Unfortunately, we are unable to say whether or not the controllers from the two design methodologies are the same. Ref. 50 does not list a bound on the cost of the system. Instead, it gives the results of an experimental test conducted with the $\mathcal{H}_2$/Popov controller. The experimental implementation of the controller on the MACE system yielded a 12.4 dB improvement over the open loop system. The LQG controller yielded a 10.2 dB improvement. Similarly, for the controller produced using our design methodology, it theoretically guarantees that the closed loop $\mathcal{H}_2$ cost will be no more than 3.907, a 13.45 dB improvement over the nominal open loop cost of 18.38.

The square of the $\mathcal{H}_2$ cost for the nominal system can be written as

$$J = \int_{-\infty}^{+\infty} \text{tr } \tilde{G}_{ed}^*(j\omega)\tilde{G}_{ed}(j\omega)d\omega \, , \tag{6.25}$$

where $\tilde{G}_{ed}$ is the closed loop transfer function between the performance signal, e, and the disturbance term d, which is a white, zero mean noise signal with identity covariance matrix. Figure 6-18 contains plots of the square root of the integrand of

Table 6.12: Optimal multipliers for the MIMO MACE problem. Results for the augmented Lagrangian case are shown to the same precision as given in Table 7.6 of Ref. 46.

Mode (Hz)	Separation and $D-K$ Iteration Based						Aug. Lagr. Based	
	65% Uncert.		82.5% Uncert.		100% Uncert.		100% Uncert.	
	h_i	n_i	h_i	n_i	h_i	n_i	h_i	n_i
1.40	$1.13e-1$	$2.76e-3$	$1.17e-1$	$3.25e-3$	$1.62e-1$	$3.39e-3$	2.29	0.059
3.35	$4.60e-2$	$8.18e-4$	$5.35e-2$	$8.15e-4$	$5.15e-2$	$9.64e-4$	4.83	0.125
4.88	$1.40e-1$	$4.62e-3$	$1.38e-1$	$5.69e-3$	$1.13e-1$	$5.97e-3$	6.79	0.152
5.92	$1.41e-1$	$3.82e-3$	$1.09e-1$	$3.73e-3$	$1.40e-1$	$4.26e-3$	10.00	0.153
8.76	$8.41e-0$	$6.56e-2$	$8.18e-0$	$1.03e-1$	$6.52e-0$	$1.18e-1$	11.00	0.152
8.91	$7.33e-2$	$7.04e-4$	$7.75e-2$	$7.80e-4$	$1.08e-1$	$1.18e-3$	18.95	0.205
9.42	$2.00e-1$	$5.01e-3$	$2.31e-1$	$6.05e-3$	$2.52e-1$	$6.66e-3$	14.92	0.190
13.31	$2.99e-1$	$4.29e-3$	$2.21e-1$	$4.33e-3$	$2.00e-1$	$4.53e-3$	16.92	0.183
13.90	$3.44e-1$	$5.70e-3$	$2.68e-1$	$6.55e-3$	$2.40e-1$	$5.80e-3$	12.97	0.177
14.80	$1.34e+1$	$3.61e-2$	$1.30e+1$	$2.98e-2$	$1.07e+1$	$4.10e-2$	20.91	0.171
33.78	$1.11e+1$	$1.75e-2$	$1.05e+1$	$2.10e-2$	$6.26e+0$	$1.39e-2$	20.00	0.063

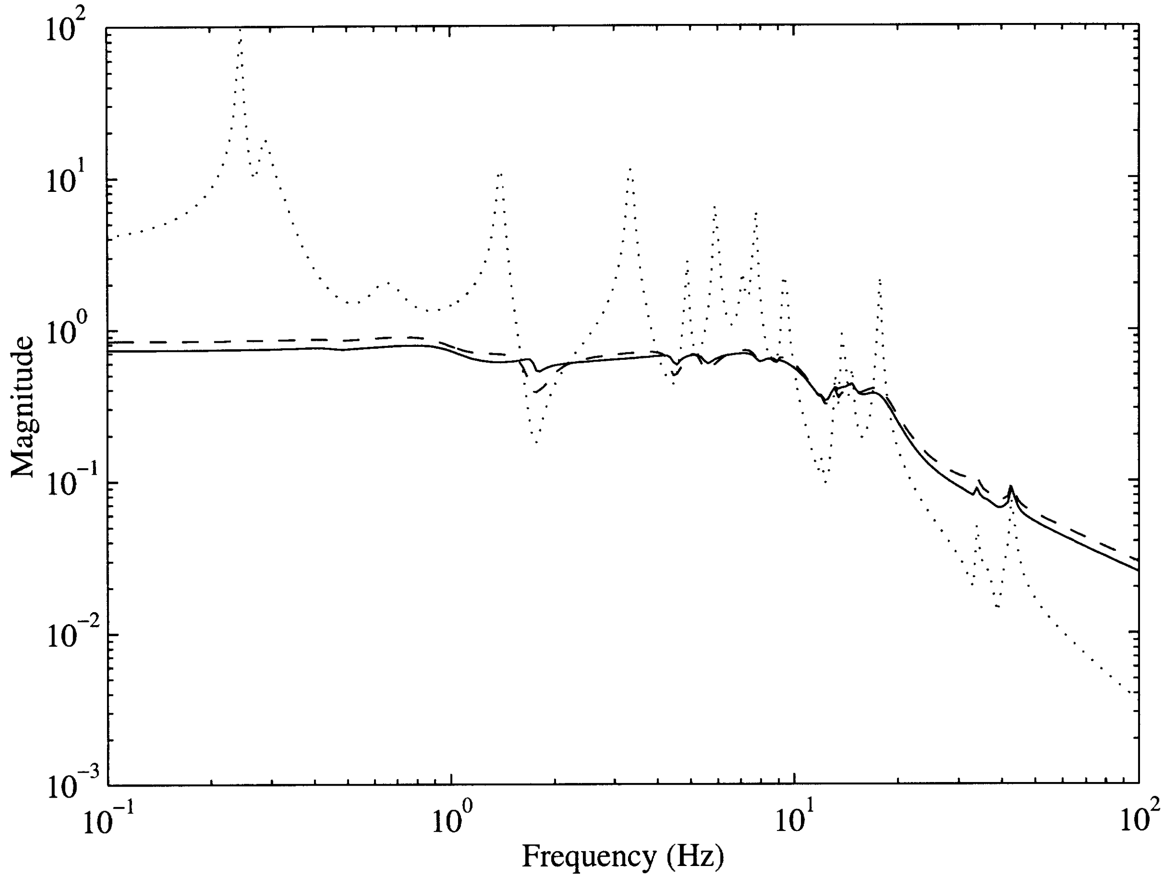

Figure 6-18: Cost function of the MIMO MACE problem vs. frequency. Nominal plant with $\mathcal{H}_2$/Popov controller designed for 100% of the required uncertainty. Dotted line = open loop; solid line = closed loop with LQG; dashed line = closed loop with $\mathcal{H}_2$/Popov.

this expression for the open loop system, the system with an LQG controller, and the system with the $\mathcal{H}_2$/Popov controller. The plot for the system with the $\mathcal{H}_2$/Popov controller is almost identical to the plot for the system with the LQG controller, except that the gain is slightly increased in most regions. A similar plot was shown in Ref. 50 for the $\mathcal{H}_2$/Popov controller designed in that reference. That plot was based on experimental data. Experimentally, the robust controller actually performed better than the LQG controller, because the experimental apparatus was not exactly equal to the desired nominal plant.

6.3.4 Comments on the Design of Robust $\mathcal{H}_2$ Controllers

The usage of the $D - K$ iteration on the previous examples demonstrates that the $D - K$ iteration is simple to start. In each of the previous designs, relatively simple initial guesses for H and N were used to start the iteration. In comparison, to design the controller by minimizing an augmented Lagrangian, one must guess both an initial controller and a set of stability multipliers. In the MIMO MACE problem, the initial controller used by How[46] to start the minimization was a robustified controller designed using a multiple model technique. If one considers the time required to obtain this type of initial condition, then the $D - K$ iteration is even more attractive.

It should also be noted that once the iteration was started, the iteration never encountered a problem starting subsequent analysis and synthesis steps. All subsequent steps ran satisfactorily. One might expect that, for instance, at the start of each synthesis step one would need to change the α_1 and α_2 parameters. Fortunately, this was never found to be necessary. However, as was discussed in the section on the MIMO MACE design, it is potentially useful to change the convergence parameters of the individual steps, as the iteration progresses. This could both reduce solution time and prevent overly suboptimal controllers from being designed.

The accuracy of our final solution is limited by the accuracy of the Riccati solver. Numerical precision decreases as problem size increases. Additionally, solution time increases with problem size. Of course, reliance on the accurate solution of Riccati equations is also common to non-robust controller design techniques. Unfortunately, these robust controllers have the disadvantage that they require the solution of a *closed loop* Riccati equation in the analysis step. This is undoubtedly the limiting factor that prevents robust controllers from being designed for arbitrarily large systems. In the discussion of the design for the MIMO MACE example, it was mentioned that certain Riccati solver routines had problems with the 118 state, closed loop Riccati equation. All computations in this thesis were performed in Matlab. Matlab software provides a variety of methods to solve Riccati equations. Riccati solvers can be found in the following Matlab software toolboxes: Control, Robust Control, μ-Synthesis, and LMI

Control. For our purposes, a series of tests were done, and it was found that the most accurate of these routines was the Schur decomposition routine in the μ-synthesis toolbox. The routine was also faster than the routines in the other toolboxes. This is the routine that was was used to solve all Riccati equations in our design examples.

The data in the three example problems also demonstrated that the existing analysis routine is significantly slower than the existing synthesis routine. Future work should concentrate on coding a routine that can solve the analysis problem more quickly. A secondary goal in the development of a new routine should be to limit the number of times that the closed loop Riccati equation must be solved. This would both reduce solution time and could potentially increase the accuracy of the solution. If a gradient routine is used, this could be achieved with a more efficient line search routine.

Because the synthesis algorithm (based on the CG or CGR iteration) has proven to be relatively fast while the analysis algorithm is relatively slow, there could potentially be a new viable alternative to a $D-K$ iteration. The alternative could be pursued for small systems or for large systems with only a few uncertainties. Rather than separating the design into analysis and synthesis steps, the routines could be integrated. The integrated routine would be initialized in the same manner as the current $D-K$ iteration—by guessing for a set of stability multipliers. It would then proceed to optimize the cost function with only the stability multipliers serving as explicit variables. However, the gradient of the cost function would not equal the constrained gradient derived for the analysis algorithm. The new gradient would implicitly allow for the controller to vary. The gradient would be approximated by calculating a set of finite differences. The stability multipliers would be perturbed slightly and a new controller synthesized at the nearby point, allowing the finite difference to be calculated. To calculate the full gradient, a synthesis would need to be performed for each stability multiplier, so this procedure would only be advisable if the problem size was small or there were not many uncertainties. The only serious drawback with this idea is that there may be a point for which a controller cannot be synthesized. As we have already seen, no method has been found to identify such a point (before synthesis is attempted). Therefore, routines would need to be written to quickly detect when a CG or CGR iteration was diverging.

We can now review the various bounds on the $\mathcal{H}_2$ cost that we have discussed and explore how they have been employed in the $D-K$ iteration. In the $\mathcal{H}_2/SN$ case, the cost function that is minimized by an LMI synthesis methodology is, by Theorem 4.1, equal to the trace quantity shown in equation (4.5). This is the same bound on the cost that is calculated in the analysis routine of our $D-K$ iteration. In Section 5.1, it was proposed that the cost function formed by summing the state feedback cost with the output estimation cost is also equal to the bound calculated in the analysis routine. It is this separation-based sum that synthesis approach implicitly minimizes. In every example problem that could be run using LMIs, this did, in fact, prove to be the case. Said another way, the cost function minimized in the analysis step is equivalent to the cost function minimized in the synthesis step.

In the $\mathcal{H}_2$/Popov case, the bound calculated by the analysis step and the cost function minimized by the LMI-based synthesis method are equal to the trace quantity in equation (4.29). Meanwhile, the cost function calculated in the separation-based synthesis technique is given in equation (5.103). This separation-based cost function is formed by summing the full information control problem cost with the output estimation cost. Unlike the $\mathcal{H}_2/SN$ case, the $\mathcal{H}_2$/Popov separation-based cost function is only expected to overbound the analysis cost function. Fortunately, in the cases examined to date, the minimization of the separation-based cost function in the synthesis step resulted in a commensurate reduction in the analysis bound. This meant that each synthesis step always resulted in a decrease in the desired cost function (except in the cases when distinctly suboptimal controllers were allowed). Significantly, even in the Popov case, the separation-based synthesis was always found to yield the same controller as was obtained via the LMI-based synthesis technique.

Overall, it seems that the proposed $D-K$ iteration meets the critical goal of being convenient to use. The time to obtain the first solution, even in the MIMO MACE case, seems reasonable, because simple initial conditions can be used to start the iteration. The initial MIMO MACE solution requires approximately five hours.

Additionally, the computational requirements of the algorithm seem reasonable. For these problems, the design algorithm seems to require solution times proportional to approximately n^4.

Chapter 7

Conclusions

This chapter begins with a summary of the work in the thesis. Then it highlights the contributions that the thesis has made to the field of robust control. The last section of the chapter discussions some of the questions that have been opened by this research.

7.1 Summary and Contributions

The goal of this thesis has been to develop new methods to design robust controllers for $\mathcal{H}_2$ performance on problems of practical dimension. Models of the MACE structural control experiment were chosen to serve as the principal plants for which controllers would be designed. It was felt that MACE was representative of many practical structural control problems. If a design routine was convenient to use for the MACE problem, then it should also be relatively convenient to use for other structural control problems.

An initial survey of the literature involving $\mathcal{H}_2$ performance analysis tools for uncertain systems led us to conclude that the use of a Popov analysis technique was advisable. Popov multipliers were chosen because they had been proven to be highly effective for systems with real parametric uncertainties. Additionally, they required less computation than more sophisticated multipliers. For systems with gain-bounded, rather than real parametric uncertainties, the multiplier was a degenerate form of the Popov multiplier. The choice of the Popov multiplier meant that controllers designed in this thesis should be equivalent to controllers previously designed by minimizing an augmented Lagrangian[46,49,50] or those designed using LMI-based optimizations for each of the steps of a $D-K$ iteration.[57]

Initially, the problem of controller synthesis was solved by examining the robustness characteristics of the closed loop system. For systems with real parametric or gain-bounded uncertainties, this led to a synthesis technique that required the minimization of a linear cost function, subject to three LMI constraints. This was a convex problem that, theoretically, could be solved in polynomial time. The cost function that was minimized was equal to a bound on the square of the closed loop $\mathcal{H}_2$ cost.

However, the LMI synthesis technique was only the first attempt to reformulate the controller synthesis problem. The LMI constraints that were derived hinted that $\mathcal{H}_2$/*SN* and $\mathcal{H}_2$/Popov controllers contained a useful separation structure. Upon examining the bound on the closed loop cost, it was found that the cost function, or a bound on it, could be separated into the sum of two parts: a cost derived from solving a robust full-state feedback or full information problem and a cost due to a robust output estimation problem. This information was then used to deduce an observer-based form for the robust controllers. A set of conditions that was sufficient to guarantee the existence of the robust controller was derived. These had the form of coupled Riccati equations. Robust controllers, then, could be derived by solving these coupled Riccati equations. For systems with gain-bounded uncertainties, it was proved that a controller derived from the LMI optimization would satisfy the sufficiency conditions of the separation principle. Although the converse of this statement was not proven, it was shown that a separation-based controller would satisfy the required robustness criteria, even if it did not minimize the cost function used in the LMI-based synthesis technique.

It was found that the LMI-based synthesis technique, while theoretically appealing, could not be used for higher-order systems (greater than approximately 18 states) on modern computers with state of the art software. Therefore, a separation-based synthesis technique was developed. A novel means of solving coupled Riccati equations was developed. These methods solved the synthesis problem by iteratively solving a set of modified Riccati equations. Two similar methods were developed, the control gain (CG) iteration and the relaxed control gain (CGR) iteration. On synthesis problems that were small enough for the LMI solver to handle, the CG and CGR iterations proved significantly faster than the LMI solver at reasonable levels of uncertainty. The primary advantage that the iterative routines had over the LMI solver was that they used Riccati equation solvers. Riccati equation routines typically solve at a rate proportional to the cube of the problem size, rather than between the fourth or the fifth power, which was the rate experienced with LMI codes.

It was also found that LMI codes were unable to solve higher-order analysis problems. Therefore, a new analysis method was developed that did not utilize LMI routines. The analysis problem became an optimization over only the stability multipliers, rather than over the multipliers and the Lyapunov matrix. This reduced the problem size to one which could be solved using conventional gradient-based optimization techniques. The gradients of the cost function were found analytically and were constrained by a Riccati robustness constraint.

The separation-based synthesis routine was combined with the new analysis routine to form a $D-K$ iteration design routine. On a small, eight-state example problem, this routine was shown to be considerably faster than a comparable $D-K$ iteration that relied upon LMI solvers for analysis and synthesis. The new design routine was then tested on both a SISO and a MIMO model of the MACE system. The routine performed well. For the MACE system, the routine met the essential goal of being convenient to use. It was easy to find initial conditions to start the routine, which meant that the time to obtain the first design solution was reasonable. Additionally, evidence suggested that, for reasonable levels of uncertainty, the methodology's solution time was proportional to approximately n^4. Thus, this method seems to offer a viable design tool for practical structural control problems.

On the small eight-state example problem and several problems of similar size, it was found that a controller derived using the separation principle matched the controller derived by solving the LMI-based minimization. It was claimed that this should hold true in general.

Since they could be set up as LMI problems, it was clear that the problems of both controller analysis and controller synthesis were convex. However, neither the analysis nor the synthesis routines used for this $D-K$ iteration were convex. The convex characteristics of the problems were not used, at least explicitly. (Implicitly, of course, in the analysis routine, gradient methods were deemed viable since it was known that no local minima existed). Convexity was sacrificed for the sake of solvability. When convex routines (LMI-based routines) failed to work on problems of practical dimension, alternative methods were developed. Certainly, this approach went against conventional thought. The current popularity of papers using LMIs demonstrates that most researchers are attempting to find ways to demonstrate that their problems are convex.

The contributions that this thesis has made to the field can now be itemized.

1. For a variety of systems, the synthesis of full-order, dynamic, LTI, $\mathcal{H}_2$ controllers can be formulated in a convex fashion, using LMIs. Previously, it had only been shown that the synthesis of static $\mathcal{H}_2$ controllers was a convex problem. In particular, this thesis shows that, given fixed stability multipliers, the problem of synthesizing *robust* $\mathcal{H}_2$ controllers for closed loop systems with two inputs and two outputs is convex. The synthesis can performed by minimizing a linear cost function subject to three LMI constraints.

2. The Lyapunov-based cost function associated with a robust $\mathcal{H}_2$ controller can be bounded by the sum of two cost functions: the cost due to a robust full information (or robust full-state feedback) problem and the cost due to a robust output estimation problem. In particular, this is true for closed loop systems with two inputs and two outputs whose uncertainty is described using Popov stability multipliers.

3. Three conditions are derived that are sufficient to guarantee the existence of a robust $\mathcal{H}_2$ controller. The conditions have the form of coupled Riccati equations. One of the conditions is derived from a robust full information (or a robust full-

state feedback) problem while the remaining two come from a robust output estimation problem.

4. New methods were developed to solve coupled Riccati equations. They are iterative schemes that successively solve modified Riccati equations. They are referred to as the CG and CGR iterative methods. These methods are general enough to be applied to coupled Riccati equations that appear in other control problems.

5. A new robust performance analysis technique was developed. The method explicitly optimizes over only the stability multipliers. By eliminating the Lyapunov matrix from the optimization, this allows modern computers to handle problems that are much larger than can be handled using an LMI framework. Gradients of the cost function were derived that implicitly account for a robustness constraint.

Furthermore, because the MACE is a complex structural system, we speculate that the results found with this system may be representative of results that would be obtained for other structural control problems. In this case, the following statements would be expected to hold:

1. At reasonable levels of robustness on systems that are small enough for an LMI solver to handle, the CG and CGR iterations solve more quickly than the comparable LMI-based synthesis problem.

2. A $D-K$ iteration, based on the separation-based synthesis technique and the new analysis technique, is a powerful design tool. On systems that are small enough to be handled by LMI-based routines, this $D-K$ iteration solves more quickly than a $D-K$ iteration that uses LMIs in both the analysis and synthesis steps. The new $D-K$ iteration is convenient to use on structural control problems of practical dimensions.

3. While solving convex problems using LMIs is theoretically attractive, this is not always the most efficient means to obtain a solution. Optimization prob-

lems with LMI constraints require a great deal of memory and computation. Since neither the synthesis nor the analysis problems were solved using convex optimization routines, it is not always the case that convex problems are best solved using convex optimization routines.

7.2 Recommendations for Future Work

Some areas of investigation should be pursued to further define the importance of this work to the field:

1. It is likely that the benefits of the suggested analysis routine have not been fully realized because of the lack of sophistication in the current analysis code. Therefore, a new and more efficient analysis code should be written. Any significant change in the efficiency of the analysis routine would lead to a significant change in total controller design time, since analysis requires the majority of the design time. If future researchers employ a gradient solver of the type that is currently used, then a more sophisticated line search routine should be developed. Other methods, such as method of centers could probably also be employed.

2. Experiments could be performed to try and quantify the benefits of using the new analysis method over an LMI-based analysis method. For instance, one could form a large set of random, uncertain plants and solve for the optimal stability multipliers using both the new analysis technique and an LMI-based analysis method. As problem dimension grows, it is expected that the benefits of the new method should become more and more apparent.

3. Criteria should be developed to detect when the CG or CGR iterations fail to synthesize a controller. Such criteria would be based on the convergence parameters of the iteration and, possibly, quantities such as the shifts in the eigenvalues of the X matrix (see equation (6.1)). The routines would have to be highly sophisticated, since the iteration typically falls into a limit cycle or

a slow spiral when it fails. Thus, examining changes between consecutive iterations would not be adequate. One would have to look at longer sequences to detect divergence. If such criteria can be developed, then, because the synthesis problem can be solved quickly, an entirely new design method could be employed. As discussed in Section 6.3.4, one could design a controller by explicitly optimizing over just the stability multipliers. At each point in the multiplier space, a controller could be optimized. This would avoid the need to perform a $D-K$ iteration.

4. For systems described by Popov multipliers, work can be done to try and determine the exact relationship between the bound on the cost found using the closed loop system and the bound on the cost that is derived using the separation principle. Relating these two bounds is complicated by the fact that the Popov stability multiplier contains a derivative term.

Appendix A

Example Problem: A System with Four Masses

This system originally appeared in Ref. 11. It has since been examined in Refs. 17, 46, 48, 57. The structure is pictured in Figure A-1. It consists of four rotating disks of nominally equal inertia linked in series by three torsion bars of nominally equal stiffness. It is a SISO system. The controller inputs a torque to disk number 2. However, the sensor is non-collocated with the control input—it measures the angular displacement of disk 4. The system is also excited by a white noise disturbance entering through disk 4. The goal of the controller is to keep disk 4 as still as possible, on average. Therefore, the controller is to minimize the $\mathcal{H}_2$ norm of a performance signal, which is composed of a weighted sum of the displacement of disk 4, the velocity of disk 4, and the control effort.

We consider two uncertainty models for this system. The first model supposes that the dimensions of disk 1 are uncertain, making its inertia uncertain. This model is interesting because slight changes in the inertia can cause poles and zeros to shift order, producing large phase changes in the model. The second uncertainty model supposes that the stiffnesses of two of the rods can be different from nominal. This model is interesting because there are two uncertainties, making the analysis problem more challenging.

We will attempt to replicate the control designs already presented by How[46] and

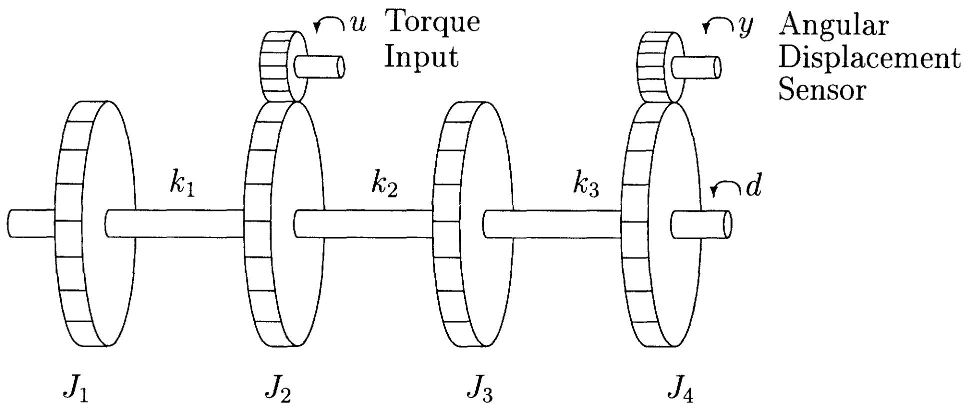

Figure A-1: The Four Mass System. Figure from Ref. 57.

Livadas.[57] The various weights for the controller are chosen to be identical to those used in the references. The nominal system can be described by the following state space model

$$A = \begin{bmatrix} 0 & I \\ -J^{-1}K & -J^{-1}D \end{bmatrix},$$

$$B_u = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{1}{m} & 0 & 0 \end{bmatrix}^T, \quad B_d = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{m} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T,$$

$$C_e = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0.1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad D_{eu} = \begin{bmatrix} 0 \\ \rho^{\frac{1}{2}} \end{bmatrix},$$

$$C_y = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad D_{yd} = \begin{bmatrix} 0 & \rho^{\frac{1}{2}} \end{bmatrix}, \tag{A.1}$$

with $\rho = 0.005$, $d = 0.01$, and

$$J = m \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, K = k \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}, D = d \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \tag{A.2}$$

where the nominal mass, m, of the disks is equal to unity, and the nominal stiffnesses of the rods, k, is also equal to unity.

Modeling an inertial uncertainty in disk 1 is complicated by the fact that the mass appears inversely in the dynamics. Therefore, following the work of Ref. 46, the inverse of the inertia of disk 1 is modelled as being uncertain. With $\tilde{J}$ representing the possible shift in the inverse of the inertia in the disk, we write that

$$\frac{1}{J_1} = \frac{1}{J_{1_{\text{nom}}}} + \tilde{J}\,, \tag{A.3}$$

where the nominal inertia, $J_{1_{\text{nom}}}$, is equal to 0.5 (it is the first entry of the matrix that defined J in equation (A.2)). This implies, for instance, that if the actual inertia varies between 1.0 and 0.25, then $\tilde{J}$ must vary between -1.0 and 2.0. The uncertainty enters the dynamics matrix as $\Delta A = -\tilde{J} B_w C_z$, with

$$B_w^T = -\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad C_z = \begin{bmatrix} -k & k & 0 & 0 & -d & d & 0 & 0 \end{bmatrix}. \tag{A.4}$$

For our design, the uncertainty limits were set to $M_2 = -M_1 = 0.159$. This lets us guarantee a level of performance for the system with inertia in the range of $0.46 < J_1 < 0.54$, *i.e.*, approximately a robustness margin of ± 8 percent.

For the second uncertainty model, we would like to examine independent variations of up to $\pm 5\%$ in k_1 and k_3, *i.e.*, in the stiffnesses of rods 1 and 3. The uncertainty in the dynamics matrix is then $\Delta A = -B_w(\Delta K)C_z$ where

$$C_z = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.5}$$

$$B_w = \begin{bmatrix} 0 & 0 & 0 & 0 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}^T, \tag{A.6}$$

$$\Delta K = \text{diag}(\Delta k_1, \Delta k_2)\,. \tag{A.7}$$

The uncertainty limits for this model were set to $M_2 = -M_1 = 0.051 I_2$.

Appendix B

Example Problem: The MACE System

The Middeck Active Control Experiment (MACE) was used to demonstrate the utility of controlled-structures technology[61] on a flexible structure in zero gravity. Experiments were initially performed on the MACE apparatus on the ground. Later experiments were performed in space, on Space Shuttle Mission STS-67 in March, 1995.[60] The experiment demonstrated that high authority structural controllers could successfully reduce the vibrations experienced by high-fidelity scientific instruments on a flexible space platform. Models based on finite-element analyses and ground-based identification experiments were used to predict the structure's zero-gravity behavior. Experiments confirmed that designing controllers to be robust to parametric uncertainties was critical for superior vibration control. Furthermore, it was shown that the re-identification of structural models, while in space, enabled controllers to be redesigned to yield superior performance. For details on this, the reader is referred to Refs. 39, 10, 40 and the references contained therein.

The MACE project provided much of the impetus for this research, because the redesign of controllers for a short-duration experiment required fast controller design techniques. For our purposes, we will be satisfied with studies of models of the MACE hardware while it was on the ground. In particular we will utilize the so-called Development Model configuration.[61] This is so that we can compare $\mathcal{H}_2$/Popov

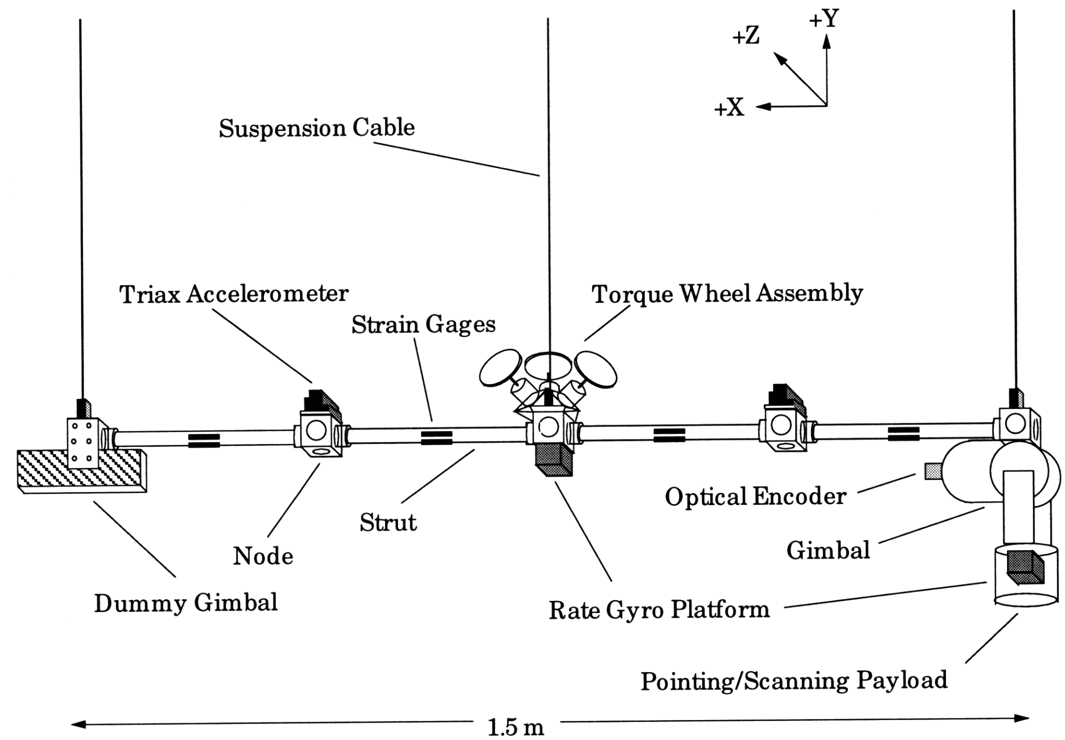

Figure B-1: The MACE system in its Development Model configuration

Property	Value
Length	1.5m
Cross-section Area	$2.25 \times 10^{-4}\text{m}^2$
Geometric Inertia (I)	$1.4 \times 10^{-8}\text{m}^4$
Mass per unit Length	0.27 kg/m
Young's Modulus	2.3×10^{9} Pa

Table B.1: Important structural properties of the MACE system controllers that we design to those designed by How, *et al.*[46,50] $\mathcal{H}_2$/Popov controllers were not tested in the space-borne experiment, because the controllers required too much time to redesign. A picture of the MACE hardware is shown in Figure B-1, and structural parameters of the overall hardware are listed in Table B.1. The figure and the data in the table were obtained from Ref. 50. The structure consists of four Lexan tubes, connected by aluminum nodes. It is suspended by wires from a sophisticated air-spring suspension system. In the center of the structure are three torque wheels, and at one end is an actuated payload—the Pointing Scanning Payload. As can be seen from the figure, the structure has a variety of sensors. The goal of the controller is to keep the Pointing Scanning Payload as still as possible, on average. For this configuration of the MACE system, broadband disturbances are injected into the system through the actuators. These actuators also serve to control the structure. Furthermore, the performance and sensed signals for the structure are the same. Collocation of these signals allows this particular configuration to have superior performance to the system which flew on the shuttle—in that system, a secondary gimbal was used to inject disturbances into the system.

Sensor noise and unmodelled high frequency dynamics must be accounted for in the controller design. Extensive tests were performed on the model on the ground to refine finite element models of the structure. The uncertainties in the model primarily reflect the expected changes that occur when the structure is removed from the effects of gravity. Removing gravity causes the damping in the structure to change and the frequencies of modes to shift.

The open loop MACE system is nominally stable because certain local control loops on the structure are closed *a priori*. These local loops are already incorporated into the models we use to design higher authority controllers. This permits us to calculate an open loop performance metric for the structure, against which we can judge the performance of our robust controllers.

The first bending mode of the MACE structure is approximately at 1.8 Hz, placing it within the bandwidth of many scientific instruments. In fact, the structure is modally rich below 100 Hz. This makes the use of a high-authority controller a

Frequency (Hz)	Damping Ratio	Mode Description
0	1	Torque wheel #1
0	1	Torque wheel #2
0	1	Torque wheel #3
0.20	0.15	Suspension — bounce
0.22	0.15	Suspension — X-axis pendulum
0.23	0.15	Suspension — Y-axis pendulum
0.23	0.15	Suspension — Z-axis pendulum
0.33	0.08	Suspension — tilt
0.20	0.15	Suspension — 1st twist
1.21	0.15	Outer gimbal pendulum
1.29	0.05	Inner gimbal pendulum
1.86	0.04	1st X-Y bending
3.13	0.04	1st X-Z bending
6.72	0.02	2nd X-Y bending
6.87	0.01	2nd X-Z bending
8.85	0.02	Suspension — 2nd twist
9.40	0.008	3rd X-Y bending
13.29	0.007	3rd X-Z bending
14.00	0.007	4th X-Y bending
14.25	0.007	Suspension — 3rd twist
17.40	0.006	4th X-Z bending
36.00	0.011	Suspension — 4th twist
39.10	0.02	Suspension — 5th twist
42.50	0.015	5th X-Y bending
64.12	0.01	5th X-Z bending

Table B.2: Modes of the MACE Structure below 65 Hz. Data from Ref. 46.

necessity if high-fidelity payloads are to be pointed accurately. The modes below 65 Hz in a finite element model of the structure are presented in Table B.2. The digital computer controlling the system consists of two Intel 80386 processors and a Weitek 3167 co-processor. This computer is able to run a 24 state SISO compensator at 1 KHz and a 59 state 3-input/3-output compensator at 500Hz.[46] Again, the reader is referred to Ref. 46 for further details.

It should be emphasized that the object of this work is not to investigate various robust control designs, but to demonstrate the efficacy of a design technique. To this end, we will design controllers for the same models and same control weightings as were investigated by How, in Ref. 46, and How *et al.* in Ref. 50. The models are based on measured transfer function data. The model size is reduced to make it appropriate for controller design. The models incorporate Pade approximations to account for computational time delays in the system.

A SISO model for the MACE system consists of 24 states. Four of these states are due to the Pade approximation. The SISO loop is the Z-axis bus rate gyro sensor and a control signal to the three torque wheels. A combination of all three wheels must be used to actuate the structure around any single axis. The chosen design conditions are the same as those that lead to the "GPC4" controller of Refs. 46, 50. We need not specify all of the parameters for the model, but we should specify a few relevant relationships:

$$\begin{aligned} C_e^T C_e &= C_y^T C_y \\ D_{eu}^T D_{eu} &= 6 \times 10^{-4} \\ C_e^T D_{eu} &= 0 \\ D_{yd}^T D_{yd} &= D_{eu}^T D_{eu}/5 \\ B_w^T D_{yd} &= 0 \\ B_w &= B_u \,. \end{aligned}$$

Four modes in this model are uncertain. These occur at 8.83, 9.40, 13.30, and 13.88 Hz. We require that the controller be robust to two percent variations in the frequencies of these modes, *i.e.*, $M_2 = -M_1 = 0.02I$.

A more complicated, MIMO model of the MACE system is also investigated. There are 3 inputs and 3 outputs, corresponding to three independent inputs to the torque wheels and three independent rate gyro measurements. The model has 59 states; this includes a three pole Pade approximation for each of the output channels. As in the SISO case, we can specify a few selected parameters in the model:

$$\begin{aligned} C_e^T C_e &= \mathrm{diag}(0.2, 1, 1) \\ D_{eu}^T D_{eu} &= 2 \times 10^{-2} I_3 \end{aligned}$$

Frequency (Hz)	$m_2 = -m_1$
1.40	0.067
3.35	0.022
4.88	0.028
5.92	0.058
8.76	0.028
8.91	0.022
9.42	0.022
13.31	0.022
13.90	0.028
14.80	0.022
33.78	0.022

Table B.3: Uncertain modes in the MIMO MACE model

$$\begin{aligned} C_e^T D_{eu} &= 0 \\ D_{yd}^T D_{yd} &= D_{eu}^T D_{eu}/10 \\ B_w^T D_{yd} &= 0 \\ B_w &= B_u \,. \end{aligned}$$

The MIMO model has eleven modes that have uncertain frequencies. These are listed in Table B.3 with their assigned levels of uncertainty. The modes at 1.4 and 5.92 Hz are considered particularly difficult to specify.

Bibliography

[1] Balakrishnan, V., "Linear Matrix Inequalities in Robustness Analysis with Multipliers," *Systems & Control Letters*, vol. 25, pp. 265–272, 1995.

[2] Balas, G., Doyle, J. C., Glover, K., and Packard, A., *μ-Analysis and Synthesis*, The Mathworks, Inc., 1991.

[3] Bender, Carl M. and Orszag, Steven A., *Advanced Mathematical Methods for Scientists and Engineers*, McGraw-Hill, New York, 1978.

[4] Bernstein, D. S. and M., Haddad W., "The Octomorphic Criterion for Multiple-Block-Structured Real Parameter Uncertainty: Real-μ Bounds Without Circles and D, N-Scales," *Proc. IEEE Conference on Decision and Control*, pp. 1984–1989, December 1993.

[5] Bernstein, Dennis S. and Haddad, Wassim M., "LQG Control with an $\mathcal{H}_\infty$ Performance Bound: A Riccati Equation Approach," *IEEE Transactions on Automatic Control*, vol. 34, no. 3, pp. 293–305, March 1989.

[6] Bernstein, Dennis S. and Hyland, David C., "The Optimal Projection Approach to Robust Fixed-Structure Control Design," *Mechanics and Control of Large Flexible Structures,* Junkins, ed., AIAA, pp. 237–293, 1990.

[7] Boussios, C. and Feron, E., "Estimating the Conservatism of Popov's Criterion," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, July 1996.

[8] Boyd, Stephen, El Ghaoui, Laurent, Feron, Eric, and Balakrishnan, Venkataramanan, *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia, 1994.

[9] Braatz, R. D., Young, P. M., Doyle, J. C., and Morari, M., "Computational Complexity of μ Calculation," *Proceedings of the American Control Conference*, pp. 1682–1683, June 1993.

[10] Campbell, Mark E., Grocott, Simon C. O., How, Jonathan P., and Woods-Vedeler, Jessica A., "Overview of Closed Loop Results for MACE," *Proc. IFAC 13th World Congress*, July 1996.

[11] Cannon, R. H. and Rosenthal, D. E., "Experiments in Control of Flexible Structures with Noncollocated Sensors and Actuators," *AIAA Journal of Guidance, Control and Dynamics*, vol. 7, no. 5, pp. 546–553, Sept.–Oct. 1984.

[12] Chen, X. and Wen, J., "A Linear Matrix Inequality Approach to the General Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Problem," *Proceedings of the American Control Conference*, pp. 1443–1447, June 1995.

[13] Chilali, M. and Gahinet, P., "$\mathcal{H}_\infty$ Design wth Pole Placement Constraints: An LMI Approach," *IEEE Transactions on Automatic Control*, vol. 41, no. 3, , March 1996.

[14] Collins Jr., Emmanuel G., Davis, Larry D., and Stephen, Richter, "Design of Reduced-Order, $\mathcal{H}_2$ Optimal Controllers Using a Homotopy Algorithm," *International Journal of Control*, vol. 61, no. 1, pp. 97–126, 1995.

[15] Collins Jr., Emmanuel G., Davis, Lawrence D., and Stephen, Richter, "Homotopy Algorithm for Maximum Entropy Design," *AIAA Journal of Guidance, Control and Dynamics*, vol. 17, no. 2, pp. 311–321, Mar.–Apr. 1994.

[16] Collins Jr., Emmanuel G., Ying, Sidney S., Haddad, Wassim M., and Richter, Stephen, "An Efficient, Numerically Robust Homotopy Algorithm for

$\mathcal{H}_2$ Model Reduction Using the Optimal Projection Equations," *Proc. IEEE Conference on Decision and Control*, pp. 1899–1904, 1994.

[17] Crawley, E. F. and Hall, S. R., "The Dynamics of Controlled Structures," Technical Report, Massachusetts Institute of Technology, Space Engineering Research Center #10-91-I, July 1991.

[18] Dahleh, Munther A. and Diaz-Bobillo, Ignacio J., *Control of Uncertain Systems: A Linear Programming Approach*, Prentice Hall, Englewood Cliffs, NJ, 1995.

[19] D'Andrea, R., "LMI Approach to Mixed Performance Objective Controllers: Application to Robust $\mathcal{H}_2$ Synthesis," Technical Report, California Institute of Technology, CDS 95-020, 1995.

[20] D'Andrea, R., "Generalized l_2 Synthesis: A New Framework for Control Design," to appear in *Proc. of the IEEE Conference on Decision and Control*, Dec. 1996.

[21] De Klerk, E. and Snyman,, "A Feasible Descent Cone Method for Linearly Constrained Minimization Problems," *Computers & Mathematics with Applications*, vol. 28, no. 6, pp. 33–44, 1994.

[22] Doyle, John, Zhou, Kemin, Glover, Keith, and Bodenheimer, Bobby, "Mixed $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Performance Objectives II: Optimal Control," *IEEE Transactions on Automatic Control*, vol. 39, no. 8, pp. 1575–1587, August 1994.

[23] Doyle, John C., Glover, Keith, Khargonekar, Pramod P., and Francis, Bruce A., "State-Space Solutions to Standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Control Problems," *IEEE Transactions on Automatic Control*, vol. 34, no. 8, pp. 831–847, August 1989.

[24] Fan, M. K. H., Tits, A. L., and Doyle, J. C., "Robustness in the Presence of Mixed Parametric Uncertainty and Unmodelled Dynamics," *IEEE Transactions on Automatic Control*, vol. AC-36, no. 1, pp. 25–38, January 1991.

[25] Feron, E., Balakrishnan, V., and Boyd, S., "Numerical Methods for $\mathcal{H}_2$ Related Problems," *Proceedings of the American Control Conference*, pp. 2921–2922, June 1992.

[26] Feron, Eric, "Analysis of Robust $\mathcal{H}_2$ Performance with Multipliers," *Proceedings of the IEEE Conference on Decision and Control*, pp. 2015–2020, 1994, to appear in *SIAM Journal of Control and Optimization*, January, 1997.

[27] Feron, Eric, Apkarian, Pierre, and Gahinet, Pascal, "Analysis and Synthesis of Robust Control Systems via Parameter-Dependent Lyapunov Functions," *IEEE Transactions on Automatic Control*, pp. 1041–1046, July 1996.

[28] Friedman, J. H., Kabamba, P. T., and Khargonekar, P. P., "Worst-Case and Average $\mathcal{H}_2$ Performance Analysis against Real Constant Parametric Uncertainty," *Automatica*, vol. 31, no. 4, pp. 649–657, 1995.

[29] Gahinet, P., Apkarian, P., and Chilali, M., "Affine Parameter-Dependent Lyapunov Functions and Real Parametric Uncertainty," *IEEE Transactions on Automatic Control*, vol. 41, pp. 436–442, March 1996.

[30] Gahinet, P. and Nemirovskii, A., "General-Purpose LMI Solvers with Benchmarks," *Proc. IEEE Conference on Decision and Control*, pp. 3162–3165, 1993.

[31] Gahinet, P. and Nemirovskii, A., "The Projective Method for Solving Linear Matrix Inequalities," *Proc. American Control Conference*, pp. 840–844, 1994.

[32] Gahinet, P., Nemirovskii, A., Laub, A. J., and Chilali, M., "The LMI Control Toolbox," *Proceedings of the IEEE Conference on Decision and Control*, pp. 2038–2041, 1994.

[33] Gahinet, Pascal and Apkarian, Pierre, "A Linear Matrix Inequality Approach to $\mathcal{H}_\infty$ Control," *International Journal of Robust and Nonlinear Control*, vol. 4, pp. 421–448, 1994.

[34] Geromel, J. C., Peres, P. L. D., and Souza, S. R., "Convex Analysis of Output Feedback Control Problems: Robust Stability and Performance," *IEEE Transactions on Automatic Control*, vol. 41, no. 7, pp. 997–1003, July 1996.

[35] Goh, K. C., Ly, J. H., Turan, L., and Safonov, M. G., "μ/K_m-Synthesis via Bilinear Matrix Inequalities," *Proc. IEEE Conference on Decision and Control*, pp. 2032–2037, Dec. 1994.

[36] Goh, Keat-Choon and Wu, Fen, "Robust $\mathcal{H}_2$ Performance Analysis using Duality and Basis Functions," to appear in *Proc. of the IEEE Conference on Decision and Control*, Dec. 1996.

[37] Grace, A., *Optimization Toolbox User's Guide*, The Mathworks, Inc., 1992.

[38] Grocott, Simon C. O., "Comparison of Control Techniques for Robust Performance on Uncertain Structural Systems," Master's thesis, Master's Thesis, Massachusetts Institute of Technology, January 1994.

[39] Grocott, Simon C. O. and Campbell, Mark E., "Control Analysis for MACE: Methods and Limitations," *Proc. IFAC 13th World Congress*, July 1996.

[40] Grocott, Simon C. O., How, Jonathan P., and Miller, David W., "A Comparison of Robust Control Techniques for Uncertain Structural Systems," *Proc. of the AIAA Guidance Navigation and Control Conference*, pp. 261–271, Aug 1994.

[41] Haddad, W. M. and Bernstein, D. S., "Parameter-Dependent Lyapunov Functions, Constant Real Parameter Uncertainty and the Popov Criterion in Robust Analysis and Synthesis Parts I and II," *Proc. IEEE Conference on Decision and Control*, pp. 2274–2279 and 2632–2633, December 1991.

[42] Haddad, W. M., How, J. P., Hall, S. R., and Bernstein, D. S., "Extensions of Mixed-μ Bounds to Monotonic and Odd Monotonic Nonlinearities using Absolute Stability Theory," *Proc. IEEE Conference on Decision and Control*, pp. 2813–2823, December 1992.

[43] Haddad, W. M., V., Kapila, and Bernstein, D. S., "Robust $\mathcal{H}_\infty$ Stabilization via Parameterized Lyapunov Bounds," *Proc. of the American Control Conference*, pp. 2374–2378, June 1995.

[44] Hall, Steven R. and How, Jonathan P., "Mixed $\mathcal{H}_2/\mu$ Performance Bounds using Dissipation Theory," *Proc. IEEE Conference on Decision and Control*, Dec. 1993.

[45] Hall, Steven R., Yang, Kyle Y., and Livadas, Carl, "On the Utility of Tensor Inequalities for Robustness Analysis," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 122–131, August 1995.

[46] How, J. P., "Robust Control Design with Real Parameter Uncertainty using Absolute Stability Theory," PhD Thesis, Massachusetts Institute of Technology, Feb. 1993.

[47] How, J. P., *Private Communication*, July 1996.

[48] How, J. P., Haddad, W. M., and Hall, S. R., "Robust Control Synthesis Examples with Real Parameter Uncertainty using the Popov Criterion," *Proc. American Control Conference*, pp. 1090–1095, June 1993.

[49] How, J. P. and Hall, S. R., "Connections Between the Popov Stability Criterion and Bounds for Real Parametric Uncertainty," *Proc. American Control Conference*, pp. 1084–1089, June 1993.

[50] How, Jonathan P., Hall, Steven R., and Haddad, Wassim M., "Robust Controllers for the Middeck Active Control Experiment Using Popov Controller Synthesis," *IEEE Transactions on Control Systems Technology*, vol. 2, no. 2, pp. 73–87, June 1994.

[51] Hyland, David C. and Bernstein, Dennis S., "The Optimal Projection Equations for Fixed-Order Dynamic Compensation," *IEEE Transactions on Automatic Control*, vol. 29, no. 11, pp. 1034–1037, November 1984.

[52] Iwasaki, T. and Skelton, R. E., "All Fixed-Order $\mathcal{H}_\infty$ Controllers: Observer-Based Structure and Covariance Bounds," *IEEE Transactions on Automatic Control*, vol. 40, no. 3, pp. 512–516, March 1995.

[53] Iwasaki, T., Skelton, R. E., and Geromel, J. C., "Linear Quadratic Suboptimal Control with Static Output Feedback," *Systems & Control Letters*, vol. 23, pp. 421–430, Dec. 1994.

[54] Jabbari, F. and Schmitendorf, W. E., "Robust Linear Controllers Using Observers," *IEEE Transactions on Automatic Control*, vol. 36, no. 12, pp. 1509–1514, Dec. 1991.

[55] Jabbari, F. and Schmitendorf, W. E., "Effects of Using Observers on Stabilization of Uncertain Linear Systems," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 266–271, Feb. 1993.

[56] Khargonekar, Pramod P. and Rotea, Mario A., "Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Control: A Convex Optimization Approach," *IEEE Transactions on Automatic Control*, vol. 36, no. 7, pp. 824–837, July 1991.

[57] Livadas, Carl, "Optimal $\mathcal{H}_2$/Popov Controller Design Using Linear Matrix Inequalities," Master's thesis, Massachusetts Institute of Technology, Feb. 1996.

[58] Maciejowski, J. M., *Multivariable Feedback Design*, Addison Wesley, New York, 1989.

[59] Mehdi, D. and Al Hamid, M., "Design of Observer-Based Robust Controllers," *Proceedings of the American Control Conference*, pp. 3373–3377, June 1995.

[60] Miller, David W., de Luis, Javier, Stover, Gregory, and Crawley, Edward F., "MACE: Anatomy of a Modern Control Experiment," *Proc. IFAC 13th World Congress*, July 1996.

[61] Miller, David W., Sepe, Raymond B., Rey, Daniel, Saarmaa, Erik, and Crawley, Edward F., "The Middeck Active Control Experiment (MACE)," *Proc. of the Fifth NASA/DOD CSI Technical Conference*, March 1992.

[62] Mustafa, Denis, "Relations Between Maximum-Entropy/$\mathcal{H}_\infty$ Control and Combined $\mathcal{H}_\infty$ /LQG Control," *Systems & Control Letters*, vol. 12, pp. 193–203, 1989.

[63] Mustafa, Denis and Glover, Keith, "Controllers which Satisfy a Closed-Loop $\mathcal{H}_\infty$ Norm Bound and Maximize an Entropy Integral," *Proceedings of the IEEE Conference on Decision and Control*, pp. 959–964, Dec. 1988.

[64] Nesterov, Y. and Nemirovski, A., *Interior Point Polynomial Methods in Convex Programming: Theory and Applications*, SIAM, Philadelphia, 1994.

[65] Packard, A. and Doyle, J. C., "The Complex Structured Singular Value," *Automatica*, vol. 29, no. 1, pp. 71–109, 1993.

[66] Packard, Andy and Doyle, John C., "Robust control with an $\mathcal{H}_2$ performance Objective," *Proceedings of the American Control Conference*, pp. 2141–2146, June 1987.

[67] Packard, Andy, Zhou, Kemin, Pandey, Pradeep, and Becker, Greg, "A Collection of Robust Control Problems Leading to LMI's," *Proc. IEEE Conference on Decision and Control*, pp. 1245–1250, Dec. 1991.

[68] Paganini, F., "Necessary and Sufficient Conditions for Robust $\mathcal{H}_2$ Performance," *Proc. of the IEEE Conference on Decision and Control*, Dec. 1995.

[69] Richter, Stephen and Collins Jr., Emmanuel, "A Homotopy Algorithm for Reduced Order Compensator Design Using the Optimal Projection Equations," *Proc. IEEE Conference on Decision and Control*, pp. 506–511, 1989.

[70] Ridgely, D. B., Valavani, L., Dahleh, M., and Stein, G., "Solution to the General Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Control Problem — Necessary Conditions for Optimality," *Proceedings of the American Control Conference*, pp. 1348–1352, June 1992.

[71] Rotea, M., "The Generalized $\mathcal{H}_2$ Control Problem," *Automatica*, vol. 29, no. 2, pp. 373–385, 1993.

[72] Rotea, M. A. and Khargonekar, Pramod P., "$\mathcal{H}_2$-Optimal Control with an $\mathcal{H}_\infty$-Constraint: The State-Feedback Case," *Automatica*, vol. 27, no. 2, pp. 307–316, 1991.

[73] Snyman, J. A. and Stander, N., "New Successive Approximation Method for Optimum Structural Design," *AIAA Journal*, vol. 32, no. 6, pp. 1310–1315, June 1994.

[74] Stoorvogel, A. A., "The Robust $\mathcal{H}_2$ Control Problem: A Worst-Case Design," *IEEE Transactions on Automatic Control*, vol. 38, no. 9, pp. 1358–1370, Sept. 1993.

[75] Sznaier, M., "An Exact Solution to General SISO Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Problems via Convex Optimization," *IEEE Transactions on Automatic Control*, vol. 39, no. 12, pp. 2511–2517, Dec. 1994.

[76] Walker, D. E. and Ridgely, D. B., "Uniqueness of the General Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Optimal Controller," *Proceedings of the American Control Conference*, pp. 1453–1457, June 1995.

[77] Whorton, M., Buschek, H., and Calise, A., "Homotopy Algorithms for Fixed Order $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Design," *Proceedings of the AIAA Guidance Navigation and Control Conference*, pp. 1083–1093, Aug. 1994.

[78] Willems, J. C., "Least Squares Stationary Optimal Control and the Algebraic Riccati Equation," *IEEE Transactions on Automatic Control*, vol. AC-16, pp. 621–242, Dec. 1971.

[79] Willems, J. C., "Dissipative Dynamical Systems Part I: General Theory," *Archive Rational Mechanics Analysis*, vol. 45, pp. 321–351, 1972.

[80] Willems, J. C., "Dissipative Dynamical Systems Part II: Linear Systems with Quadratic Supply Rates," *Archive Rational Mechanics Analysis*, vol. 45, pp. 352–393, 1972.

[81] Willems, J. C. and Trentelman, H. L., "The Dissipation Inequality and the Algebraic Riccati Equation," Ch.8 of *The Riccati Equation,* Bittanti, Laub, and Willems, eds., Springer-Verlag, pp. 197–242, 1991.

[82] Yang, Kyle Y., Livadas, Carl, and Hall, Steven R., "Using Linear Matrix Inequalities to Design Controllers for Robust $\mathcal{H}_2$ Performance," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, July 1996.

[83] Young, P. M. and Doyle, J. C., "Properties of the Mixed μ Problem and Its Bounds," *IEEE Transactions on Automatic Control*, pp. 155–159, January 1996.

[84] Young, P. M., Newlin, P. N., and Doyle, J. C., "Let's Get Real," *ASME Advances in Robust and Nonlinear Control Systems*, pp. 5–12, 1992.

[85] Zhou, K., Doyle, J. C., and Glover, K., *Robust and Optimal Control*, Prentice Hall, New Jersey, 1996.

[86] Zhou, Kemin, Glover, Keith, Bodenheimer, Bobby, and Doyle, John, "Mixed $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Performance Objectives I: Robust Performance Analysis," *IEEE Transactions on Automatic Control*, vol. 39, no. 8, pp. 1564–1574, August 1994.