# Optimal $\mathcal{H}_2$/Popov Controller Design Using Linear Matrix Inequalities

by

**Carolos Livadas**

S.B. in Aeronautics and Astronautics, MIT (1993)
S.B. in Computer Science, MIT (1993)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements
for the degree of

**Master of Science in Aeronautics and Astronautics**

at the

**Massachusetts Institute of Technology**

February 1996



Author______________________________
Department of Aeronautics and Astronautics
January 19, 1996

Certified by______________________________
Professor Steven R. Hall
Department of Aeronautics and Astronautics
Thesis Supervisor

Accepted by______________________________
Professor Harold Y. Wachman
Chairman, Departmental Graduate Committee

# Optimal $\mathcal{H}_2$/Popov Controller Design Using Linear Matrix Inequalities

by

Carolos Livadas

Submitted to the Department of Aeronautics and Astronautics
on January 19, 1996, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

## Abstract

The purpose of this thesis is to develop an efficient scheme for the design of controllers that guarantee $\mathcal{H}_2$ performance in view of linear or nonlinear real parametric uncertainties. To begin, dissipation techniques are used to design parameter-dependent Lyapunov functions that impose constraints on the magnitude and the time-variation of the uncertainties. This approach reduces the conservatism of the robustness criteria. The robust stability problem is formulated in terms of a linear matrix inequality problem that constitutes a convex constraint and can be solved in polynomial time. The robust performance problem is obtained by the introduction of a performance metric that bounds the $\mathcal{H}_2$ cost of the closed-loop system and is shown to be equivalent to the robust performance problem obtained by the $\Omega$-bound framework and the use of Popov multipliers. Although the resulting robust performance criteria are in matrix inequality form, they are not linear in the scaling matrices and the compensator dynamics. Therefore, the robust performance problem is used to optimize the performance metric with respect to the Popov scaling matrices and the controller dynamics separately. Both of these optimizations are formulated in terms of standard linear matrix inequality problems which can be solved in polynomial time. The iterative robust controller synthesis procedure obtained by interleaving the two optimizations yields a robust controller that obtains a locally optimal bound on the $\mathcal{H}_2$ cost of the closed-loop system. Since the robust performance criteria used in the iteration are equivalent to those obtained by the Popov multiplier approach, the robust controller obtained by this iterative scheme is denoted the optimal $\mathcal{H}_2$/Popov controller.

Benchmark robust control problems are used to evaluate the effectiveness of the iterative controller synthesis scheme. Several controller designs indicate that this LMI-based approach to robust controller synthesis overcomes the numerical and computational problems of other equivalent robust control techniques and therefore constitutes a viable synthesis scheme for real-life systems.

Thesis Supervisor: Steven R. Hall, Sc.D.
Title: Associate Professor of Aeronautics and Astronautics

## Acknowledgments

The completion of the research leading to this thesis involved the continuous guidance and support of certain people. First, I would like to thank my research advisor, Prof. Steven R. Hall. His knowledge and insight have been an invaluable asset to both my research and my graduate studies at MIT. Second, I would like to thank Kyle Y. Yang. Kyle's suggestions and ideas were always helpful and it is doubtful whether this thesis could have been completed without his help. Many of the results presented in the thesis are based on ideas from Kyle's doctoral research. Finally, I would like to thank my family for its continuous love and support, and all my friends at MIT for making my student life interesting and enjoyable.

This research was performed in the MIT Space Engineering Research Center. The research was funded by NASA SERC grant NAGW-1335.

# Contents

# List of Figures

# List of Tables

# Nomenclature

| | | |
|---|---|---|
| $F(\cdot)$ | = | symmetric matrix, linearly dependent on its variable space |
| $F_i$ | = | $i^{\text{th}}$ symmetric matrix |
| $U, V$ | = | full column rank matrices |
| $N_U, N_V$ | = | null spaces of full column rank matrices $U$ and $V$ |
| $\mathcal{E}_i$ | = | ellipsoid of the $i^{\text{th}}$ step of the ellipsoid algorithm |
| $\Delta(\cdot)$ | = | nonlinear uncertainty function |
| $\Delta A$ | = | uncertainty in the plant dynamics |
| $\mathcal{U}$ | = | set of allowable uncertainty functions $\Delta(\cdot)$ |
| $M_2, M_1$ | = | upper and lower sector (and slope) bounds for $\Delta(\cdot)$ |
| $M$, $\Sigma$ | = | nominal, perturbed plant |
| $K$ | = | compensator |
| $\Delta$ | = | uncertainty block |
| $\tilde{M}$ | = | closed-loop system, *i.e.*, plant and compensator |
| $V(\cdot)$ | = | Lyapunov function |
| $V_M, V_\Delta$ | = | storage functions for the plant $M$ and the uncertainty block $\Delta$ |
| $r(\cdot,\cdot)$ | = | supply rate |
| $r_M, r_\Delta$ | = | supply rates for the plant $M$ and the uncertainty block $\Delta$ |
| $W(s)$ | = | Popov multiplier (stability multiplier) |
| $H, N$ | = | Popov multiplier scaling matrices |
| $J(\mathcal{U})$ | = | square of the closed-loop $\mathcal{H}_2$ norm over all uncertainties in $\mathcal{U}$ |
| $\mathcal{J}(\mathcal{U})$ | = | bound on $J(\mathcal{U})$ |
| $x$, $u$, $y$, $d$, $e$ | = | system states, inputs, outputs, disturbances, performance variables |
| $n_x$, $n_u$, $n_y$, $n_d$, $n_e$ | = | dimensions of the respective variables |
| $A, B_d, B_w, C_e, C_z$ | = | state space matrices of the plant $M$ |

| | | |
|---|---|---|
| $A_c, B_c, C_c$ | = | state space matrices of the compensator $K$ |
| $\tilde{A}, \tilde{B}_d, \tilde{B}_w, \tilde{C}_e, \tilde{C}_z$ | = | closed-loop state space matrices |
| $R_{xx}$, $R_{uu}$, $V_{xx}$, $V_{yy}$ | = | weighting matrices for $\mathcal{H}_2$ design problem |
| $P = R^{-1}$ | = | Lyapunov matrix |
| $\tilde{P} = \tilde{R}^{-1}$ | = | closed-loop Lyapunov matrix |
| $\mathcal{T}$ | = | congruence transformation |
| $\varepsilon$ | = | $W$-$K$ iteration stopping accuracy |
| $\mathbb{R}$ , $\mathbb{R}^{r\times s}$, $\mathbb{R}^{r}$ | = | sets of real numbers, $r \times s$ real valued matrices, $\mathbb{R}^{r\times 1}$ |
| $\mathbb{C}$ , $\mathbb{C}^{r\times s}$, $\mathbb{C}^{r}$ | = | sets of complex numbers, $r \times s$ complex valued matrices, $\mathbb{C}^{r\times 1}$ |
| $\mathbb{N}^{r}$, $\mathbb{D}^{r}$, $\mathbb{S}^{r}$ | = | $r \times r$ nonnegative definite, diagonal and symmetric matrices |
| $\mathbb{E}(\cdot)$ | = | expected value |
| $(\cdot)^T$, $(\cdot)^*$, $(\cdot)^{-1}$, $(\cdot)^\dagger$ | = | transpose, complex conjugate transpose, inverse, pseudo-inverse |
| $(\cdot)^\perp$ | = | matrix whose columns form a basis for the space that is perpendicular (or normal) to the matrix within the parentheses |
| Tr $(\cdot)$ | = | trace operator |
| $\mathcal{F}(\cdot,\cdot)$ | = | linear fractional transformation |
| $\Vert\cdot\Vert_2$ | = | 2-norm |
| $\mathcal{H}_2$ | = | Hardy space of square-integrable functions on the imaginary axis, with analytic continuation into the right half-plane |
| $\mathcal{H}_\infty$ | = | Hardy space of essentially bounded functions on the imaginary axis, with analytic continuation into the right half-plane |
| $\mathcal{L}_2$ | = | set of finite energy or square-integrable signals |

## Acronyms

| | | |
|---|---|---|
| LMI | = | linear matrix inequality |
| LMIP | = | linear matrix inequality problem |
| EVP | = | eigenvalue problem |
| GEVP | = | generalized eigenvalue problem |
| RMS | = | root-mean-square |
| LTI | = | linear time-invariant |

# Chapter 1

# Introduction

The performance of many systems is obtained through the use of high authority controllers. Many controller design techniques rely on models that are assumed to accurately represent the dynamics of the system at hand. Often, however, the models do not comprise a complete description of the system dynamics. The discrepancy between the real life system and the model is usually because of *real parametric* uncertainties, either inherent in the identification of the system dynamics, or due to neglected high frequency dynamics. In fact, slight perturbations in the system dynamics may degrade the performance of a closed-loop system involving a nominal controller, or even cause instability. Such effects have introduced the challenge of robustifying the stability and performance characteristics of a closed-loop system. More specifically, the *robust stability* problem involves the design of a controller that guarantees asymptotic stability for a system whose perturbations are assumed to lie in a specific range. Similarly, the problem of *robust performance* involves the design of a controller that guarantees a certain level of closed-loop system performance, again, for a specified set of allowable perturbations. The design of techniques that yield controllers which guarantee robust stability or performance has been the focus of much recent research in control theory.

The development of schemes for the design of robust controllers involves certain trade-offs. The sacrifice of nominal performance for robustness guarantees is common, and in fact inevitable in robust control design. Furthermore, the exact description of the uncertainty in a system usually results in stability or performance criteria that are computationally intractable. Subsequently, the construction of effective design criteria involves the trade-

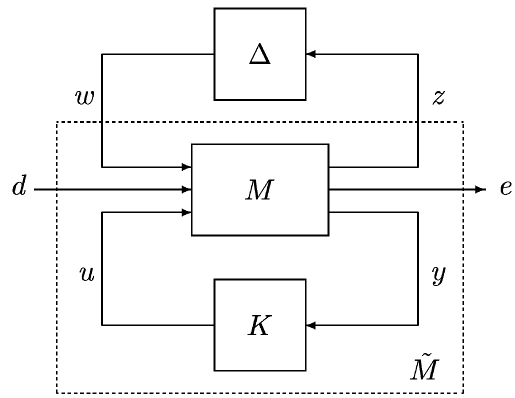



**Figure 1-1:** Robustness analysis system configuration involving a system $M$, a controller $K$ and an uncertainty block $\Delta$ which is assumed to represent real parametric and sector bounded system uncertainties.

off between the accuracy of the robustness criteria and the computational efficiency of the controller synthesis scheme. In fact, computational considerations force the use of approximations to the robustness criteria. Such approximations involve the design of robust controllers with respect to bounds on the actual performance objective or with respect to simplified uncertainty sets. However, the use of performance bounds that are too conservative, or the use of oversimplified uncertainty descriptions, result in robustness criteria that are too conservative for practical use.

A common approach when dealing with system uncertainties is to assume the uncertainties are complex. This assumption leads to much simpler robustness criteria because the phase of the uncertainty is ignored. In the case where the system uncertainty is due to the neglected high frequency dynamics whose phase is arbitrary, such an approach is actually very effective. On the other hand, in the case of real parametric uncertainties, the additional freedom in the phase of the uncertainty results in robustness criteria that are overly conservative.

The main objective of this thesis is to develop a robust controller synthesis procedure that involves less conservative robustness criteria. The standard system configuration for robust controller synthesis, as shown in Fig. 1-1, is adopted. The uncertainty is assumed to be real parametric and sector bounded. Performance robustness tests are developed as state space performance criteria and are expressed in the form of *linear matrix inequalities* (LMIs). The LMI representation allows the use of efficient optimization algorithms in the controller design. Although recent research on real parametric uncertainty has focused on systems involving linear perturbations, the approach adopted in this thesis, following the development in Refs. 24, 25 and 27, assumes the uncertainties are nonlinear and considers linear uncertainties as a special case of this broader uncertainty class.

## 1.1 Background and Previous Research

Many analysis and synthesis techniques have been developed to deal with uncertain systems. This section introduces several that are comparable to the design technique developed in this thesis. Emphasis is given to the type of uncertainties considered, the computational requirements and the qualitative evaluation of the controllers obtained from each of the approaches.

The use of a complex model to represent the uncertainty of a system is shared by various robustness analysis and synthesis techniques. $\mathcal{H}_\infty$ theory comprises a framework that guarantees robust stability for a system involving unstructured, complex uncertainties [14, 16]. The uncertainties, however, may constitute a specification of the plant uncertainty, or the $\mathcal{H}_\infty$ performance [13]. The $\mathcal{H}_2/\mathcal{H}_\infty$ problem involves the robust approach to the $\mathcal{H}_2$ problem or LQG controller design. The performance metric is the $\mathcal{H}_2$ norm of the closed-loop system and the uncertainty is assumed to be $\mathcal{H}_\infty$ norm bounded. If the $\mathcal{H}_2$ performance specification is represented as an additional uncertainty block in the $\mathcal{H}_\infty$ framework, the resulting robust stability problem involves *structured uncertainty* which is examined using $\mu$-analysis techniques [11]. Braatz *et al.* [7] have shown that the calculation of the structured singular value $\mu$ is computationally intractable. Consequently, upper and lower bounds for $\mu$ are used as an approximation [49]. In fact, these bounds may be refined by the introduction of scaling matrices, referred to as $D$-scales [11]. Unfortunately, the $D$-scales must be evaluated at each frequency point, and then be approximated by curve fitting [31]. The combination of $\mu$ analysis with $\mathcal{H}_\infty$ synthesis results in a $D$-$K$ iteration which is denoted as $\mu$-synthesis [3, 31]. Robust control design using $\mu$-synthesis has been proven to be very effective for low order systems involving complex uncertainties. However, in the case of real parametric uncertainties the $\mu$-synthesis approach and similarly all other robust control schemes involving a complex representation of the system uncertainty have been shown to be arbitrarily conservative.

The $\mu$-synthesis approach has recently been extended to provide less conservative robustness criteria when considering real parametric uncertainties [12, 15, 30, 50]. Once again, bounds on the performance are introduced because the computation of the singular value $\mu$ is intractable. Even though the calculation of the upper bound for $\mu$ has been shown to involve a convex optimization, the difficulty in using the scaling matrices to continue the synthesis iteration is still present. The multivariable stability margin $K_m$ introduced by Safonov *et al.* [10, 37–40, 44] is equivalent to $\mu$-synthesis and is based on the use of Generalized Popov Multipliers. The advantage of this approach lies in the replacement of the curve fit required for the $D$-scales by a convex optimization. The $K_m$ approach is related to the robust control techniques developed in this thesis.

The use of Lyapunov functions to examine the problems of robust stability and performance comprises an appealing robustness analysis technique because it entails state space criteria. The challenge of using this approach for systems with real parametric uncertainty involves the design of nonconservative Lyapunov functions [5, 28, 29, 33, 34]. The use of traditional quadratic Lyapunov functions employ a complex model of the system uncertainty. Haddad and Bernstein [20] have shown that such an approach ignores the phase of the uncertainty and in the case of real parametric uncertainty results in over-conservative robustness criteria. Moreover, in Ref. 22, Haddad and Bernstein developed less conservative criteria through the use of *parameter-dependent Lyapunov functions* which explicitly impose restrictions on the time variation of the system uncertainty. For a system involving sector bounded nonlinear uncertainties represented by the function $\Delta(z)$, the parameter-dependent Lyapunov functions are of the form

$$V(x, \Delta) = x^T P x + 2 \int_0^{C_z x} \Delta(\sigma)\, d\sigma \ , \tag{1.1}$$

where $x$ denotes the state of the system, and $z = C_z x$. This form is actually based on the Lur'e-Postnikov Lyapunov function developed in *absolute stability theory* [1, 32, 36].

The construction of parameter-dependent Lyapunov functions which adequately restrict the time variation of the uncertainties is simplified through the use of system dissipation theory. In fact, the nonlinearity dependent Lur'e-Postnikov Lyapunov function of the classical Popov criterion can be generalized to a parameter-dependent Lyapunov function for linear real parameter uncertainty [20, 24].

The robust control development of this thesis is based on the developments in Refs. 20 and 24. Nonconservative parameter-dependent Lyapunov functions are developed using dissipation theory and Popov robustness criteria equivalent to those of Ref. 24 are presented in the form of linear matrix inequalities.

## 1.2 Thesis Overview

The goal of this thesis is to develop an efficient technique for designing robust performance controllers for systems involving real parametric uncertainty. The stability and performance criteria are developed using Lyapunov theory. Dissipation arguments are used to obtain parameter-dependent Lyapunov functions that restrict the time variation of the uncertainties. The resulting robustness criteria are expressed in the form of linear matrix inequalities, and are subsequently incorporated in a $\mathcal{H}_2$ design problem. The robust stability and performance criteria developed are shown to be equivalent to the ones obtained through the Popov multiplier approach of Refs. 24, 25 and 27. The robust performance criteria are then used to develop an iterative scheme tackling the problem of robust controller synthesis. Since the performance involves an $\mathcal{H}_2$ performance metric and the robustness criteria are equivalent to the ones obtained through the use of Popov multipliers, the resulting controller is denoted as the *optimal $\mathcal{H}_2$/Popov controller*.

Chapter 2 provides some background that is used extensively throughout the thesis. First, the chapter introduces the aspects of dissipation theory which are used to develop the parameter-dependent Lyapunov functions. The notions of *supply rates*, *storage functions* and dissipation are presented and the implications of system dissipation to system stability and performance are explained. Subsequently, the chapter introduces the notion of linear matrix inequalities and presents some useful techniques pertaining to their analysis and solution.

Chapter 3 develops robust stability and performance analysis criteria using parameter-dependent Lyapunov functions obtained through dissipation principles. These criteria are expressed in the form of linear matrix inequalities and their equivalence to those developed in Refs. 24, 25 and 27 is portrayed. Subsequently, the robustness analysis criteria are used to develop an iterative scheme for the design of optimal $\mathcal{H}_2$/Popov controllers denoted the $W$-$K$ iteration.

The effectiveness of this iterative controller design scheme is evaluated in Chapter 4. In this chapter, several optimal $\mathcal{H}_2$/Popov controllers are designed for two standard robust control design problems. Finally, Chapter 5 presents the conclusions of the thesis and claims the practicality of the $W$-$K$ iteration for the design of optimal $\mathcal{H}_2$/Popov controllers.

# Chapter 2

# Background

## 2.1 Dissipation Theory Overview

Dissipation theory comprises a powerful method of obtaining parameter-dependent Lyapunov functions for systems involving parametric uncertainty. Consider a dynamic system $M$ with the time domain characteristics

$$\begin{aligned} \dot{x}(t) &= g(x(t), w(t)) \\ z(t) &= h(x(t), w(t)) \ . \end{aligned} \tag{2.1}$$

In the case where the system $M$ is linear time-invariant (LTI), the simplified system dynamics are

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bw(t) \\ z(t) &= Cx(t) + Dw(t) \ . \end{aligned} \tag{2.2}$$

According to Refs. 35, 47 and 48, a *supply rate* comprises a function in $w$, $z$ and possibly their derivatives, which is locally integrable, *i.e.*,

$$\int_{t_1}^{t_2} |r(w(\tau), z(\tau))| \, d\tau < \infty \ , \tag{2.3}$$

and

$$\oint r(w(\tau), z(\tau)) \, d\tau \geq 0 \ . \tag{2.4}$$

The supply rate is chosen based on the system characteristics and properties. In fact, it usually comprises a quadratic function, because such a form leads to Riccati type stability conditions.

**Definition (Willems [47, 48])** *A system $M$ of the form in Eqn. (2.1) or Eqn. (2.2), with states denoted by $x \in \mathbb{R}^n$, is said to be passive with respect to the supply rate $r(w, z)$ if there exists a positive definite function $V_M : \mathbb{R}^n \to \mathbb{R}$, called a storage function, that satisfies the dissipation inequality*

$$V_M(x(t_2)) \leq V_M(x(t_1)) + \int_{t_1}^{t_2} r(w(\tau), z(\tau))\, d\tau \;, \tag{2.5}$$

*for all $t_1$, $t_2$ and all $x$, $z$, and $w$ satisfying Eqn. (2.1) or Eqn. (2.2).*

Assuming $V_M$ is differentiable, the derivative form of Eqn. (2.5) is

$$\dot{V}_M(x(t)) \leq r(w(t), z(t)) \;, \;\; t \geq 0 \;, \tag{2.6}$$

where $\dot{V}_M$ denotes the total derivative of $V_M(x)$ along the state trajectory $x(t)$ (see Refs. 47 and 48). For a conservative system, Eqns. (2.5) and (2.6) are equalities. Moreover, a system is *strongly dissipative* if the strict form of Eqn. (2.6) is satisfied, *i.e.*, $\dot{V}_M(x(t)) < r(w(t), z(t))$, $(x \neq 0)$. Usually, storage functions are assumed differentiable, and therefore Eqn. (2.6) is applicable.

For many mechanical systems, it is helpful to think of the supply rate as a function representing the power supplied to the system and of the storage function as a function representing the energy contained within the system. Even when such a physical interpretation is often not valid, the implication of system stability from dissipation arguments in still valid.

In the case of two or more interconnected dissipative systems, the storage functions may be combined to form a Lyapunov function for the interconnected system. The stability of interconnected systems is based on the fact that the interconnection of two dissipative systems is stable.

**Lemma (Willems [47, 48])** *Consider two dynamic systems $M_1$ and $M_2$ with state space representation as in Eqn. (2.1) or Eqn. (2.2), and input output pairs $(w_1, z_1)$ and $(w_2, z_2)$ respectively. For the system interconnection illustrated in Fig. 2-1, with $w_1 = z_2$*

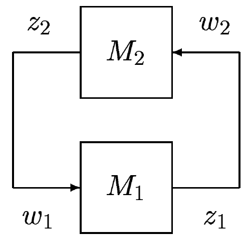



**Figure 2-1:** Neutral interconnection of two systems $M_1$ and $M_2$.

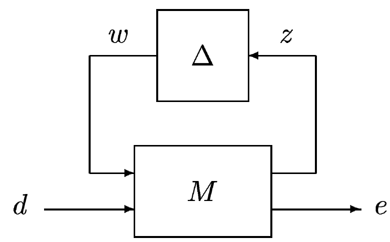



**Figure 2-2:** Robust stability and performance analysis configuration for a system with a single uncertainty block $\Delta$.

*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. Supposing 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* $w_2 = z_1$*, the solution* $(x_1, x_2) = 0$ *of the feedback interconnection of* $M_1$ *and* $M_2$ *is Lyapunov stable with Lyapunov function* $V = V_{M_1} + V_{M_2}$*.*

In the framework of robust control, the uncertain system is commonly represented by the uncertainty feedback loop of Fig. 2-2. The equivalence of this configuration with the neutral interconnection of Fig. 2-1 implies that the stability of uncertain systems can be analyzed using a dissipation framework. Indeed, if the uncertainty block $\Delta$ and the plant $M$ are dissipative with respect to the supply rates $r_\Delta$ and $r_M = -r_\Delta$, respectively, then the closed-loop system is dissipative and, in fact, stable.

In the case where multiple uncertainty blocks are considered, shown in Fig. 2-3, stability conditions are obtained in a similar fashion. Given that each uncertainty block $\Delta_i$ is dissipative with respect to some supply rate $r_{\Delta_i}$, the interconnected system is stable if the system $M$ is shown to be dissipative with respect to the supply rate

$$r_M = -\sum_i r_{\Delta_i} \ . \tag{2.7}$$

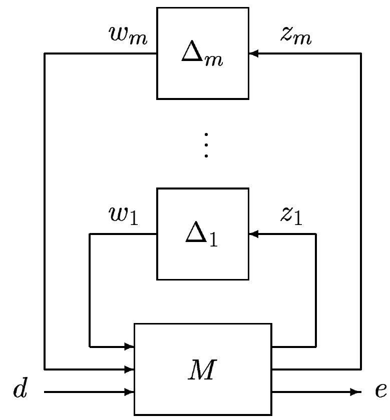



**Figure 2-3:** Robust stability and performance analysis configuration for a system with multiple uncertainty blocks $\Delta_i$.

Furthermore, robust performance may also be tackled in this framework. Suppose the performance metric is chosen to be

$$J(t) = \int_t^\infty r_P(\tau)\, d\tau = \int_t^\infty e(\tau)^T e(\tau)\, d\tau \ , \tag{2.8}$$

and the uncertainty block $\Delta$ is dissipative with respect to the supply rate $r_\Delta$. If the system $M$ is dissipative with respect to the supply rate $r_M = -r_\Delta - r_P$, the performance metric $J$ is bounded by the storage function of the interconnected system, *i.e.*, $J \leq V = V_M + V_\Delta$. In the case of multiple uncertainty blocks, the performance $J$ is similarly bounded if $M$ is dissipative with respect to the supply rate $r_M = -\sum_i r_{\Delta_i} - r_P$, where the $r_{\Delta_i}$ correspond to supply rates with respect to which each of the uncertainty blocks $\Delta_i$ are dissipative. Moreover, the introduction of $D$ scales simply involves scaling the supply rates of the uncertainty blocks $\Delta_i$ by $d_i^2$ and requiring that the block $M$ is dissipative with respect to $r_M = -\sum_i d_i^2 r_{\Delta_i} - r_P$.

## 2.2 Introduction to Linear Matrix Inequalities

A linear matrix inequality (LMI) entails a sign definiteness constraint on a matrix that depends linearly on its variable space. An LMI of size $n$ with variable space of size $m$ may be expressed in the form

$$F(x) \triangleq F_0 + \sum_{i=1}^{m} x_i F_i > 0 \ , \tag{2.9}$$

where $F_i \in \mathbb{R}^{n\times n}$ and are symmetric, *i.e.*, $F_i = F_i^T$.

Requiring the matrix $F(x)$ to be positive definite is a convex constraint on the variable space $x$. In fact, in the case where the individual matrices $F_i$ are diagonal, $F(x) > 0$ constitutes a set of $n$ linear inequalities.

The nonstrict form of the LMI in Eqn. (2.9) is expressed as $F(x) \geq 0$. Often, the solution to a nonstrict LMI is assumed to be the closure of the solution of the corresponding strict LMI. However, caution must be taken because the nonstrict LMI includes an implicit equality constraint, and moreover allows the matrix $F(x)$ to be singular. These two characteristics are absent in the corresponding strict LMI. Therefore, infeasibility of the strict LMI may incorrectly suggest that the nonstrict LMI is infeasible.

Multiple LMIs, $F_1(x) > 0$, $F_2(x) > 0$, $\ldots$ , $F_k(x) > 0$ may be combined using a block diagonal LMI whose diagonal blocks are the individual LMIs at hand, so that

$$\begin{bmatrix} F_1(x) & & & \\ & F_2(x) & & \\ & & \ddots & \\ & & & F_k(x) \end{bmatrix} > 0 \ . \tag{2.10}$$

Nonlinear matrix inequalities in Schur complement form specify convex constraints and can easily be converted to LMI format. In particular, the set of nonlinear inequalities

$$R(x) > 0\,, \ \text{and} \ Q(x) - S(x)^T R(x)^{-1} S(x) > 0 \ , \tag{2.11}$$

where $Q(x)$ and $R(x)$ are symmetric, *i.e.*, $Q(x) = Q(x)^T$, $R(x) = R(x)^T$, and $S(x)$ depends affinely on $x$ are equivalent to the LMI

$$\begin{bmatrix} Q(x) & S(x)^T \\ S(x) & R(x) \end{bmatrix} > 0 \ . \tag{2.12}$$

Often, slack variables are introduced in order to express certain problems in terms of

LMIs. Suppose, for instance, that the constraint to be enforced is of the form

$$\mathrm{Tr}\ \left[S(x)^T P(x)^{-1} S(x)\right] < 1\ ,\ \ P(x) > 0\ , \tag{2.13}$$

where $P(x)$ is symmetric, *i.e.*, $P(x) = P(x)^T$, and $S(x)$ depends affinely on $x$. Introducing a slack matrix variable $X = X^T$, the constraints of Eqn. (2.13) may be expressed as

$$\mathrm{Tr}\ X < 1\ ,\ \begin{bmatrix} X & S(x)^T \\ S(x) & P(x) \end{bmatrix} > 0\ . \tag{2.14}$$

For a detailed presentation of linear matrix inequalities, see Ref. 6.

### 2.2.1 Variable Space Elimination

A useful tool when using matrix inequalities involves the elimination of part of the variable space. Denoting the part of the variable space to be eliminated by $\Theta$ and assuming that the matrix inequality may be expressed in the form

$$F(x,\Theta) = \Psi(x) + U(x)^T \Theta V(x) + V(x)^T \Theta^T U(x) > 0\ , \tag{2.15}$$

where $\Psi(x) = \Psi(x)^T$ and all matrices are of compatible dimensions, the matrix inequality of Eqn. (2.15) is satisfiable with respect to $\Theta$ if and only if

$$\begin{aligned} N_U(x)^T \Psi(x) N_U(x) &> 0\ , \\ N_V(x)^T \Psi(x) N_V(x) &> 0\ , \end{aligned} \tag{2.16}$$

where $N_U$ and $N_V$ are matrices whose columns form bases for the null spaces of the matrices $U$ and $V$ respectively. Although the matrix inequalities in Eqn. (2.16) do not depend on $\Theta$, they comprise necessary and sufficient conditions for the matrix inequality of Eqn. (2.15) to be feasible.

The elimination of part of the variable space is often useful when the matrix inequality $F > 0$ is bilinear in the variable space. The necessary and sufficient conditions of Eqn. (2.16) are often LMIs that can be solved for the variable space not eliminated, *i.e.*, $x$. Supposing the matrix inequalities of Eqn. (2.16) are satisfied for some $x = x_0$, Eqn. (2.15) becomes an LMI constraint on $\Theta$, *i.e.*, $F(x_0, \Theta) > 0$.

### 2.2.2 Feasibility and Optimization using LMIs

Linear matrix inequalities are widely used in optimization problems. This is due to the fact that constraints may easily be expressed in an LMI formulation. Most of the problems may be identified as either LMI problems (LMIPs), eigenvalue problems (EVPs) or generalized eigenvalue problems (GEVPs).

LMIPs involve problems in which the feasibility of an LMI is investigated. In control theory, such problems usually involve stability conditions.

EVPs focus on minimizing the maximum eigenvalue of a matrix which depends affinely on a variable, subject to an LMI constraint. More explicitly, an EVP involves the minimization of a scalar $\lambda$ subject to the inequalities

$$\lambda I - A(x) > 0\,, \text{ and } B(x) > 0 \;, \tag{2.17}$$

where $A(x)$ and $B(x)$ are symmetric, *i.e.*, $A(x) = A(x)^T$, $B(x) = B(x)^T$, and depend affinely on the optimization variable $x$.

An equivalent formulation of the EVP involves the minimization of a linear function $y(x) = c^T x$ subject to an LMI constraint $F(x) > 0$ where $F(x)$ is an affine function of the variable space $x$. Moreover, in the case where the individual LMI matrices $F_i$ are diagonal, the EVP reduces to the standard linear programming problem of optimizing a linear function subject to a set of linear inequality constraints on the variable space $x$.

GEVPs are actually quasiconvex problems. They involve minimizing the generalized eigenvalue of a pair of matrices, which depend affinely on the variable space, subject to an LMI constraint. More explicitly, GEVPs involve the minimization of a scalar $\lambda$ subject to

$$\lambda B(x) - A(x) > 0\,,\; B(x) > 0\,, \text{ and } C(x) > 0 \;, \tag{2.18}$$

where $A(x)$, $B(x)$ and $C(x)$ are symmetric, *i.e.*, $A(x) = A(x)^T$, $B(x) = B(x)^T$ and $C(x) = C(x)^T$, and depend affinely on the optimization variable $x$.

An equivalent formulation of the GEVP involves the minimization of a scalar variable $\lambda$ subject to an LMI constraint $A(x, \lambda) > 0$ where $A(x, \lambda)$ is affine in $x$ for fixed $\lambda$ and affine in $\lambda$ for fixed $x$ and $A(x, \lambda)$ is monotonic in $\lambda$, *i.e.*, $\lambda > \mu \Rightarrow A(x, \lambda) > A(x, \mu)$.

These standard problems involving LMIs may easily be solved using the LMILAB Matlab

Toolbox [6, 17, 18].

### 2.2.3 Solution Methods

The advantage in formulating optimization problems in terms of LMIs lies in the fact that such problems may be solved in polynomial time. Although multiple ellipsoid algorithms and interior point methods may be used to solve the standard LMI problems of Section 2.2.2, only the simplest form of the ellipsoid algorithm will be outlined here. The goal of such a presentation is to provide the reader with a general overview of the way in which such algorithms are structured.

The simplest ellipsoid algorithm assumes that the problem at hand has at least one optimal point, *i.e.*, the constraints are feasible for at least one point in the variable space. In the case of a LMIP, any feasible point is considered optimal. The approach to finding an optimal point is based on determining, at each iteration step, a half-ellipsoid which contains an optimal point. Suppose the initial variable space is contained in an ellipsoid $\mathcal{E}_0$ and is guaranteed to contain an optimal point. A cutting plane through the center, $x_0$, of the ellipsoid $\mathcal{E}_0$, can be computed which guarantees that an optimal point lies in a half-space. Since the ellipsoid is guaranteed to contain an optimal point, an optimal point must lie in the half-ellipsoid, *i.e.*, the intersection of the ellipsoid $\mathcal{E}_0$ and the half-space containing the optimal point. The next step involves determining the ellipsoid $\mathcal{E}_1$ of minimum volume which contains the half-ellipsoid of $\mathcal{E}_0$ that contains an optimal point. $\mathcal{E}_1$ is then guaranteed to contain an optimal point. This process is repeated until the center point of some ellipsoid $\mathcal{E}_k$ satisfies the LMI constraints and exceeds the global minimum by less than some prespecified accuracy, *i.e.*, is optimal. The sequence of ellipsoids generated are guaranteed to contain an optimal point and moreover, their volume decreases geometrically implying polynomial time convergence (see Ref. 6).

In actuality, LMI problems are solved using more sophisticated algorithms that converge faster. The polynomial time convergence of such algorithms make the LMI problem formulation a computationally attractive optimization tool that is widely used in recent control theory applications.

# Chapter 3

# Robust Control Design

In engineering applications, systems are analyzed using models that are approximations of the actual system dynamics. The discrepancy between a real-life system and its model is usually due to uncertainty in the identification of the system's parameters, or unmodeled high frequency dynamics. These uncertainties may degrade performance or cause system instability. Robust control design strives to guarantee stability or performance for an uncertain system.

Even though system uncertainties correspond to perturbations in real quantities, such as the stiffness in structural analysis problems, most robust techniques, such as complex $\mu$-synthesis, allow the uncertainty to be complex. This assumption introduces conservatism into the stability or the performance criteria. In an effort to reduce this conservatism, recent research, such as real $\mu$-synthesis, has concentrated on constraining the system uncertainty to be real.

In the case of robust performance, the goal is to guarantee a certain level of closed-loop performance, despite the uncertainty in the system. The performance objective adopted in this thesis involves the a root-mean-square (RMS) metric on the output. In the case of linear systems, this is simply the worst-case $\mathcal{H}_2$ norm of the uncertain system.

Section 3.1 introduces the system configuration, the class of uncertainties to be considered, and the performance objective. Sections 3.2 and 3.3 present a dissipation approach to Popov robust stability and performance analysis. Finally, Section 3.4 develops an iterative scheme for robust performance synthesis which yields the optimal $\mathcal{H}_2$/Popov controller.

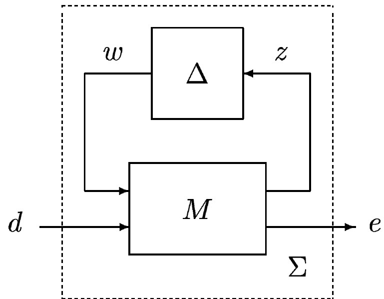



**Figure 3-1:** Representation of uncertain system using an uncertainty feedback loop. The nominal system and the uncertainty block are denoted by $M$ and $\Delta$ respectively.

## 3.1 Introduction

Robust control involves the stability and performance analysis of uncertain systems. In the case of linear systems, the uncertainty of a system may be represented in the form of perturbations of its state space characteristics. An uncertain system may therefore be represented as

$$\begin{aligned} \dot{x} &= (A + \Delta A)x + (B_d + \Delta B_d)d \\ e &= (C_e + \Delta C_e)x \ . \end{aligned} \tag{3.1}$$

Using a technique presented in Ref. 43, the uncertainty may be isolated solely to the $A$ matrix of a system equivalent to the one in Eqn. (3.1). The state space characteristics of the resulting system are expressed as

$$\begin{aligned} \dot{x} &= (A + \Delta A)x + B_d d \\ e &= C_e x \ . \end{aligned} \tag{3.2}$$

The uncertainty, characterized by a perturbation $\Delta A$ from the nominal $A$ matrix, may also be represented using an uncertainty feedback loop, as shown in Fig. 3-1. The state space characteristics for this configuration are given by

$$\begin{aligned} \dot{x} &= Ax + B_w w + B_d d \\ z &= C_z x \\ e &= C_e x \ , \end{aligned} \tag{3.3}$$

where the uncertainty in the $A$ matrix, for a linear $\Delta$ block, is given by $\Delta A = B_w \Delta C_z$.

In the formulation that follows, the uncertainty of the system is assumed to be real

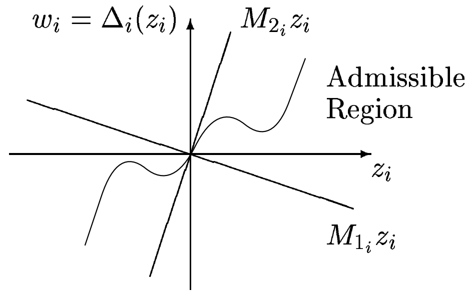



**Figure 3-2:** Sector bound for each uncertainty input-output map. The output $w_i$ of the uncertainty block $\Delta$ is required to lie within the region bounded by the lines $M_{2_i} z_i$ and $M_{1_i} z_i$ from above and below respectively.

parametric, *i.e.*, real perturbations of the system parameters. In other robust control techniques, such as complex $\mu$-synthesis, the uncertainty is allowed to be complex. However, this additional freedom in the uncertainty introduces conservatism in the stability and performance criteria. In fact, in many mechanical systems, allowing the uncertainty to be complex results in nonphysical characteristics, such as negative damping.

In this thesis, the uncertainty is assumed to be sector bounded, as shown in Fig. 3-2, and, moreover, is allowed to be nonlinear. The sector bound constraint for each of the uncertainty input-output maps may be expressed by the inequality

$$(w_i - M_{1_i} z_i)^T (M_{2_i} - M_{1_i})^{-1} (M_{2_i} z_i - w_i) \geq 0\,, \text{ and } w_i(0) = 0 \;, \tag{3.4}$$

where $M_{1_i}$ and $M_{2_i}$ are the slopes of the sector boundaries and $M_{2_i} > M_{1_i}$. For multiple uncertainties, *i.e.*, an uncertainty block $\Delta$ of size $n_\Delta$, the individual inequalities of the form in Eqn. (3.4) for each uncertainty input-output map may be combined in the vector expression

$$(\Delta(z) - M_1 z)^T M_d (M_2 z - \Delta(z)) \geq 0\,, \text{ and } \Delta(0) = 0 \;, \tag{3.5}$$

where $M_1 = \text{diag}(M_{1_i}, i = 1, \ldots, n_\Delta)$, $M_2 = \text{diag}(M_{2_i}, i = 1, \ldots, n_\Delta)$, $M_2 > M_1$, and

$$M_d = (M_2 - M_1)^{-1} \;. \tag{3.6}$$

The set of such sector bounded uncertainty blocks is represented by

$$\mathcal{U} \triangleq \left\{\Delta(z) \,\middle|\, (\Delta(z) - M_1 z)^T M_d (M_2 z - \Delta(z)) \geq 0 \,, \text{ and } \Delta(0) = 0 \right\} \,, \tag{3.7}$$

and in the case when the $\Delta$ block is assumed to be linear,

$$\mathcal{U} \triangleq \{\Delta \,|\, \Delta A = B_w F C_z \,, \text{ where } M_1 \leq F \leq M_2\} \,. \tag{3.8}$$

Another point of differentiation among robust control techniques lies in the choice of the performance metric. In this thesis an $\mathcal{H}_2$ or root-mean-square (RMS) performance metric is adopted. In the case of a linear time-invariant (LTI) systems, this performance metric is characterized by the worst-case $\mathcal{H}_2$ norm. In Ref. 42, the $\mathcal{H}_2$ norm of a system is interpreted in terms of impulsive-input responses. Let $\{d_1, \ldots, d_{n_d}\}$ denote a basis for the input space and let $e_i$ represent the response of the system to an impulsive input of the form $d_i\delta$. The $\mathcal{H}_2$ norm of the system is the sum of the squares of the $\mathcal{L}_2$ norms of the impulsive-input responses $e_i$. In fact, because the system is uncertain, the highest $\mathcal{H}_2$ norm over all allowable system perturbations, $\Delta A$, must be chosen. Denoting the closed loop system by $\Sigma$, the worst-case $\mathcal{H}_2$ norm is defined as

$$\|\Sigma\|_2^2 \triangleq \sup_{\Delta \in \mathcal{U}} \sum_{i=1}^{n_d} \|e_i(t)\|_2^2 \,, \tag{3.9}$$

$$\triangleq \sup_{\Delta \in \mathcal{U}} \sum_{i=1}^{n_d} \int_0^\infty e_i(t)^T e_i(t) \; dt \,. \tag{3.10}$$

An equivalent definition of the $\mathcal{H}_2$ norm for an LTI system involves the system response to zero-mean and unit covariance white noise input. Once again, since the system is uncertain, the worst-case $\mathcal{H}_2$ norm must be considered, and therefore

$$\|\Sigma\|_2^2 \triangleq \sup_{\Delta \in \mathcal{U}} \lim_{t \to \infty} \mathbb{E}\left[\|e(t)\|_2^2\right] \,, \tag{3.11}$$

$$\triangleq \sup_{\Delta \in \mathcal{U}} \lim_{t \to \infty} \mathbb{E}\left[\frac{1}{t}\int_0^t e(\tau)^T e(\tau) \; d\tau\right] \,. \tag{3.12}$$

Furthermore, for a linear $\Delta$ block, assuming the uncertain system is stable and supposing there are no feedthrough terms, the $\mathcal{H}_2$ norm is usually calculated using the observability or controllability Gramians. More precisely,

$$\|\Sigma\|_2^2 \triangleq \sup_{\Delta \in \mathcal{U}} \operatorname{Tr} \left[Q_{\Delta A} C_e^T C_e\right] = \sup_{\Delta \in \mathcal{U}} \operatorname{Tr} \left[P_{\Delta A} B_d B_d^T\right] \ , \tag{3.13}$$

where $Q_{\Delta A} > 0$ and $P_{\Delta A} > 0$ are the controllability and observability Gramians respectively, *i.e.*, the nonnegative definite solutions to

$$0 = (A + \Delta A) Q_{\Delta A} + Q_{\Delta A} (A + \Delta A)^T + B_d B_d^T \ , \tag{3.14}$$

$$0 = (A + \Delta A)^T P_{\Delta A} + P_{\Delta A} (A + \Delta A) + C_e^T C_e \ . \tag{3.15}$$

In the case of nonlinear and time varying systems, the notion of an $\mathcal{H}_2$ norm is not valid. However, in Ref. 42 it is shown that the expressions of Eqns. (3.10) and (3.12) may be adopted as the performance objective of nonlinear systems. Some caution must be taken because, in the nonlinear case, the two definitions are not equivalent. In fact, Eqn. (3.10) involves the transient response of the system, where as Eqn. (3.12) represents the steady state response.

Unfortunately, the worst case $\mathcal{H}_2$ norm of a system can only be calculated by an exhaustive search over all allowable perturbations of the uncertain system. Obviously, a robust control technique involving this brute-force calculation is computationally intractable. Subsequently, robust control techniques have been designed to minimize bounds on the worst-case $\mathcal{H}_2$ norms of the uncertain systems considered. The approach presented in this thesis, which is based on the system dissipation framework introduced in Refs. 47 and 48, adopts the straight-forward extension of the $\mathcal{H}_2$ cost of Eqn. (3.10) to nonlinear systems. The performance metric is therefore defined as

$$J(\mathcal{U}) \triangleq \sup_{\Delta(z) \in \mathcal{U}} \sum_{i=1}^{n_d} \|e_i(t)\|_2^2 \ . \tag{3.16}$$

In robustness analysis, a sufficient condition for robust stability of the system in Fig. 3-1 may be obtained by showing that the neutral interconnection of the blocks $M$ and $\Delta$ is dissipative. Assuming the uncertainty block $\Delta$ is dissipative with respect to a supply rate $r_\Delta$, the closed loop system is dissipative, *i.e.*, stable, if the system $M$ is dissipative with respect to the supply rate $r_M = -r_\Delta$.

Hall and How [23] tackle robust performance by introducing a performance supply rate of the form $r_P = e^T e$. If the interconnection of $\Delta$ and $M$ is dissipative with respect to the supply rates $r_\Delta$and $r_M = -r_\Delta - r_P$, the closed loop performance

$$J(t) = \int_t^\infty e^T e \; d\tau = \int_t^\infty r_P \; d\tau \;\; , \tag{3.17}$$

is bounded by the closed loop storage function, $V(x(t))$, *i.e.*,

$$J(t) \leq V(x(t)) = V_M(t) + V_\Delta(t) \;\; . \tag{3.18}$$

Using Eqn. (3.17) it can easily be seen that the worst-case closed loop $\mathcal{H}_2$ norm can be written as

$$J(\mathcal{U}) = \sup_{\Delta(z)\, \in\, \mathcal{U}} \sum_{i=1}^{n_d} J_i(t) = \sup_{\Delta(z)\, \in\, \mathcal{U}} \sum_{i=1}^{n_d} \int_0^\infty e_i^T e_i \; dt \;\; , \tag{3.19}$$

where the $J_i(t)$ are assumed to be the cost-to-go accumulated by the system responses to impulsive inputs of the form $d = d_i \delta(\tau - t)$ spanning the input space. However, using the inequality of Eqn. (3.18), and since the Lyapunov functions hold for all allowable uncertainty blocks $\Delta$, the worst-case closed loop cost-to-go is bounded by

$$J(\mathcal{U}) = \sup_{\Delta(z)\, \in\, \mathcal{U}} \sum_{i=1}^{n_d} J_i(t) \leq \sum_{i=1}^{n_d} V_i(x(t)) \;\; , \tag{3.20}$$

where $V_i(x(t))$ corresponds to the storage function for an input of the form $d = d_i \delta(\tau - t)$. Moreover, using the state space dynamics of Eqn. (3.3) the bound becomes

$$J(\mathcal{U}) \leq \sum_{i=1}^{n_d} V_i(B_d d_i) \triangleq \mathcal{J}(\mathcal{U}) \;\; . \tag{3.21}$$

## 3.2 Popov Robust Stability Analysis

An uncertain system is considered to be robustly stable whenever stability is assured for all allowable system perturbations. This section develops sufficient conditions for robust stability for the uncertain system configuration shown in Fig. 3-1. The state space characteristics of Eqn. (3.3) are adopted and the uncertainties are assumed to be nonlinear and sector bounded as shown in Fig. 3-2. The importance of this section lies in the fact that the robust stability conditions are expressed in terms or LMIs. Thus, the question of robust stability is reduced to a convex feasibility problem that can be solved in polynomial time.

The development of many robust stability criteria ignore the phase information of the uncertainties. In general, most real-life systems involve uncertainties that are either constant or slowly time varying. Therefore, the freedom introduced by neglecting the phase of the uncertainty is artificial and adds conservatism to the stability criteria.

In the dissipation framework, the time variation of the uncertainties may be restricted by choosing the supply rates and the storage functions to describe the time domain characteristics of the uncertainty block $\Delta$.

Recall that the sector bound constraints for the individual input-output maps of the uncertainty block $\Delta$ are described by the inequalities in Eqn. (3.4). Weighing each of these inequalities by a scale $H_i > 0$ results in a weighted form of the sector bound constraint of Eqn. (3.5), given by

$$(w - M_1 z)^T (M_2 - M_1)^{-1} H (M_2 z - w) \geq 0 \ , \tag{3.22}$$

where $H = \text{diag}(H_i, i = 1, \dots, n_\Delta) > 0$.

Moreover, the sector bound constraints of the uncertainty input-output maps shown in Fig. 3-2 constrain the integral of the difference of the curve $w_i$ and the line $M_{1_i} z_i$ to be positive. Again, a diagonal scaling matrix $N = \text{diag}(N_i, i = 1, \dots, n_\Delta) \geq 0$ may be introduced to weight the contribution of each of the input-output map integrals. In vector notation, the resulting constraint is

$$\int_0^{z(t)} (w - M_1 \zeta)^T N \, d\zeta \geq 0 \ , \tag{3.23}$$

or

$$\int_0^{t} (w - M_1 z)^T N \dot{z} \, d\tau \geq 0 \ . \tag{3.24}$$

Eqns. (3.22), (3.23) and (3.24) involve inequalities that describe the uncertainties considered in Fig. 3-2. Their inclusion in the robust stability criteria is thus crucial. Recall, however, that in order to ensure stability in the dissipation framework, the uncertainty block $\Delta$ must be dissipative with respect to its supply rate. Defining the supply rate and the storage function of the uncertainty block $\Delta$ in terms of the positive definite quantities in Eqns. (3.22) and (3.24) as

$$r_{\Delta} = 2\left[(w - M_1 z)^T N \dot{z} + (w - M_1 z)^T (M_2 - M_1)^{-1} H (M_2 z - w)\right] \ , \tag{3.25}$$

$$V_{\Delta} = 2 \int_0^t (w - M_1 z)^T N \dot{z} \, d\tau \ , \tag{3.26}$$

the dissipation condition for the uncertainty block $\Delta$, *i.e.*, $\dot{V}_{\Delta} \leq r_{\Delta}$, is trivially satisfied.

Since the uncertainty block $\Delta$ has been defined to be dissipative with respect to the supply rate $r_{\Delta}$ presented in Eqn. (3.25), a sufficient closed loop stability criterion is the dissipativity of the block $M$ with respect to the supply rate $r_M = -r_{\Delta}$. Choosing the storage function for the block $M$ as $V_M = x^T P x$, where $P > 0$, the closed loop stability criterion is $\dot{V}_M \leq r_M = -r_{\Delta}$, or, more explicitly,

$$\begin{aligned} \dot{V}_M + r_{\Delta} = {} & \dot{x}^T P x + x^T P \dot{x} \\ & + 2\left[(w - M_1 z)^T N \dot{z} + (w - M_1 z)^T (M_2 - M_1)^{-1} H (M_2 z - w)\right] \leq 0 \ . \end{aligned} \tag{3.27}$$

Expanding all the terms in Eqn. (3.27) using the system's state space characteristics from Eqn. (3.3), and recalling that $M_d = (M_2 - M_1)^{-1} > 0$, Eqn. (3.27) may be written as the linear matrix inequality in $P$, $H$ and $N$

$$\left[\begin{array}{c|c} \begin{matrix} A^T\left(P - C_z^T N M_1 C_z\right) + \left(P - C_z^T M_1 N C_z\right) A \\ -C_z^T M_1 M_d H M_2 C_z - C_z^T M_2 H M_d M_1 C_z \end{matrix} & (\cdots)^T \\ \hline \begin{matrix} B_w^T\left(P - C_z N M_1 C_z\right) + N C_z A \\ + M_d H M_2 C_z + H M_d M_1 C_z \end{matrix} & \begin{matrix} -H M_d - M_d H \\ + B_w^T C_z^T N + N C_z B_w \end{matrix} \end{array}\right] \leq 0 \ . \tag{3.28}$$

The existence of $P$, $H$ and $N$ that satisfy the LMI in Eqn. (3.28) implies the robust stability of the closed loop system $\Sigma$, and therefore the LMI in Eqn. (3.28) comprises a sufficient condition for robust stability. As mentioned above, expressing the robust stability criterion as an LMI is important, because the feasibility of an LMI constitutes a convex problem that can be solved in polynomial time.

Refs. 24 and 27 address this robust stability problem using stability multipliers. The stability multipliers are used in an effort to include uncertainty phase information into the stability criteria. Adopting the stability multipliers, or Popov multipliers,

$$W = H + Ns \ , \tag{3.29}$$

proposed by Refs. 8,36 and 51, the uncertain system is guaranteed to be stable with respect to the sector bounded uncertainties of Fig. 3-2, if there exist $P$, $H$, and $N$ that satisfy the LMI

$$\left[\begin{array}{c|c} \left(A^T + C_z^T M_1 B_w^T\right) P + P\left(A + B_w M_1 C_z\right) & (\cdots)^T \\ \hline HC_z + B_w^T P + NC_z\left(A + B_w M_1 C_z\right) & \begin{array}{c} -HM_d - M_d H \\ +B_w^T C_z^T N + NC_z B_w \end{array} \end{array}\right] \leq 0 \ . \tag{3.30}$$

The two seemingly independent robust stability criteria of Eqns. (3.28) and (3.30) are in fact equivalent. Let $\mathcal{T}$ denote the congruence transformation

$$\mathcal{T} = \left[\begin{array}{cc} I & 0 \\ M_1 C_z & I \end{array}\right] \ . \tag{3.31}$$

Multiplying Eqn. (3.28) on the right and left by the congruence transformation $\mathcal{T}$ and its transpose $\mathcal{T}^T$, respectively, Eqn. (3.30) is recovered.

The equivalence between the two results indicates that the two arbitrary scalings $H$ and $N$ introduced in Eqns. (3.22), (3.23) and (3.24) are actually the scaling matrices for the stability multipliers in Eqn. (3.29). Because of this equivalence, the scaling matrices $H$ and $N$ may be referred to in the following sections as the Popov multiplier weighting matrices.

## 3.3 $\mathcal{H}_2$/Popov Robustness Analysis

The previous section addressed the problem of guaranteeing stability for a system subject to a set of allowable perturbations. Often, however, stability alone does not fulfill the performance requirements of a system. This section addresses the problem of determining the optimal performance that can be attained by an uncertain system subject to the allowable system perturbations. The system considered is represented by the state space characteristics in Eqn. (3.3) and the uncertainty input-output maps are assumed to be sector bounded as shown in Fig. 3-2 and moreover allowed to be nonlinear.

From dissipation theory, the cost function defined as $J = \int_t^\infty e^T e \, d\tau$ is overbounded by the closed loop storage function, provided that the closed loop system is dissipative with the supply rate $r = -r_P = -e^T e$. Therefore, since the closed loop storage function is defined to be $V = V_M + V_\Delta$, the robust performance condition is the dissipation condition for the closed loop, or

$$\dot{V} = \dot{V}_M + \dot{V}_\Delta \leq r_M + r_\Delta = -r_P \ . \tag{3.32}$$

The supply rates and the storage functions for the blocks $\Delta$ and $M$ are chosen as in the robust stability analysis of the previous section, *i.e.*,

$$\begin{aligned} V_\Delta &= 2\int_0^t (w - M_1 z)^T N \dot{z} \, d\tau & (3.33)\\ r_\Delta &= 2\left[(w - M_1 z)^T N \dot{z} + (w - M_1 z)^T (M_2 - M_1)^{-1} H (M_2 z - w)\right] & (3.34)\\ V_M &= x^T P x \ , \ \text{ where } P > 0 & (3.35)\\ r_M &= -r_\Delta - r_P \ , & (3.36) \end{aligned}$$

where $H > 0$, $N \geq 0$, $M_2 > M_1$ and $H$, $N$, $M_1$ and $M_2$ are diagonal.

Recalling that the $\Delta$ block is dissipative with respect to the supply rate $r_\Delta$ of Eqn. (3.34), the robust performance criterion of Eqn. (3.32) is implied whenever the dissipation condition $\dot{V}_M \leq -r_\Delta - r_P$ is satisfied. More explicitly,

$$\begin{aligned} \dot{V}_M + r_\Delta + r_P = \dot{x}^T P x + x^T P \dot{x} + x^T C_e^T C_e x \\ + 2\left[(w - M_1 z)^T \dot{z} + (w - M_1 z)^T (M_2 - M_1)^{-1} H (M_2 z - w)\right] \leq 0 \ . \end{aligned} \tag{3.37}$$

Using the state space system dynamics in Eqn. (3.3), the sufficient robust performance criterion of Eqn. (3.37) may be transformed into the LMI constraint in $P$, $N$ and $H$ of the form

$$\left[\begin{array}{c|c} \begin{array}{c} A^T\left(P - C_z^T N M_1 C_z\right) + \left(P - C_z^T M_1 N C_z\right) A \\ -C_z^T M_1 M_d H M_2 C_z - C_z^T M_2 H M_d M_1 C_z + C_e^T C_e \end{array} & (\cdots)^T \\ \hline \begin{array}{c} B_w^T\left(P - C_z N M_1 C_z\right) + N C_z A \\ +M_d H M_2 C_z + H M_d M_1 C_z \end{array} & \begin{array}{c} -H M_d - M_d H \\ +B_w^T C_z^T N + N C_z B_w \end{array} \end{array}\right] \leq 0 \ . \tag{3.38}$$

The LMI of Eqn. (3.38) comprises a sufficient condition for robust performance.

The goal of this section, however, is to determine the optimal performance that can be guaranteed for the closed loop system in view of the allowable system perturbations. More precisely, it is desired to minimize the $\mathcal{H}_2$ norm of the closed loop system subject to the robust performance constraint of Eqn. (3.38). Recall that the cost $J(t)$ of the system in bounded by the closed loop storage function $V(t)$, so that

$$\begin{aligned} J(t) \leq V(t) = V_M(t) + V_\Delta(t) &= x^T P x + 2 \int_0^z (w - M_1 \zeta)^T N \, d\zeta \\ &= x^T P x + 2 \int_0^x (w - M_1 C_z \xi)^T N C_z \, d\xi \ . \end{aligned} \tag{3.39}$$

From the uncertainty sector bound constraint of Fig. 3-2, it can easily be seen that

$$w_i - M_{1_i} z_i \leq (M_{2_i} - M_{1_i}) z_i \ . \tag{3.40}$$

Therefore, the bound on the closed loop cost from Eqn. (3.39) may be further bounded by

$$\begin{aligned} J(t) \leq V(t) = V_M(t) + V_\Delta(t) &= x^T P x + 2 \int_0^x (w - M_1 C_z \xi)^T N C_z \, d\xi \\ &\leq x^T P x + 2 \int_0^x \left[ (M_2 - M_1) \, C_z \xi \right]^T N C_z \, d\xi \ . \end{aligned} \tag{3.41}$$

This bound on the $\mathcal{H}_2$ norm of the system may be calculated using the impulse response argument introduced in the evaluation of the actual $\mathcal{H}_2$ norm in Eqn. (3.21). Letting $\{d_1, \ldots, d_{n_d}\}$ be a basis for the space spanned by the input $d$, the bound on the cost-to-go of the uncertain system is given by the the sum of the responses to impulses of the form $d_i \ \delta(\tau - t)$, *i.e.*,

$$\begin{aligned} J(\mathcal{U}) \leq \mathcal{J}(\mathcal{U}) &= \sum_i^{n_d} d_i^T \left[ B_d^T \left( P + C_z^T \left( M_2 - M_1 \right) N C_z \right) B_d \right] d_i \\ &= \text{Tr} \ \left[ B_d^T \left( P + C_z^T \left( M_2 - M_1 \right) N C_z \right) B_d \right] \ . \end{aligned} \tag{3.42}$$

This is precisely the bound obtained in Refs. 20, 24 and 25 using the $\Omega$-bound framework. As in the robust stability case, the robust performance condition obtained in Refs. 24 and 25 may be recovered by applying the congruence transformation of Eqn. (3.31) to Eqn. (3.38).

In summary, this section has reduced the robust performance problem to an EVP. The cost function to be minimized is the bound on the cost-to-go of the system as defined in Eqn. (3.42), and the LMI constraint is presented in Eqn. (3.38).

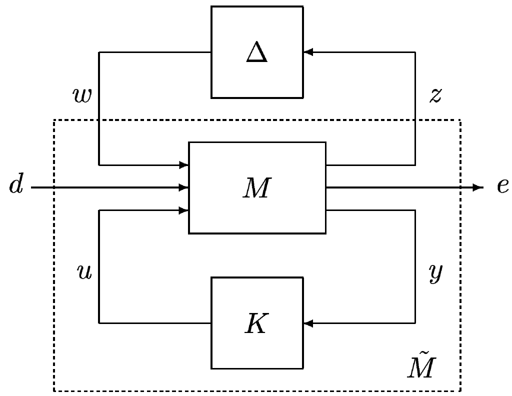


**Figure 3-3:** Uncertain system with controller feedback loop.

## 3.4 Optimal $\mathcal{H}_2$/Popov Controller Synthesis

In the $\mathcal{H}_2$/Popov controller synthesis problem, we consider a nominal system $M$ with uncertainty $\Delta$ and controller $K$, as shown in Fig. 3-3. The nominal closed-loop system is given by

$$\tilde{M} = \mathcal{F}(M, K) \ , \tag{3.43}$$

where $\mathcal{F}(\cdot,\cdot)$ denotes a linear-fractional transformation. For a given controller $K$, $\tilde{M}$ is fixed, and the performance of the closed-loop system can be analyzed using the methods of the previous section. Our goal is to determine the optimal controller $K$, *i.e.*, the stabilizing controller that minimizes the $\mathcal{H}_2$/Popov performance bound. In this section, a procedure for determining the optimal controller is developed, using linear matrix inequalities.

To begin, we need to form the closed-loop system $\tilde{M}$ from the dynamics of the plant $M$ and the controller $K$. The state space representation of the system $M$ is given by

$$\begin{aligned} \dot{x} &= Ax + B_u u + B_w w + B_d d \\ z &= C_z x \\ e &= \begin{bmatrix} C_e \\ 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ D_{eu} \end{bmatrix} u \\ y &= C_y x + D_{yd} d \ . \end{aligned} \tag{3.44}$$

The compensator is assumed to be dynamic, with no feedthrough terms since a direct feedthrough would cause the $\mathcal{H}_2$ cost of the closed-loop system to be infinite. The state space description is

$$\begin{aligned} \dot{x}_c &= A_c x_c + B_c y \\ u &= C_c x \ . \end{aligned} \tag{3.45}$$

The closed-loop system $\tilde{M}$ may therefore be described by an augmented state space description of the form

$$\begin{aligned} \dot{\tilde{x}} &= \tilde{A}\tilde{x} + \tilde{B}_w w + \tilde{B}_d d \\ z &= \tilde{C}_z \tilde{x} \\ e &= \tilde{C}_e \tilde{x} \ , \end{aligned} \tag{3.46}$$

where the augmented state is

$$\tilde{x} = \begin{bmatrix} x \\ x_c \end{bmatrix} \ , \tag{3.47}$$

and the augmented system matrices are

$$\begin{aligned} \tilde{A} &\triangleq \begin{bmatrix} A & B_u C_c \\ B_c C_y & A_c \end{bmatrix}, \ \tilde{B}_w \triangleq \begin{bmatrix} B_w \\ 0 \end{bmatrix}, \quad \tilde{B}_d \triangleq \begin{bmatrix} B_d \\ B_c D_{yd} \end{bmatrix}, \\ \tilde{C}_z &\triangleq \begin{bmatrix} C_z & 0 \end{bmatrix}, \quad \tilde{C}_e \triangleq \begin{bmatrix} C_e & 0 \\ 0 & D_{eu} C_c \end{bmatrix} . \end{aligned} \tag{3.48}$$

Note that the augmented system $\tilde{M}$ in Eqn. (3.46) has the same form as the system in Eqn. (3.3). Furthermore, the input-output map of the uncertainty block $\Delta$ is assumed to be sector bounded as shown in Fig. 3-2 and allowed to be nonlinear. The sector bound constraint for the uncertainty block $\Delta$ may also be represented by the inequality in Eqn. (3.5). Therefore, the cost-to-go for the uncertain system may be bounded by Eqn. (3.42), so that

$$\mathcal{J}(\mathcal{U}) = \mathrm{Tr}\ \left[\tilde{B}_d^T \left(\tilde{P} + \tilde{C}_z^T (M_2 - M_1) N \tilde{C}_z\right) \tilde{B}_d\right] \ , \tag{3.49}$$

where the Lyapunov matrix $\tilde{P}$ and the multiplier matrices $H$ and $N$ satisfy the matrix inequalities

$$\left[\begin{array}{c|c} \tilde{A}^T\left(\tilde{P}-\tilde{C}_z^T N M_1 \tilde{C}_z\right)+\left(\tilde{P}-\tilde{C}_z^T M_1 N \tilde{C}_z\right)\tilde{A} & \\ -\tilde{C}_z^T M_1 M_d H M_2 \tilde{C}_z - \tilde{C}_z^T M_2 H M_d M_1 \tilde{C}_z & (\cdots)^T \\ +\tilde{C}_e^T \tilde{C}_e & \\ \hline \tilde{B}_w^T\left(\tilde{P}-\tilde{C}_z N M_1 \tilde{C}_z\right)+N\tilde{C}_z\tilde{A} & -HM_d - M_d H \\ +M_d H M_2 \tilde{C}_z + H M_d M_1 \tilde{C}_z & +\tilde{B}_w^T \tilde{C}_z^T N + N\tilde{C}_z \tilde{B}_w \end{array}\right] \leq 0\,, \quad (3.50)$$

$$\tilde{P} > 0\,,\;\; H > 0\,,\;\; \text{and}\, N \geq 0\;. \qquad (3.51)$$

The closed-loop system is guaranteed to have $\mathcal{H}_2$ performance that is lower than the bound of Eqn. (3.49) when the matrix inequalities in Eqns. (3.50) and (3.51) are satisfied. Note, however, that Eqn. (3.50) is not a linear matrix inequality, because it involves nonlinear terms in $\tilde{P}$, $H$, $N$ and the compensator dynamics, *i.e.*, $A_c$, $B_c$, and $C_c$. Likewise, the $\mathcal{H}_2$ bound of Eqn. (3.49) is nonlinear in the unknown matrix variables.

Therefore, the problem, as described so far, is not an EVP. In the case of the $\mathcal{H}_2$ bound of Eqn. (3.49), the complication of the nonlinear term involving $\tilde{P}$ and $B_c$ may be eliminated by fixing $B_c$ to be some prespecified full column-rank matrix. This arbitrary choice of $B_c$ does not restrict the dynamics of the compensator, because the compensator is unique only to within a similarity transformation. So long as $B_c$ is chosen to be full column-rank, the set of possible controllers is not reduced. In this thesis, $B_c$ is chosen to equal $C_y^T$, although other choices are certainly possible. The $\mathcal{H}_2$ bound of Eqn. (3.49) is then linear in $\tilde{P}$ and $N$.

On the other hand, dealing with the nonlinear terms in the matrix inequality Eqn. (3.50) is more involved. In fact, the transformation of the robust performance matrix inequality constraint of Eqn. (3.50) to an equivalent LMI appears to be impossible. The next two sections describe how the matrix inequality of Eqn. (3.50) may be used as an LMI constraint when optimizing the $\mathcal{H}_2$ bound of Eqn. (3.49) with respect to the partial variable spaces of either the scalings $H$ and $N$ or the compensator dynamics. An iterative Popov controller design scheme may be adopted by repeating the partial optimizations until the $\mathcal{H}_2$ bound has converged and the optimal scalings H and N and compensator dynamics have been obtained. Since LMI optimizations are convex and of polynomial time complexity, this iterative scheme constitutes a computationally efficient $\mathcal{H}_2$/Popov controller synthesis procedure.

Since the scalings $H$ and $N$ may be interpreted as the Popov multiplier weighting matrices, the optimization of the $\mathcal{H}_2$ norm bound with respect to these scales is referred to as the Popov multiplier, or $W$, optimization. Similarly, the optimization of the $\mathcal{H}_2$ norm bound with respect to the compensator dynamics is referred to as the Popov controller, or $K$, optimization. Collectively, the iterative scheme involving both optimization steps is denoted the $W$-$K$ iteration.

### 3.4.1 Popov Multiplier Optimization

This section presents the optimization of the $\mathcal{H}_2$ norm bound with respect to the Popov multiplier matrices, *i.e.*, the dissipation scalings $H$ and $N$. Since the compensator dynamics are assumed fixed, this problem reduces to the robust performance analysis problem of Section 3.3. Indeed, this optimization is an EVP involving the minimization of the $\mathcal{H}_2$ norm bound defined in Eqn. (3.49) subject to the linear matrix inequality constraint in Eqn. (3.50) and the sign definiteness constraints $\tilde{P} > 0$, $H > 0$ and $N \geq 0$.

In fact, decomposing $\tilde{P}$ and assuming that $D_{yd} B_d^T = 0$, the bound on the $\mathcal{H}_2$ norm may be expressed as

$$\mathcal{J}(\mathcal{U}) = \operatorname{Tr}\left[B_d^T \tilde{P}_{11} B_d + B_d^T C_z^T (M_2 - M_1) N C_z B_d + D_{yd}^T B_c^T \tilde{P}_{22} B_c D_{yd}\right] . \tag{3.52}$$

Similarly, decomposing of $\tilde{P}$ and expanding the state space matrices of $\tilde{M}$ as defined in Eqn. (3.48), the robust stability constraint of Eqn. (3.50) becomes

$$\left[\begin{array}{c|c|c} \begin{array}{c} A^T(P_{11} - C_z^T N M_1 C_z) \\ +(P_{11} - C_z^T M_1 N C_z)A \\ C_y^T B_c^T P_{21} + P_{12} B_c C_y \\ -C_z^T M_1 M_d H M_2 C_z \\ -C_z^T M_2 H M_d M_1 C_z + C_e^T C_e \end{array} & (\cdots)^T & (\cdots)^T \\ \hline \begin{array}{c} C_c^T B_u^T (P_{11} - C_z^T N M_1 C_z) \\ +A_c^T P_{21} + P_{21} A + P_{22} B_c C_y \end{array} & \begin{array}{c} C_c^T B_u^T P_{12} + P_{21} B_u C_c \\ +A_c^T P_{22} + P_{22} A_c \\ +C_c^T D_{eu}^T D_{eu} C_c \end{array} & (\cdots)^T \\ \hline \begin{array}{c} B_w^T (P_{11} - C_z^T N M_1 C_z) \\ +M_d H M_2 C_z + H M_d M_1 C_z \\ +N C_z A \end{array} & B_w^T P_{12} + N C_z B_u C_c & \begin{array}{c} -H M_d - M_d H \\ +B_w^T C_z^T N + N C_z B_w \end{array} \end{array}\right] \leq 0 \; , \tag{3.53}$$

which is a linear matrix inequality in $\tilde{P}_{11}$, $\tilde{P}_{12} = \tilde{P}_{21}^T$, $\tilde{P}_{22}$, $H$ and $N$.

This optimization determines the optimal scalings $H$ and $N$ that result in the minimum $\mathcal{H}_2$ performance bound for the assumed controller dynamics. Since the optimization is specified as an EVP, it can be solved in polynomial time.

### 3.4.2 Popov Controller Optimization

This section develops a method for optimizing the $\mathcal{H}_2$ bound for the robust performance synthesis problem for fixed multiplier matrices $H$ and $N$. The optimization once again must minimize the $\mathcal{H}_2$ norm bound in Eqn. (3.49), or more explicitly Eqn. (3.52), subject to the matrix inequality constraint in Eqn. (3.50). Unfortunately, Eqn. (3.50) is nonlinear in $\tilde{P}$ and the compensator dynamics. The quadratic terms in the compensator dynamics may be eliminated by using Schur complements. Furthermore, bilinear terms in $\tilde{P}$, $N$ and the compensator dynamics may be eliminated using the approach presented in Section 2.2.

More specifically, letting

$$\Theta = \begin{bmatrix} A_c \\ C_c \end{bmatrix} , \tag{3.54}$$

the augmented system matrices $\tilde{A}$ and $\tilde{C}_e$ may be expressed as

$$\begin{aligned} \tilde{A} &= \tilde{A}_1 + \tilde{A}_2 \Theta \tilde{A}_3 , \\ \tilde{C}_e &= \tilde{C}_1 + \tilde{C}_2 \Theta \tilde{C}_3 , \end{aligned} \tag{3.55}$$

where

$$\begin{aligned} \tilde{A}_1 &\triangleq \begin{bmatrix} A & 0 \\ B_c C_y & 0 \end{bmatrix}, \quad \tilde{A}_2 \triangleq \begin{bmatrix} 0 & B_u \\ I & 0 \end{bmatrix}, \quad \tilde{A}_3 \triangleq \begin{bmatrix} 0 & I \end{bmatrix} , \\ \tilde{C}_1 &\triangleq \begin{bmatrix} C_e & 0 \\ 0 & 0 \end{bmatrix}, \quad \tilde{C}_2 \triangleq \begin{bmatrix} 0 & 0 \\ 0 & D_{eu} \end{bmatrix}, \quad \tilde{C}_3 \triangleq \begin{bmatrix} 0 & I \end{bmatrix} . \end{aligned} \tag{3.56}$$

For $\tilde{A}$ and $\tilde{C}_e$ as in Eqn. (3.55) the matrix inequality of Eqn. (3.50) is

$$\left[\begin{array}{c|c} \begin{array}{c} \left(\tilde{A}_1 + \tilde{A}_2\Theta\tilde{A}_3\right)^T\left(\tilde{P} - \tilde{C}_z^T N M_1 \tilde{C}_z\right) \\ + \left(\tilde{P} - \tilde{C}_z^T M_1 N \tilde{C}_z\right)\left(\tilde{A}_1 + \tilde{A}_2\Theta\tilde{A}_3\right) \\ -\tilde{C}_z^T M_1 M_d H M_2 \tilde{C}_z - \tilde{C}_z^T M_2 H M_d M_1 \tilde{C}_z \\ + \left(\tilde{C}_1 + \tilde{C}_2\Theta\tilde{C}_3\right)^T\left(\tilde{C}_1 + \tilde{C}_2\Theta\tilde{C}_3\right) \end{array} & (\cdots)^T \\ \hline \begin{array}{c} \tilde{B}_w^T\left(\tilde{P} - \tilde{C}_z N M_1 \tilde{C}_z\right) + N\tilde{C}_z\left(\tilde{A}_1 + \tilde{A}_2\Theta\tilde{A}_3\right) \\ + M_d H M_2 \tilde{C}_z + H M_d M_1 \tilde{C}_z \end{array} & \begin{array}{c} -HM_d - M_d H \\ + \tilde{B}_w^T \tilde{C}_z^T N + N\tilde{C}_z\tilde{B}_w \end{array} \end{array}\right] \leq 0 \ . \quad (3.57)$$

The quadratic $\Theta$ term in the upper left block of Eqn. (3.57), may be eliminated using Schur complements. The resulting matrix inequality is

$$\left[\begin{array}{c|c|c} \begin{array}{c} (\tilde{A}_1 + \tilde{A}_2\Theta\tilde{A}_3)^T(\tilde{P} - \tilde{C}_z^T N M_1 \tilde{C}_z) \\ +(\tilde{P} - \tilde{C}_z^T M_1 N \tilde{C}_z)(\tilde{A}_1 + \tilde{A}_2\Theta\tilde{A}_3) \\ -\tilde{C}_z^T M_1 M_d H M_2 \tilde{C}_z \\ -\tilde{C}_z^T M_2 H M_d M_1 \tilde{C}_z \end{array} & (\cdots)^T & (\cdots)^T \\ \hline \begin{array}{c} \tilde{B}_w^T(\tilde{P} - \tilde{C}_z N M_1 \tilde{C}_z) \\ +N\tilde{C}_z(\tilde{A}_1 + \tilde{A}_2\Theta\tilde{A}_3) \\ +M_d H M_2 \tilde{C}_z + H M_d M_1 \tilde{C}_z \end{array} & \begin{array}{c} -HM_d - M_d H \\ +\tilde{B}_w^T\tilde{C}_z^T N + N\tilde{C}_z\tilde{B}_w \end{array} & 0 \\ \hline (\tilde{C}_1 + \tilde{C}_2\Theta\tilde{C}_3) & 0 & -I \end{array}\right] \leq 0 \ . \quad (3.58)$$

Note that Eqn. (3.58) may be expressed as

$$\left[\begin{array}{c|c|c} \begin{array}{c} \tilde{A}_1^T(\tilde{P} - \tilde{C}_z^T N M_1 \tilde{C}_z) \\ +(\tilde{P} - \tilde{C}_z^T M_1 N \tilde{C}_z)\tilde{A}_1 \\ -\tilde{C}_z^T M_1 M_d H M_2 \tilde{C}_z - \tilde{C}_z^T M_2 H M_d M_1 \tilde{C}_z \end{array} & (\cdots)^T & (\cdots)^T \\ \hline \begin{array}{c} \tilde{B}_w^T(\tilde{P} - \tilde{C}_z N M_1 \tilde{C}_z) + N\tilde{C}_z\tilde{A}_1 \\ +M_d H M_2 \tilde{C}_z + H M_d M_1 \tilde{C}_z \end{array} & \begin{array}{c} -HM_d - M_d H \\ +\tilde{B}_w^T\tilde{C}_z^T N + N\tilde{C}_z\tilde{B}_w \end{array} & 0 \\ \hline \tilde{C}_1 & 0 & -I \end{array}\right]$$
$$+ \begin{bmatrix} (\tilde{P} - \tilde{C}_z^T M_1 N \tilde{C}_z)\tilde{A}_2 \\ N\tilde{C}_z\tilde{A}_2 \\ \tilde{C}_2 \end{bmatrix} \Theta \begin{bmatrix} \tilde{A}_3^T \\ 0 \\ 0 \end{bmatrix}^T + \begin{bmatrix} \tilde{A}_3^T \\ 0 \\ 0 \end{bmatrix} \Theta^T \begin{bmatrix} (\tilde{P} - \tilde{C}_z^T M_1 N \tilde{C}_z)\tilde{A}_2 \\ N\tilde{C}_z\tilde{A}_2 \\ \tilde{C}_2 \end{bmatrix}^T \leq 0 \ , \quad (3.59)$$

which is of the form

$$\Psi + U^T\Theta V + V^T\Theta^T U \leq 0 \ , \quad (3.60)$$

with

$$\Psi = \left[\begin{array}{c|c|c} \begin{array}{c}\tilde{A}_1^T(\tilde{P} - \tilde{C}_z^T N M_1 \tilde{C}_z) \\ +(\tilde{P} - \tilde{C}_z^T M_1 N \tilde{C}_z)\tilde{A}_1 \\ -\tilde{C}_z^T M_1 M_d H M_2 \tilde{C}_z \\ -\tilde{C}_z^T M_2 H M_d M_1 \tilde{C}_z\end{array} & (\cdots)^T & (\cdots)^T \\ \hline \begin{array}{c}\tilde{B}_w^T(\tilde{P} - \tilde{C}_z N M_1 \tilde{C}_z) + N\tilde{C}_z\tilde{A}_1 \\ +M_d H M_2 \tilde{C}_z + H M_d M_1 \tilde{C}_z\end{array} & \begin{array}{c}-HM_d - M_d H \\ +\tilde{B}_w^T \tilde{C}_z^T N + N\tilde{C}_z \tilde{B}_w\end{array} & 0 \\ \hline \tilde{C}_1 & 0 & -I \end{array}\right] , \quad (3.61)$$

$$U = \left[\begin{array}{ccc} \tilde{A}_2^T(\tilde{P} - \tilde{C}_z^T N M_1 \tilde{C}_z) & \tilde{A}_2^T C_z^T N & \tilde{C}_2^T \end{array}\right] , \quad (3.62)$$

$$V = \left[\begin{array}{ccc} \tilde{A}_3 & 0 & 0 \end{array}\right] . \quad (3.63)$$

From the elimination method of Section 2.2.1, Eqn. (3.59) is satisfiable if and only if

$$N_U^T \Psi N_U \leq 0 , \quad (3.64)$$

and

$$N_V^T \Psi N_V \leq 0 , \quad (3.65)$$

where $N_U$ and $N_V$ are matrices whose columns form basis for the nullspaces of $U$ and $V$ respectively. It is easily verified that $N_U$ and $N_V$ can be chosen to be

$$N_U = \left[\begin{array}{c|c|c|c} \tilde{R}\begin{bmatrix} I \\ 0 \end{bmatrix} & \begin{array}{c} 0 \\ 0 \end{array} & \begin{array}{c} 0 \\ 0 \end{array} & \begin{array}{c} 0 \\ 0 \end{array} \\ \hline M_1 \tilde{C}_z \tilde{R}\begin{bmatrix} I \\ 0 \end{bmatrix} & I & 0 & 0 \\ \hline \begin{array}{c} 0 \\ -(D_{eu}^T)^\dagger B_u^T \end{array} & \begin{array}{c} 0 \\ -(D_{eu}^T)^\dagger B_u^T C_z^T N \end{array} & \begin{array}{c} I \\ 0 \end{array} & \begin{array}{c} 0 \\ N_{(D_{eu}^T)} \end{array} \end{array}\right] , \quad (3.66)$$

$$N_V = \left[\begin{array}{c|c|c} I & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & I & 0 \\ \hline 0 & 0 & I \end{array}\right] , \quad (3.67)$$

where $\tilde{R} = \tilde{P}^{-1}$, $(D_{eu}^T)^\dagger$ is the pseudo-inverse of $D_{eu}^T$, and $N_{D_{eu}^T}$ is the null space of $D_{eu}^T$. Therefore, the matrix inequality constraint of Eqn. (3.59) is equivalent to the two matrix inequalities

$$
N_U^T \Psi N_U =
$$

$$
\left[\begin{array}{c|c|c|c}
\begin{array}{c}\tilde{R}_{11}\hat{A}^T + \hat{A}\tilde{R}_{11} \\ -B_u^T\left(D_{eu}^T D_{eu}\right)B_u\end{array} & (\cdots)^T & (\cdots)^T & 0 \\
\hline
\begin{array}{c}HC_z\tilde{R}_{11} + NC_z\hat{A}\tilde{R}_{11} + B_w^T \\ -NC_zB_u\left(D_{eu}^T D_{eu}\right)^{-1}B_u\end{array} & \begin{array}{c}-HM_d - M_dH \\ +\tilde{B}_w^T\tilde{C}_z^T N + N\tilde{C}_z\tilde{B}_w \\ -NC_zB_u\left(D_{eu}^T D_{eu}\right)^{-1}B_u^T C_z^T N\end{array} & 0 & 0 \\
\hline
C_e\tilde{R}_{11} & 0 & -I & 0 \\
\hline
0 & 0 & 0 & -I
\end{array}\right] \leq 0\ , \tag{3.68}
$$

and

$$
N_V^T \Psi N_V =
$$

$$
\left[\begin{array}{c|c|c}
\hat{A}^T\tilde{P}_{11} + \tilde{P}_{11}\hat{A} + C_y^T B_c^T \tilde{P}_{21} + \tilde{P}_{12}B_cC_y & (\cdots)^T & (\cdots)^T \\
\hline
HC_z + NC_z\hat{A} + B_w^T\tilde{P}_{11} & \begin{array}{c}-HM_d - M_dH \\ +\tilde{B}_w^T\tilde{C}_z^T N + N\tilde{C}_z\tilde{B}_w\end{array} & 0 \\
\hline
C_e & 0 & -I
\end{array}\right] \leq 0\ , \tag{3.69}
$$

where $\hat{A} \triangleq A + B_w M_1 C_z$, and

$$
\tilde{P} \triangleq \begin{bmatrix} \tilde{P}_{11} & \tilde{P}_{12} \\ \tilde{P}_{21} & \tilde{P}_{22} \end{bmatrix} \text{ and } \tilde{R} \triangleq \begin{bmatrix} \tilde{R}_{11} & \tilde{R}_{12} \\ \tilde{R}_{21} & \tilde{R}_{22} \end{bmatrix} . \tag{3.70}
$$

The fourth row and column of the matrix in Eqn. (3.68) trivially satisfy the sign definiteness constraint and are thus redundant. Also, since the third row and column of the matrix in Eqn. (3.69) involve constants, they may be folded inward to form the Schur complement. Therefore, the matrix inequality constraints in Eqns. (3.68) and (3.69) are respectively equivalent to

$$
\left[\begin{array}{c|c|c}
\tilde{R}_{11}\hat{A}^T + \hat{A}\tilde{R}_{11} - B_u^T\left(D_{eu}^T D_{eu}\right)B_u & (\cdots)^T & (\cdots)^T \\
\hline
\begin{array}{c}HC_z\tilde{R}_{11} + NC_z\hat{A}\tilde{R}_{11} + B_w^T \\ -NC_zB_u\left(D_{eu}^T D_{eu}\right)^{-1}B_u\end{array} & \begin{array}{c}-HM_d - M_dH \\ +B_w^T C_z^T N + NC_zB_w \\ -NC_zB_u\left(D_{eu}^T D_{eu}\right)^{-1}B_u^T C_z^T N\end{array} & 0 \\
\hline
C_e\tilde{R}_{11} & 0 & -I
\end{array}\right] \leq 0\ , \tag{3.71}
$$

$$\left[\begin{array}{c|c} \hat{A}^T\tilde{P}_{11}+\tilde{P}_{11}\hat{A} & \\ +C_y^TB_c^T\tilde{P}_{21}+\tilde{P}_{12}B_cC_y+C_e^TC_e & C_z^TH+\hat{A}^TC_z^TN+\tilde{P}_{11}B_w \\ \hline HC_z+NC_z\hat{A}+B_w^T\tilde{P}_{11} & -HM_d-M_dH+B_w^TC_z^TN+NC_zB_w \end{array}\right] \leq 0\ . \tag{3.72}$$

Note that the matrix inequalities in Eqns. (3.71) and (3.72) are linear in $\tilde{P}_{11}$, $\tilde{P}_{21}=\tilde{P}_{12}^T$, and $\tilde{R}_{11}$, and therefore comprise LMI constraints.

Note that the above formulation has assumed that $\tilde{P}=\tilde{R}^{-1}$, and therefore this equality constraint must be enforced. Because the $\mathcal{H}_2$/Popov performance bound of Eqn. (3.52) will tend to minimize $\tilde{P}$, enforcing the inequality constraint $\tilde{P}\geq\tilde{R}^{-1}$ suffices. In fact, as will be explained shortly, the freedom in the Lyapunov matrix $\tilde{P}$ may be used *a posteriori* to transform the active inequality constraint $\tilde{P}\geq\tilde{R}^{-1}$ to an equality.

In order to be able to use LMI optimization routines, the additional constraint $\tilde{P}\geq\tilde{R}^{-1}$ must be expressed as a linear matrix inequality. Indeed, note that the constraint $\tilde{P}\geq\tilde{R}^{-1}$ comprises the Schur complement of the equivalent linear matrix inequality constraint

$$\left[\begin{array}{c|c} \tilde{P} & I \\ \hline I & \tilde{R} \end{array}\right] \geq 0\ . \tag{3.73}$$

In fact, since the $\mathcal{H}_2$ bound of Eqn. (3.52) and the LMI constraints Eqns. (3.71) and (3.72) only depend on $\tilde{P}_{11}$, $\tilde{P}_{21}=\tilde{P}_{12}^T$, $\tilde{P}_{22}$, and $\tilde{R}_{11}$, the part of the LMI constraint of Eqn. (3.73) pertaining to the optimization is

$$\begin{bmatrix} \tilde{P}_{11} & \tilde{P}_{12} & I \\ \tilde{P}_{21} & \tilde{P}_{22} & 0 \\ I & 0 & \tilde{R}_{11} \end{bmatrix} \geq 0\ . \tag{3.74}$$

The optimization of the $\mathcal{H}_2$/Popov performance bound of Eqn. (3.52) with respect to the compensator dynamics has been reduced to an EVP involving the cost of Eqn. (3.52) subject to the LMI constraints of Eqns. (3.71), (3.72) and (3.74). The solution of this EVP yields the values of $\tilde{P}_{11}$, $\tilde{P}_{12}=\tilde{P}_{21}^T$, $\tilde{P}_{22}$, and $\tilde{R}_{11}$ that optimize the $\mathcal{H}_2$/Popov performance bound.

After completion of the performance objective optimization, the freedom in the Lyapunov matrix $\tilde{P}$ can be used to enforce the constraint $\tilde{P}=\tilde{R}^{-1}$. Examining the LMI constraints in Eqns. (3.71), (3.72) and (3.74), it can easily be verified that only the part of $\tilde{P}_{12}$ that is spanned by $B_c$ affects the inequalities, and therefore only this part is set by the optimization. Conversely, the part of the $\tilde{P}_{12}$ block in the null space of $B_c$ is free. These partitions of the $\tilde{P}_{12}$ will be denoted by $\tilde{P}_{12}^{||} = \left(\tilde{P}_{21}^{||}\right)^T$ and $\tilde{P}_{12}^{\perp} = \left(\tilde{P}_{21}^{\perp}\right)^T$ respectively.

The Schur complement of the LMI in Eqn. (3.74) is

$$\tilde{P}_{11} - \tilde{R}_{11}^{-1} \geq \tilde{P}_{12}^{||} \tilde{P}_{22}^{-1} \tilde{P}_{21}^{||} \ . \tag{3.75}$$

Augmenting the $\tilde{P}_{12} = \tilde{P}_{21}^T$ block to include the space perpendicular to $B_c$, the inequality of Eqn. (3.75) is transformed to the equality

$$\tilde{P}_{11} - \tilde{R}_{11}^{-1} = \left(\tilde{P}_{21}^{||} + \tilde{P}_{21}^{\perp}\right)^T \tilde{P}_{22}^{-1} \left(\tilde{P}_{21}^{||} + \tilde{P}_{21}^{\perp}\right) \ . \tag{3.76}$$

After introducing a similarity transformation $U$, Eqn. (3.76) may be written in the symmetric form

$$\left(\tilde{P}_{11} - \tilde{R}_{11}^{-1}\right)^{\frac{1}{2}} \left(\tilde{P}_{11} - \tilde{R}_{11}^{-1}\right)^{\frac{1}{2}} = \left[\left(\tilde{P}_{21}^{||} + \tilde{P}_{21}^{\perp}\right)^T \tilde{P}_{22}^{-\frac{1}{2}} U^T\right] \left[U \tilde{P}_{22}^{-\frac{1}{2}} \left(\tilde{P}_{21}^{||} + \tilde{P}_{21}^{\perp}\right)\right] \ . \tag{3.77}$$

Equating the factors of Eqn. (3.77) and performing some manipulations,

$$U^T = \tilde{P}_{22}^{-\frac{1}{2}} \tilde{P}_{21}^{||} \left(\tilde{P}_{11} - \tilde{R}_{11}^{-1}\right)^{-\frac{1}{2}} + \tilde{P}_{22}^{-\frac{1}{2}} \tilde{P}_{21}^{\perp} \left(\tilde{P}_{11} - \tilde{R}_{11}^{-1}\right)^{-\frac{1}{2}} \ . \tag{3.78}$$

Since $U$ is a similarity transformation, it must be full rank, *i.e.*, span the complete space. Therefore, the part of U not spanned by the term involving $\tilde{P}_{21}^{||}$ must be spanned by the term involving $\tilde{P}_{21}^{\perp}$. More explicitly,

$$\left[\tilde{P}_{22}^{-\frac{1}{2}} \tilde{P}_{21}^{||} \left(\tilde{P}_{11} - \tilde{R}_{11}^{-1}\right)^{-\frac{1}{2}}\right]^{\perp} = \tilde{P}_{22}^{-\frac{1}{2}} \tilde{P}_{21}^{\perp} \left(\tilde{P}_{11} - \tilde{R}_{11}^{-1}\right)^{-\frac{1}{2}} \ , \tag{3.79}$$

where the left hand side of Eqn. (3.79) denotes the part of the similarity transformation $U$ which is perpendicular to the term in the brackets, *i.e.*, the part of $U$ generated by $\tilde{P}_{21}^{||}$. Indeed, solving for $\tilde{P}_{21}^{\perp}$,

$$\tilde{P}_{21}^{\perp} = \tilde{P}_{22}^{\frac{1}{2}} \left[\tilde{P}_{22}^{-\frac{1}{2}} \tilde{P}_{21}^{||} \left(\tilde{P}_{11} - \tilde{R}_{11}^{-1}\right)^{-\frac{1}{2}}\right]^{\perp} \left(\tilde{P}_{11} - \tilde{R}_{11}^{-1}\right)^{\frac{1}{2}} \ , \tag{3.80}$$

and letting $\tilde{P}_{21} = \tilde{P}_{21}^{||} + \tilde{P}_{21}^{\perp}$ guarantees that the equality $\tilde{P} = \tilde{R}^{-1}$ is satisfied. Moreover, because $\tilde{P}_{21}^{\perp}$ does not affect the LMI constraints in Eqns. (3.71) and (3.72), the robust performance guarantees are not compromised.

The unknown compensator dynamics $A_c$ and $C_c$ may now be obtained by the feasibility solution of the matrix inequality of Eqn. (3.57). With $\tilde{P}$ and $\tilde{R}$ fixed, the matrix inequality in Eqn. (3.57) comprises an LMIP in the compensator dynamics.

### 3.4.3 Summary

The preceding sections have developed a $W$-$K$ iteration that solves the robust performance controller synthesis problem. In fact, it was shown that the robust performance criteria are equivalent to the ones obtained in Refs. 24 and 25. Each iteration step comprises of two optimizations, the first optimizes the $\mathcal{H}_2$ norm bound with respect to the scaling matrices $H$ and $N$, or $W$ optimization, and the second with respect to the compensator dynamics, or $K$ optimization. Both optimizations however, have been formulated in terms of standard LMI problems and are therefore convex and solvable in polynomial time. In order to calculate the optimal controller, this set of optimizations must be repeated iteratively until the $\mathcal{H}_2$/Popov performance bound has converged. Even though the $W$ and $K$ optimizations are individually convex, one cannot infer convexity for the combined problem. Consequently, the final controller may only be interpreted as being locally optimal.

# Chapter 4

# Implementation of Optimal $\mathcal{H}_2$/Popov Controller Synthesis and Example Controller Designs

In the previous chapter, an LMI-based iterative scheme was developed for designing performance robust controllers for systems involving nonlinear, sector-bounded uncertainties. The goals of this chapter are, first, to evaluate the accuracy of the $W$-$K$ iteration and second, to demonstrate its computational practicality and claim its potential use in the design of controllers for real-life, high-order systems. The study of the $W$-$K$ iteration is accomplished through the design of optimal $\mathcal{H}_2$/Popov controllers for two benchmark robust control problems introduced in Refs. 9 and 46.

Section 4.1 presents some important implementation issues pertaining to the $W$-$K$ iteration. The section introduces iteration parameters such as initial conditions or accuracy that may affect the performance of the $W$-$K$ iterative design scheme. In Section 4.2 the performance of the robust controller design scheme is evaluated by designing Popov controllers for example systems. First, the example designs are used to obtain the optimal $W$-$K$ iteration configuration with respect to the iteration parameters. This is done by determining the effects of the iteration parameters on the numerical characteristics and the computational time of the $W$-$K$ iterative control design scheme. Next, the $W$-$K$ iteration is compared to the gradient search method introduced in Refs. 24 and 25. Apart from validating the correctness of the $W$-$K$ iterative scheme, this comparison illustrates its numerical robustness and practicality.

## 4.1 Implementation Issues of the $W$-$K$ Iteration

This section presents the implementation issues that arose while developing the $W$-$K$ iteration scheme. These issues are crucial to the numerical stability and computational efficiency of the LMI based $W$-$K$ iteration scheme.

### 4.1.1 Initial Conditions

One of the difficulties that arises when using optimization schemes is the lack of initial conditions from which to initiate the optimization. Even though initial conditions are merely required to satisfy the optimization constraints, they are often hard to obtain. In fact, it is common practice to choose initial conditions that are educated guesses. Fortunately, the optimizations involved in the $W$-$K$ iteration are formulated in terms of standard LMI problems whose solution methods (LMILAB [6, 17, 18]) do not require initial conditions.

Conversely however, each of the optimizations within the iteration step requires partial knowledge of the variable space. In particular, the $W$ and $K$ optimizations require prior knowledge of the compensator dynamics and the Popov multiplier scalings, respectively. Since a performance-robust controller is not available *a priori* , the only viable approach is to choose initial values for the Popov multiplier matrices and to initiate the iteration with a $K$ optimization. There is no methodology for choosing the Popov multipliers that ensures the feasibility of the LMI constraints of the $K$ optimization.

Experimental analysis based on the two example systems presented in the following section has resulted in the feasibility characteristics shown in Table 4.1. In this table the the $\times$ and $\surd$ entries indicate that the $K$ optimization for both systems were found infeasible and feasible respectively. Although no feasibility guarantees may be provided, the choice of $H = I$ and $N = I$ (or $N = 10^{-6}\,I$) seem to constitute good initial guesses for the Popov multiplier scaling matrices. In this thesis, the $W$-$K$ iteration schemes are initialized with $H = N = I$.

Another pertinent issue is the effect of the initial conditions on the convergence characteristics of the $W$-$K$ iteration. It is often the case that a lucky choice of initial conditions can result in fewer iteration steps.

**Table 4.1:** Analysis of the feasibility of the initial $K$ optimization with respect to the initialization of the Popov multiplier scaling matrices $H$ and $N$.

| | | $N$ | | |
|---|---|---|---|---|
| | | $10^{-6}\,I$ | $I$ | $10^{6}\,I$ |
| $H$ | $10^{-6}\,I$ | $\times$ | $\times$ | $\times$ |
| | $I$ | $\surd$ | $\surd$ | $\times$ |
| | $10^{6}\,I$ | $\times$ | $\times$ | $\times$ |

An outline of the control flow of the complete $W$-$K$ iteration as prescribed by the above discussion is shown in Fig. 4-1.

### 4.1.2 Numerical Stability

Often, due to numerical imprecision, naive implementations of optimizations lead to ill-defined solutions. In particular, when constraints are formulated in terms of inequalities, the numerical imprecision may introduce additional degrees of freedom into the optimization problem that should be absent. In the case of the $K$ optimization, such problems arise with the variable $\tilde{P}_{22}$. When the $\tilde{P}_{22}$ block is assumed to be arbitrary, the part that theoretically does not affect the cost or the LMI constraints actually participates in the optimization due to numerical roundoff. More precisely, this part of $\tilde{P}_{22}$ tends to infinity, and therefore leads to solutions that are ill-defined.

This numerical problem is solved by *a priori* specifying the part of $\tilde{P}_{22}$ not affecting the optimization. It is easily seen that the $\mathcal{H}_2$ norm bound of Eqn. (3.52) in only affected by the part of $\tilde{P}_{22}$ that is aligned with $B_c$, denoted by $\tilde{P}_{22}^{\|}$. Therefore, the part of $\tilde{P}_{22}$ that does not affect the optimization, namely the part that is perpendicular to $B_c$, may be prespecified by letting

$$\begin{aligned}\tilde{P}_{22} &= \tilde{P}_{22}^{\|} + \tilde{P}_{22}^{\perp} \\ &= \left[\left(B_c^T\right)^{\dagger} \bar{P} B_c^{\dagger}\right] + \left[I - \left(B_c^T\right)^{\dagger} B_c^T\right],\end{aligned} \tag{4.1}$$

where $\left(B_c^T\right)^{\dagger}$ denotes the pseudo-inverse of $B_c^T$. Constraining $\tilde{P}_{22}$ to have this form eliminates the numerical problems encountered otherwise. A fortunate side-effect of letting $\tilde{P}_{22}$

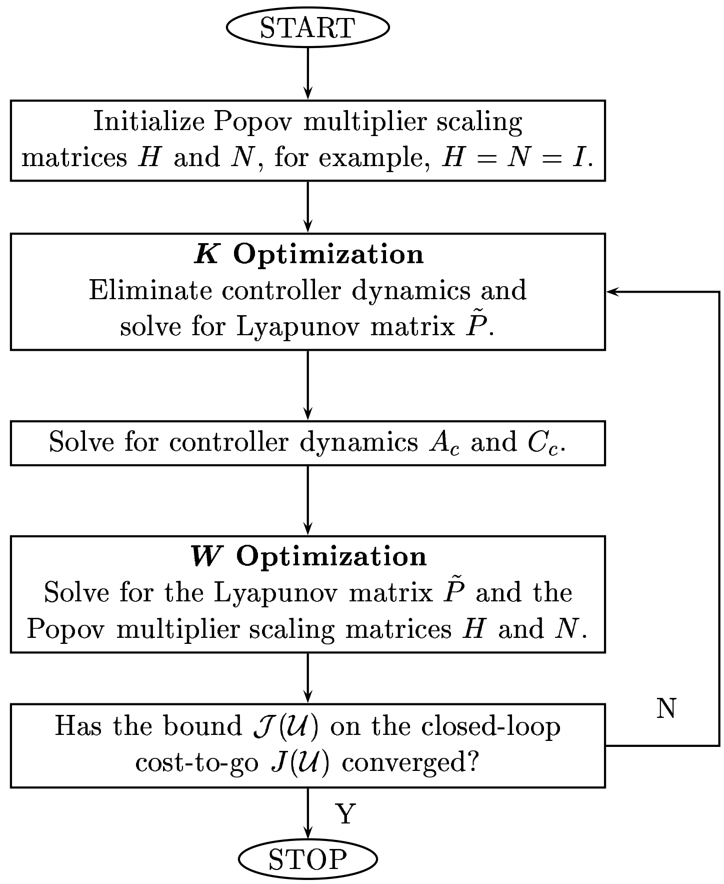



**Figure 4-1:** Flow diagram for the $W$-$K$ iteration.

have the form specified in Eqn. (4.1) is the reduction of the variable space of the optimization and consequently of the overall computational time.

### 4.1.3 Convergence Criterion

Another important consideration is the convergence criterion. In this thesis, a pragmatic approach to convergence is taken. The algorithm is assumed to have converged when an iteration fails to optimize the cost by more than some prespecified percentage. More precisely, the stopping criterion for the $W$-$K$ iteration involves the fractional reduction of the $\mathcal{H}_2$ norm bound during a complete $W$-$K$ iteration step. Ideally, the error in the $\mathcal{H}_2$ bound at each iteration would be used as the stopping criterion, *i.e.*, the iteration would stop when

$$\frac{\mathcal{J}(\mathcal{U})_k - \mathcal{J}(\mathcal{U})^*}{\mathcal{J}(\mathcal{U})^*} < \varepsilon \ , \tag{4.2}$$

where $\mathcal{J}(\mathcal{U})_k$ is the $\mathcal{H}_2$ bound at iteration $k$, $\mathcal{J}(\mathcal{U})^*$ is the optimal $\mathcal{H}_2$ bound, and $\varepsilon > 0$ is the stopping accuracy. Unfortunately, the optimal $\mathcal{H}_2$ bound is not known *a priori* and therefore the stopping criterion of Eqn. (4.2) is not feasible. Instead, the stopping criterion used depends on the fractional reduction in the $\mathcal{H}_2$ bound with respect to the $\mathcal{H}_2$ bound of the current iteration step. This stopping criterion may be written as

$$\frac{\mathcal{J}(\mathcal{U})_{k-1} - \mathcal{J}(\mathcal{U})_k}{\mathcal{J}(\mathcal{U})_k} < \varepsilon \ , \tag{4.3}$$

where $\mathcal{J}(\mathcal{U})_k$ and $\mathcal{J}(\mathcal{U})_{k-1}$ are the $\mathcal{H}_2$ norm bounds of iteration $k$ and $k-1$ respectively, and $\varepsilon > 0$ is the stopping accuracy. An unfortunate consequence of the use of such a stopping criterion is that if the cost function does not decrease significantly during a particular iteration step, the $W$-$K$ iteration will terminate prematurely and a suboptimal controller will result.

Although the stopping accuracy must be selected keeping in mind the accuracy required and the time limitations for the application at hand, it is important to note that the controllers obtained by intermediate iteration steps satisfy the robustness requirements. That is, although the performance may not be optimal, the performance obtained at each iteration step is guaranteed throughout the allowable uncertainty region defined by the specifications of the application. This is convenient, because it enables the use of intermediate controllers as substitutes for the optimal $\mathcal{H}_2$/Popov controller while the $W$-$K$ iteration is in progress. This may be of use in the rare occasions when the design of optimal $\mathcal{H}_2$/Popov controllers is done in real-time and a performance-robust controller is needed prior to the termination of the $W$-$K$ iteration.

Another important point is that the accuracy of the complete $W$-$K$ iteration is lower than the accuracy of the individual $W$ or $K$ optimizations. In fact, the complete $W$-$K$ iteration is only as accurate as the less accurate optimization it uses.

Throughout this thesis, the $W$-$K$ iteration was considered to have converged when the bound on the $\mathcal{H}_2$ norm had changed, fractionally, by less than $10^{-6}$ during a complete $W$-$K$ optimization pair.

### 4.1.4 Possible Iteration Optimizations

The fact that the constraints in both $W$ and $K$ optimizations are identical suggests that the efficiency of the $W$-$K$ iteration may be improved by taking advantage of initial conditions. Since the solution of one optimization satisfies the constraints of its succeeding optimization, it may be used as an initial value for this optimization. The use of initial conditions should eliminate any optimization effort used to match the optimal cost attained by the preceding optimization. Conversely, since the solution to each optimization is bound to be against the active EVP constraints, the use of such solutions may force the optimization to be performed along the constraint boundary and may therefore degrade computational efficiency. The following section uses a benchmark problem to shed some light onto this dilemma.

### 4.1.5 Compensator Dynamics Solution in the $K$ Optimization

Recall from Chapter 3 that in the $K$ optimization the cost function, as well as the robustness constraints were nonlinear. This obstacle was overcome by eliminating part of the variable space from the optimization. The resulting EVP could be used to optimize the $\mathcal{H}_2$ norm bound with respect to the Lyapunov matrix $\tilde{P}$ and its inverse $\tilde{R}$. Its solution could subsequently be used to solve for the compensator dynamics $A_c$ and $C_c$ using an LMIP.

In practice however, the LMIP used to solve for the compensator dynamics is often infeasible. Its infeasibility seems to stem from the solution of the EVP. Apparently, the numerical solution of the EVP is numerically too tight against its constraints and the subsequent LMIP becomes infeasible due to numerical imprecision. This problem is dealt with by introducing progressively stricter constraints in the corresponding EVP. In effect, when the resulting LMIP is infeasible, the previously solved EVP is resolved using stricter constraints, *i.e.*, supposing the initial EVP involved the LMI constraint $X \leq 0$ for some matrix $X$, the stricter form of the EVP would involve the LMI $X \leq -t\,I$ with the tolerance $t > 0$. Although this approach may lead to a reduction in the accuracy of the solution, it is believed to be the only viable solution.

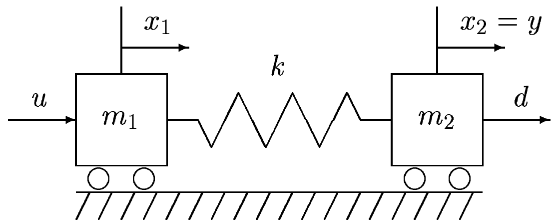



**Figure 4-2:** Mass-spring system.

### 4.1.6 Computer Resources

The computer platform used in all the designs present in this thesis was a SUN SPARC 20. Since the computational times are platform dependent, they should not be interpreted in an absolute but in a relative scale, *i.e.*, the ratio of the computational times of two particular controller designs should remain the same across all platforms.

## 4.2 Benchmark Problems

This section evaluates the iterative approach to Popov controller synthesis developed in the previous chapter. The evaluation is based on the use of the $W$-$K$ iteration to design Popov controllers for two robust control benchmark problems introduced in Refs. 9 and 46. Both benchmark problems involve real parametric uncertainties, and thus comprise valuable tests for the evaluation of Popov controller design schemes.

The first test example involves the noncollocated mass-spring system, presented in Ref. 46. The uncertainty in the system lies in the spring stiffness. The second benchmark problem involves the four disk system introduced in Ref. 9. This system involves multivariable uncertainty, namely, perturbations in two spring stiffnesses. This is an important benchmark problem, because slight stiffness perturbations result in large variations in the flexible modes of the system.

### 4.2.1 Mass-Spring System

The mass-spring system considered in this section is shown in Fig. 4-2. The importance of studying this system lies in the fact that it constitutes a generic model for an uncertain dynamical system that has one rigid-body and one vibrational mode.

The robust control problem, as introduced in Ref. 46, involves designing a controller that optimizes the guaranteed $\mathcal{H}_2$ performance at the output $y$, subject to uncertainty in the spring stiffness $k$. This control problem is noncollocated, because the input $u$ and the output $y$ are not at the same point. In fact, the control $u$ acts on $m_1$, and the output $y$ is measured at $x_2$. The nominal dynamics may be expressed as in Eqn. (3.44), with

$$
\begin{aligned}
A &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k_{\text{nom}} & k_{\text{nom}} & 0 & 0 \\ k_{\text{nom}} & -k_{\text{nom}} & 0 & 0 \end{bmatrix}, \quad B_u = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad B_d = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, \\
C_z &= \begin{bmatrix} 1 & -1 & 0 & 0 \end{bmatrix}, \qquad B_w = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \end{bmatrix}, \\
C_e &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad D_{eu} = \begin{bmatrix} 0 \\ \rho^{\frac{1}{2}} \end{bmatrix}, \\
C_y &= \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}, \qquad D_{yd} = \begin{bmatrix} 0 & \rho^{\frac{1}{2}} \end{bmatrix},
\end{aligned}
\tag{4.4}
$$

and $k_{\text{nom}} = 1$, and $\rho = 0.001$. For an actual spring stiffness of $k = k_{\text{nom}} + \Delta k$, the system dynamics involve the perturbed dynamics $A + B_w \Delta k C_z$. The goal of the synthesis procedure is to generate a controller that achieves good nominal performance, and guarantees robust stability and performance for perturbed spring stiffness values in the range $0.5 \le k \le 2$, *i.e.*, $-0.5 \le \Delta k \le 1$.

**Configuration of the *W*-*K* Iteration**

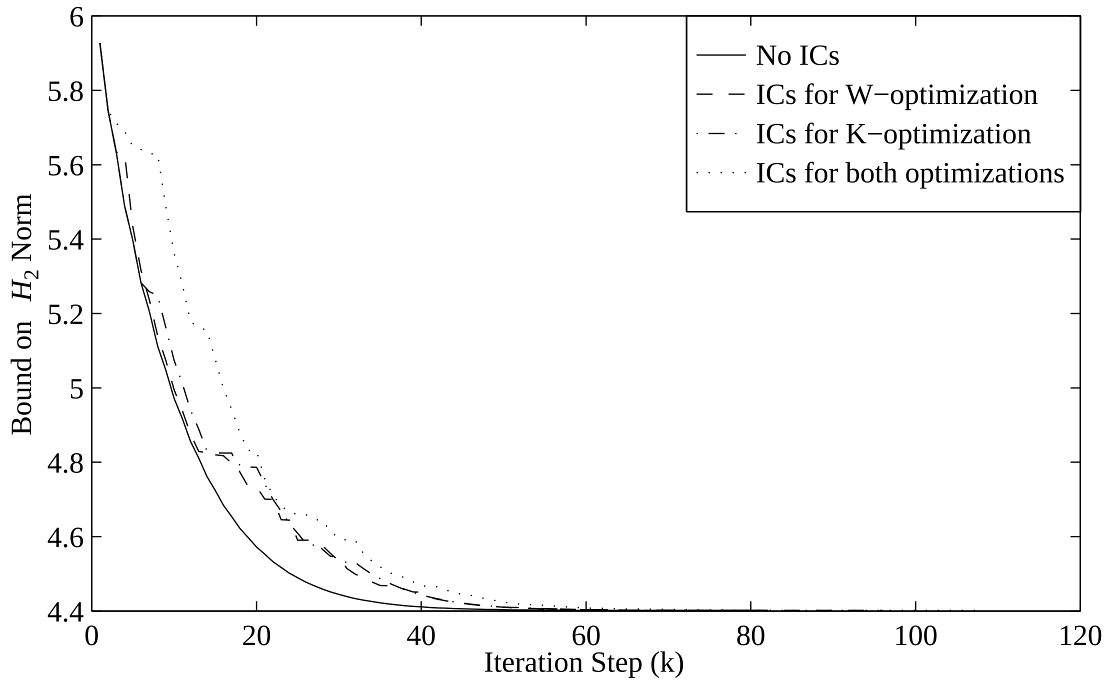
In the previous section, the question of whether the results of each optimization should be used to initialize the succeeding optimization was raised. This question can be answered by designing optimal $\mathcal{H}_2$/Popov controllers for the same perturbation ranges using the possible initial condition configurations. The results of such an approach are shown in Fig. 4-3. The figure presents plots of the performance objective (the bound on the $\mathcal{H}_2$ norm), first with respect to iteration step, and second with respect to the actual computation time. The difference between these two plots is that the computational effort of the iteration steps is not uniform. The computational time of each optimization is split between searching for the feasible region, and subsequently optimizing the $\mathcal{H}_2$ norm bound within the feasible region. The motivation behind the use of initial conditions is the attempt to eliminate the former part of the optimization, namely the search for the feasible region. It is important to note that all iteration schemes terminate with identical controller designs, and are therefore indistinguishable in that respect.

The convergence plots in Fig. 4-3 show that in certain configurations the optimizations are actually being hindered by the use of initial conditions. This is indicated by the relative roughness of the plots corresponding to runs using initial conditions. Although this may not seem crucial, the smoothness of the plot of the cost function with respect to iteration step is very important. This is because the $W$-$K$ iteration scheme uses an accuracy stopping criterion. Fig. 4-3 shows that the scheme where each iteration is started with random initial conditions results in a smooth optimization curve, and is therefore better behaved.

Apart from smoothness, however, total computational time is of great importance. The efficient use of the $W$-$K$ iteration scheme for large real life systems dictates the minimization of the computational time. To this extent, Fig. 4-3 indicates that the fastest configuration is the one where the optimal solution to the $K$ optimization is used to initialize the $W$ optimization. The second most efficient configuration is the one where no initial conditions are used.

The two considerations discussed above, namely smoothness of the optimization cost and the computational time, indicate that the scheme that does not use initial conditions combines smooth and well behaved optimization characteristics with low computational cost. This configuration is therefore used as the benchmark $W$-$K$ iteration scheme for the following sections.

Because of the obvious tradeoff between accuracy of solution and computational time, it is very important to select a desirable stopping accuracy. The stopping criterion for the $W$-$K$ iteration scheme involves the fractional change in the $\mathcal{H}_2$ bound with respect to the bound at the respective iteration step. Fig. 4-4 counterposes this fractional change to the error in the $\mathcal{H}_2$ norm bound at each iteration step. The iteration considered in this plot corresponds to the one of figure Fig. 4-3 in which no initial conditions are used. It seems that the error in the $\mathcal{H}_2$ norm bound is less that one order of magnitude higher than the fractional change at each iteration step. That is to say that if an accuracy of $10^{-4}$ is required, the iteration could be performed with an accuracy of $10^{-5}$. However, it is important to note that this rule is specific to this system and cannot be used as a general



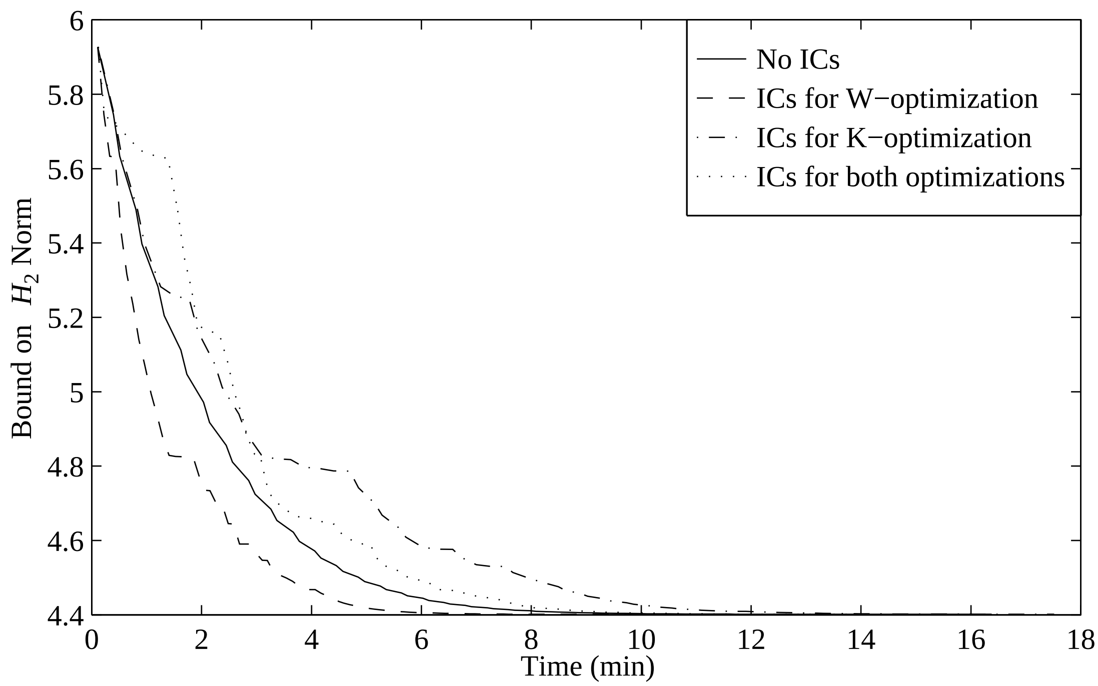




**Figure 4-3:** Progress of the $W$-$K$ iteration scheme during the design of the optimal $\mathcal{H}_2$/Popov controller for the mass-spring system with uncertainty lying in the range $-0.4 < \Delta k < 0.6$. The four curves represent $W$-$K$ iteration configurations in which initial conditions are i. not used, ii. used only in the $W$ optimizations, iii. used only in the $K$ optimizations and iv. used in both optimizations.

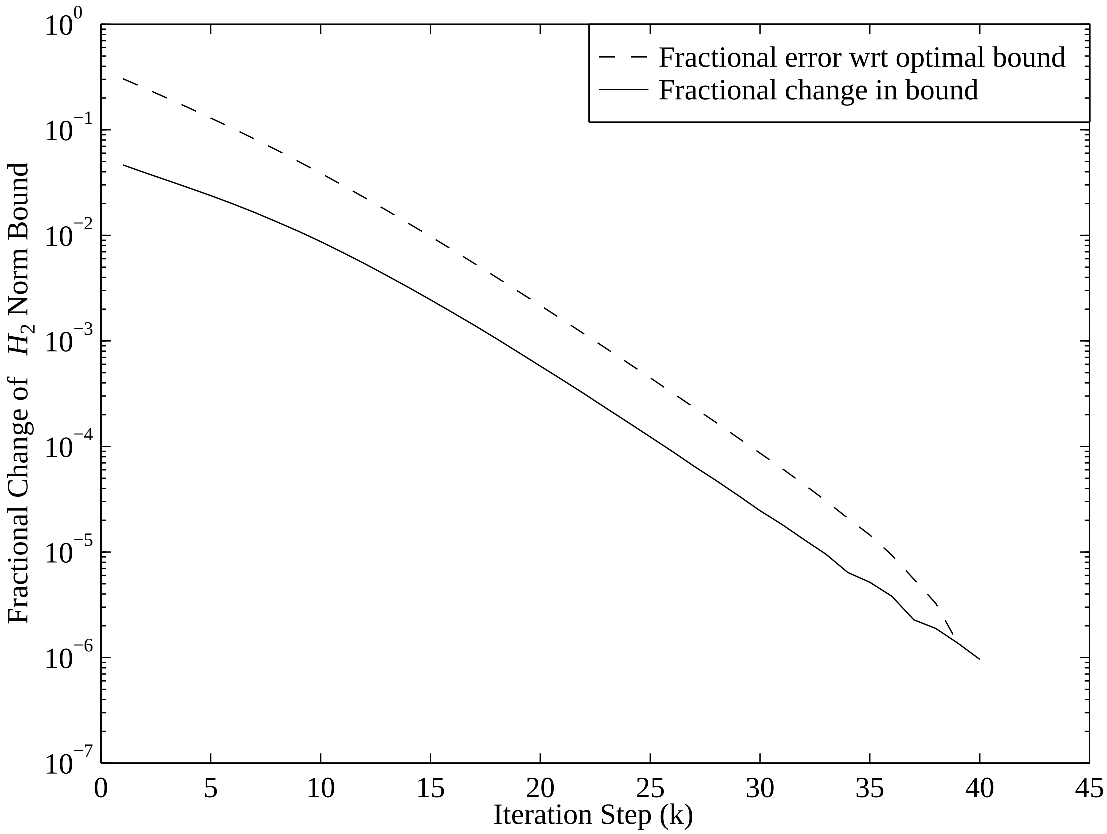



**Figure 4-4:** Fractional change in the $\mathcal{H}_2$ norm bound during the design of the optimal $\mathcal{H}_2$/Popov controller for the two mass-spring system with uncertainty lying in the range $-0.4 < \Delta k < 0.6$. The dashed curve represents the fractional error with respect to the final $\mathcal{H}_2$ bound, which is considered to be locally optimal, and the solid curve represents the fractional change with respect to the bound at each iteration step.

rule of thumb. In fact, the convergence plots of the example considered in the next section do not share these well-behaved convergence characteristics.

### Optimal $\mathcal{H}_2$/Popov Controller Designs obtained by the $W$-$K$ Iteration

In this section, several optimal $\mathcal{H}_2$/Popov controllers are designed for the mass-spring system. Since the $W$-$K$ iteration scheme is claimed to be equivalent to the gradient search approach introduced by How [24] and How *et al.* [25], it is only natural that their performance be compared. Therefore, controller designs are performed with perturbation bounds identical to the ones used for the designs by How. Two sets of controllers are constructed for the same perturbation bounds. The first set involves controllers using Popov multipliers of the form used by How, namely $W = I + Ns$. These designs are used in order to validate the correctness of the $W$-$K$ iteration, and in particular the Matlab routines developed. The second set of controllers involves the use of Popov multipliers of the form $W = H + Ns$. This set is used in order to determine how much the performance may be improved by using more general Popov multipliers, and furthermore to see how the computational time is affected when the variable space is increased.

As stated above, first the Popov multipliers are assumed to have the form $W = I + Ns$. For these cases, the performance of the optimal $\mathcal{H}_2$/Popov controllers is presented in Fig. 4-5 using so-called "performance buckets." Two curves correspond to each controller design. The "boxes" correspond to the performance guaranteed by the optimal $\mathcal{H}_2$/Popov controllers. Their widths represent the uncertainty ranges for which performance is guaranteed. The "buckets" comprise the actual performance obtained by the respective closed loop systems as a function of the perturbation of the stiffness $k$ from its nominal value $k_{\text{nom}}$. Note that the curves in Fig. 4-5 are normalized with respect to the nominal LQG $\mathcal{H}_2$ cost.

Even though the LQG design is guaranteed to obtain the minimum cost for the nominal plant, it is not very robust, as seen in Fig. 4-5. On the other hand, the $\mathcal{H}_2$/Popov controller designs sacrifice nominal performance in order to guarantee robust performance within the allowable perturbation ranges prescribed in each design. In fact, robust performance is actually ensured for a much larger perturbation range. This is an indication of the conservative nature of the design criteria. An interesting observation regarding this system is that the stability or performance guarantees for negative uncertainty are harder to attain. This can be inferred by the sharpness of the performance buckets, and by their distance to the respective robustness constraints.

Another interesting characteristic of the performance buckets of Fig. 4-5 is the discrepancy between the bound on the $\mathcal{H}_2$ norm of the system and the actual $\mathcal{H}_2$ norm. This discrepancy is another measure of the conservatism in the design criteria. In fact, the goal of most design techniques, such as real $\mu$-synthesis as opposed to complex $\mu$-synthesis, is to determine a bound that is less conservative, *i.e.*, to reduce the gap between the bound and the actual $\mathcal{H}_2$ norm. The stability robustness ranges and the corresponding $\mathcal{H}_2$ performance bounds are presented in Table 4.2 and Table 4.3 respectively. The actual $\mathcal{H}_2$ bounds obtained through the gradient search of Refs. 24 and 25 are not available.

Moreover, the optimal $\mathcal{H}_2$/Popov controllers yielded by the $W$-$K$ iteration are prac-

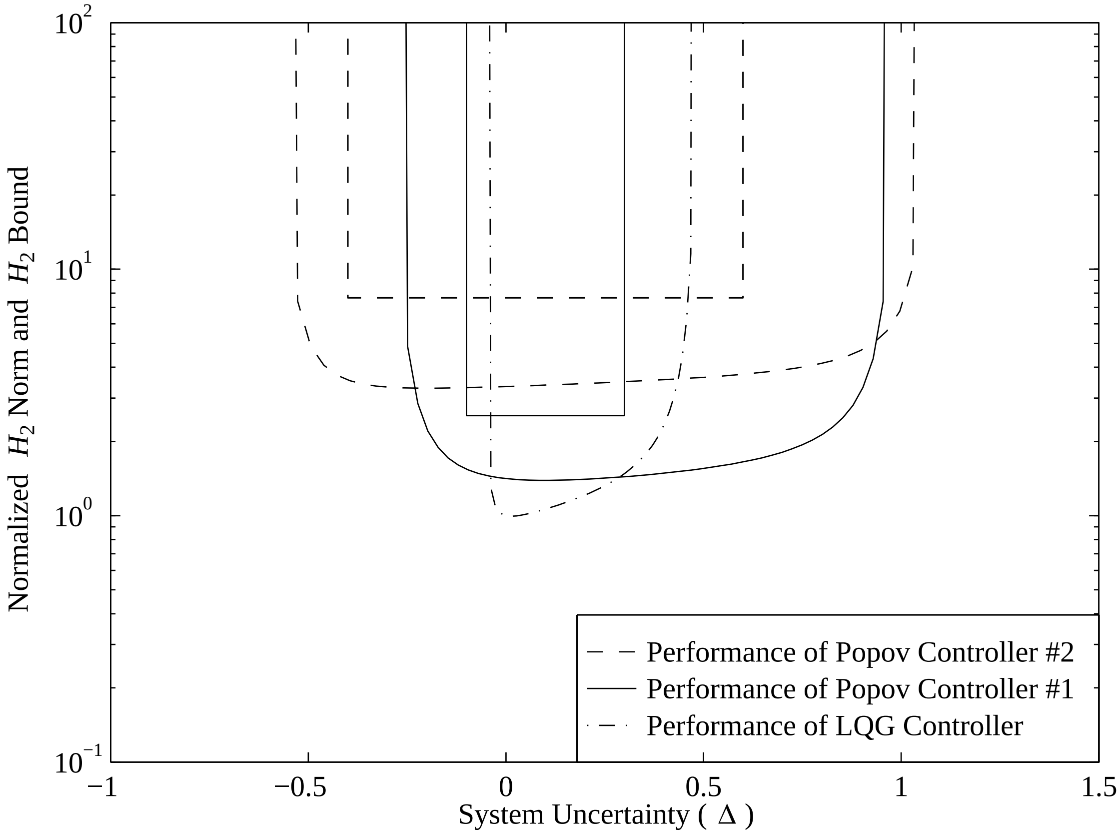



**Figure 4-5:** Performance of two optimal $\mathcal{H}_2$/Popov controllers. Each design is represented by two curves. The box corresponds to the guaranteed closed loop $\mathcal{H}_2$ performance and the bucket to the actual closed loop $\mathcal{H}_2$ performance attained by the respective controller. Recall that all curves are normalized by the nominal LQG $\mathcal{H}_2$ cost.

**Table 4.2:** Actual and guaranteed stability robustness bounds for optimal $\mathcal{H}_2$/Popov controller designs with stability multipliers of the form $W = I + Ns$.

| Controller Design | Actual Negative Stability Margin | Guaranteed Stab. Margins (Negative) | Guaranteed Stab. Margins (Positive) | Actual Positive Stability Margin |
|---|---|---|---|---|
| LQG | -0.0378 | — | — | 0.4032 |
| Popov # 1 | -0.1974 | -0.1000 | 0.3000 | 0.8521 |
| Popov # 2 | -0.5269 | -0.4000 | 0.6000 | 0.9962 |
| Popov # 2 (Ref. 24) | -0.55 | -0.4 | 0.6 | 1.05 |

**Table 4.3:** $\mathcal{H}_2$ norms and guaranteed $\mathcal{H}_2$ norm bounds for optimal $\mathcal{H}_2$/Popov controller designs with stability multipliers of the form $W = I + Ns$.

| Controller Design | Nominal $\mathcal{H}_2$ Norm Actual | Normalized | Bound on $\mathcal{H}_2$ Norm Actual | Normalized | Optimal Multiplier $W = I + Ns$ |
|---|---|---|---|---|---|
| LQG | 0.6018 | 1.0000 | — | — | — |
| Popov # 1 | 0.8515 | 1.4149 | 1.5313 | 2.5445 | $1 + 0.1720\,s$ |
| Popov # 2 | 2.0106 | 3.3409 | 4.6050 | 7.6520 | $1 + 0.3305\,s$ |
| Popov # 2 (Ref. 24) | — | 3.34 | — | — | $1 + 0.33\,s$ |

**Table 4.4:** Pole-zero characteristics of optimal $\mathcal{H}_2$/Popov controller designs with stability multipliers of the form $W = I + Ns$.

| Controller Design | Gain | Poles | Zeros |
|---|---|---|---|
| LQG | -1.2840 | $-1.7635 \pm 4.2636\,j$<br>$-4.9827 \pm 3.4291\,j$ | $-0.6481$<br>$0.1534 \pm 1.2282\,j$ |
| Popov # 1 | -0.7384 | $-5.8251 \pm 3.3421\,j$<br>$-2.2898 \pm 5.2190\,j$ | $-0.4452$<br>$0.1027 \pm 1.1828\,j$ |
| Popov # 2 | -0.3622 | $-7.1671 \pm 0.8233\,j$<br>$-3.6357 \pm 6.2257\,j$ | $-0.2905$<br>$0.0419 \pm 0.9320\,j$ |
| Popov # 2 (Ref. 24) | -0.36 | $-7.16 \pm 0.88\,j$<br>$-3.63 \pm 6.23\,j$ | $-0.29$<br>$0.04 \pm 0.93\,j$ |

**Table 4.5:** Computational time required for the design of optimal $\mathcal{H}_2$/Popov controllers with stability multipliers of the form $W = I + Ns$.

| Controller Design | Number of Iterations | Comp. Time (hr:min:sec) | Comp. Time/Iteration (sec) |
|---|---|---|---|
| Popov # 1 | 9 | 00:03:00 | 21 |
| Popov # 2 | 31 | 00:11:35 | 23 |

tically identical to those obtained in Ref. 24. This equivalence may easily be seen by comparing the pole-zero representations of the controllers in Table 4.4. The computational times for the iterations resulting in the respective controllers are shown in Table 4.5. Unfortunately, Refs. 24 and 25 do not specify the computational time requirements, so that a comparison of computational efficiency cannot be conducted.

In the case where the Popov multipliers are assumed to have the form $W = H + Ns$, the controllers are actually very different. This should be expected, because the use of a more

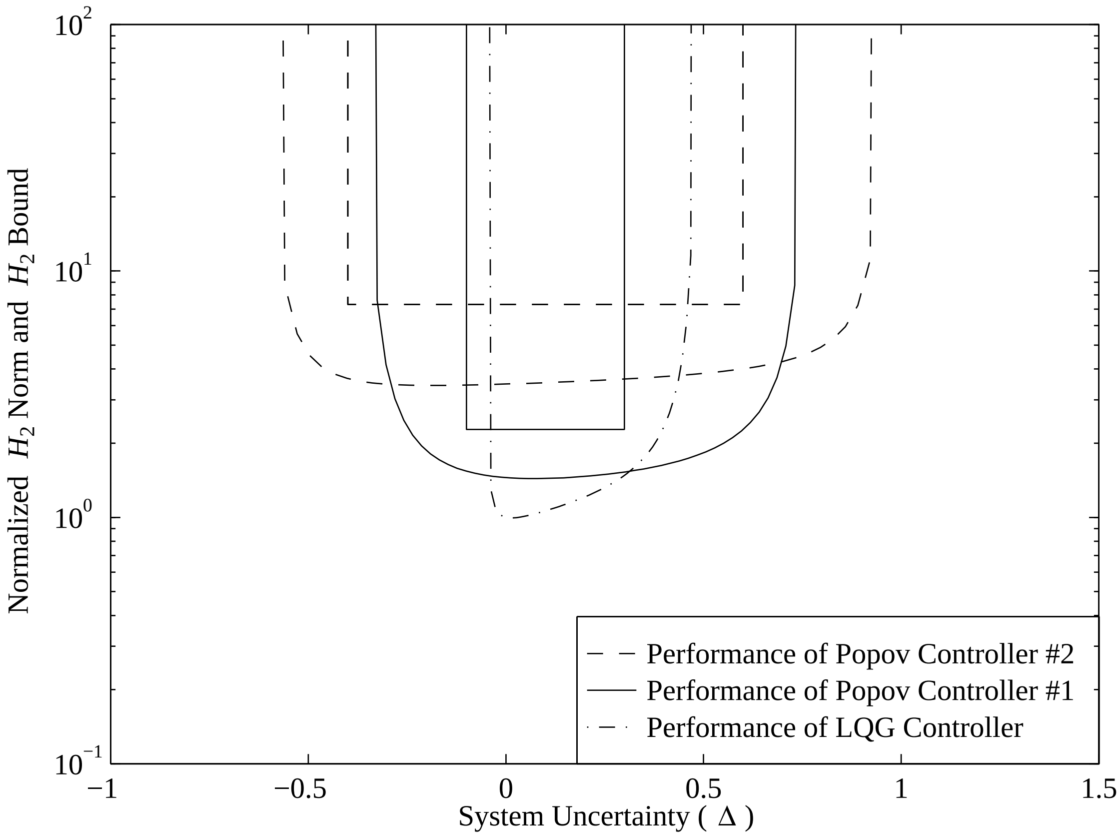



**Figure 4-6:** Performance of two optimal $\mathcal{H}_2$/Popov controllers. Each design is represented by two curves. The box corresponds to the guaranteed closed loop $\mathcal{H}_2$ performance and the bucket comprises the actual closed loop $\mathcal{H}_2$ performance attained by the respective controller. Recall that all curves are normalized by the nominal LQG $\mathcal{H}_2$ cost.

general form of Popov multipliers results in a less conservative performance bound. The performance buckets corresponding to the resulting Popov controller designs are shown in Fig. 4-6. The actual and guaranteed stability robustness bounds for these designs are shown in Table 4.6. The performance bounds are shown in Table 4.7. It is easily seen that even if the bound on the $\mathcal{H}_2$ norm has been improved in comparison with the previous controllers (Tables 4.2 and 4.3), the actual robustness ranges have shrunk. As would be expected, the reduction in the $\mathcal{H}_2$ bound, *i.e.*, the optimization cost, forces the performance of the controller closer to the robustness constraints.

The computational times for these $W$-$K$ iterations are shown in Table 4.9. A comparison to the computational times of the previous case where $W = I + Ns$, *i.e.*, Table 4.5, indicates that the computational time is very sensitive to variable space size. This dependence is very

**Table 4.6:** Actual and guaranteed stability robustness bounds for optimal $\mathcal{H}_2$/Popov controller designs with stability multipliers of the form $W = H + Ns$.

| Controller Design | Actual Negative Stability Margin | Guaranteed Stability Margin (Negative) | Guaranteed Stability Margin (Positive) | Actual Positive Stability Margin |
|---|---|---|---|---|
| LQG | -0.0378 | — | — | 0.4032 |
| Popov # 1 | -0.2357 | -0.1000 | 0.3000 | 0.5957 |
| Popov # 2 | -0.5280 | -0.4000 | 0.6000 | 0.8905 |

**Table 4.7:** $\mathcal{H}_2$ norms and guaranteed $\mathcal{H}_2$ norm bounds for optimal $\mathcal{H}_2$/Popov controller designs with stability multipliers of the form $W = H + Ns$.

| Controller Design | Nominal $\mathcal{H}_2$ Norm Actual | Nominal $\mathcal{H}_2$ Norm Normalized | Bound on $\mathcal{H}_2$ Norm Actual | Bound on $\mathcal{H}_2$ Norm Normalized | Optimal Multiplier $W = H + Ns$ |
|---|---|---|---|---|---|
| LQG | 0.6018 | 1.0000 | — | — | — |
| Popov # 1 | 0.8731 | 1.4508 | 1.3684 | 2.2738 | $0.4067 + 0.1034\,s$ |
| Popov # 2 | 2.0923 | 3.4765 | 4.4016 | 7.3141 | $0.6274 + 0.2348\,s$ |

**Table 4.8:** Pole-zero characteristics of optimal $\mathcal{H}_2$/Popov controller designs with stability multipliers of the form $W = H + Ns$.

| Controller Design | Gain | Poles | Zeros |
|---|---|---|---|
| LQG | -1.2840 | $-1.7635 \pm 4.2636\,j$<br>$-4.9827 \pm 3.4291\,j$ | $-0.6481$<br>$0.1534 \pm 1.2282\,j$ |
| Popov # 1 | -0.7574 | $-5.7758 \pm 3.2147\,j$<br>$-2.2456 \pm 5.2575\,j$ | $-0.4415$<br>$0.0144 \pm 1.1398\,j$ |
| Popov # 2 | -0.3573 | $-7.5389$<br>$-6.4451$<br>$-3.6762 \pm 6.5511\,j$ | $-0.2841$<br>$-0.0100 \pm 0.9054\,j$ |

important, since the ultimate goal is to use the $W$-$K$ iterative scheme to design optimal $\mathcal{H}_2$/Popov controllers for real-life systems, which are usually of high order.

It is interesting to note what the characteristics of the robustifying controller are in the case of the mass-spring system. Fig. 4-7 presents the frequency characteristics of the open loop system, the LQG controller, and an optimal $\mathcal{H}_2$/Popov controller guaranteeing robustness for uncertainties in the range $-0.4 < \Delta k < 0.6$.

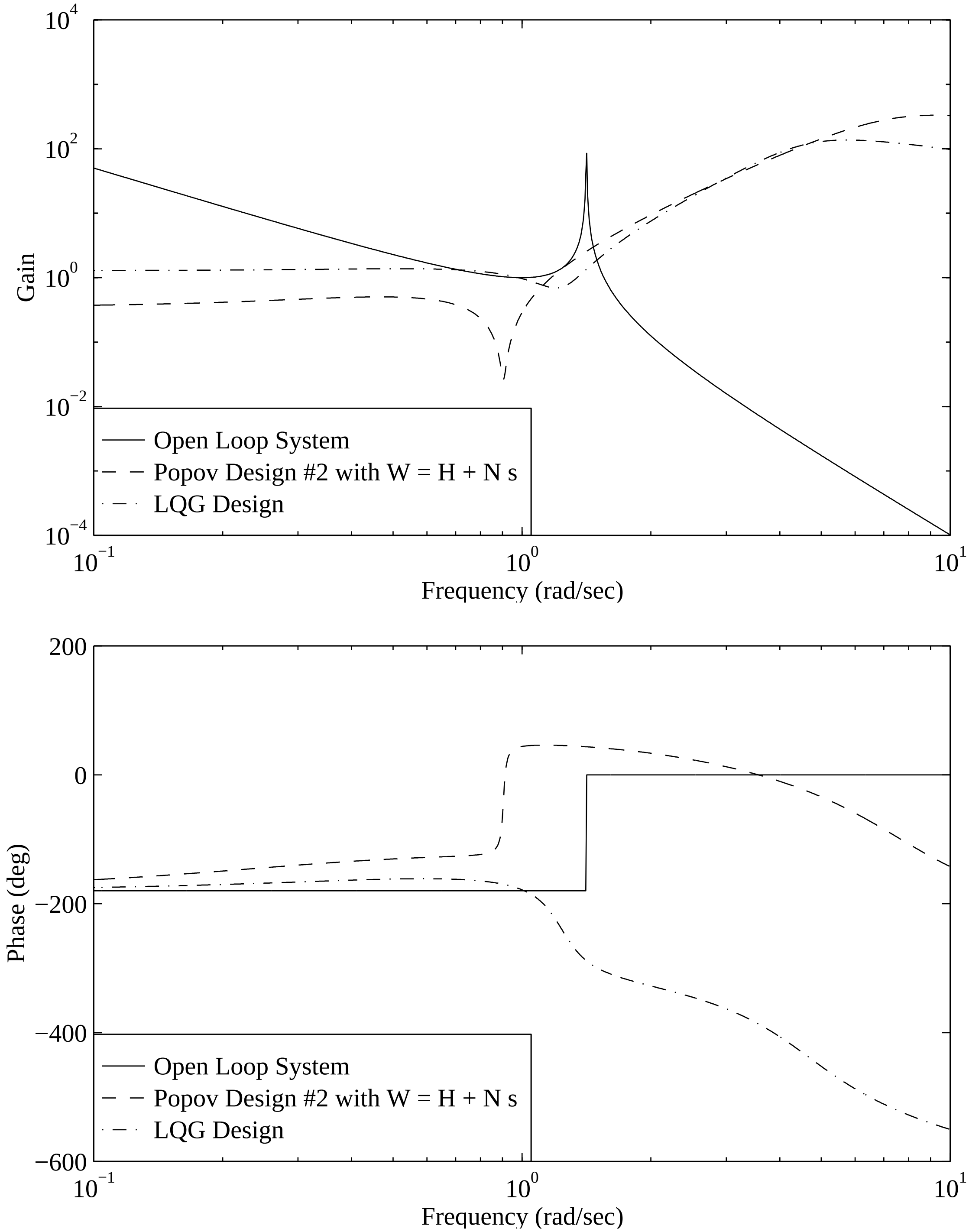



**Figure 4-7:** Comparison of the Bode plots corresponding to the system $M$, the LQG controller, and the Popov controller # 2. The stability multiplier used is of the form $W = H + Ns$.

**Table 4.9:** Computational time required for the design of optimal $\mathcal{H}_2$/Popov controllers with stability multipliers of the form $W = H + Ns$.

| Controller Design | Number of Iterations | Comp. Time (hr:min:sec) | Comp. Time/Iteration (sec) |
|---|---|---|---|
| Popov # 1 | 28 | 00:09:59 | 22 |
| Popov # 2 | 41 | 00:16:22 | 24 |

### 4.2.2 Coupled Rotating Disk System

This section considers the coupled rotating disk system introduced by Cannon and Rosenthal [9]. The system is comprised of four disks connected by a flexible shaft, as shown in Fig. 4-8, and is representative of lightly damped structures. This system is of twice the order of the mass-spring system considered in the previous section. More importantly, it involves multivariable uncertainty, namely uncertainty in the spring stiffnesses $k_1$ and $k_3$. The system dynamics are given by the state space representation of Eqn. (3.44), with

$$
\begin{aligned}
A &= \begin{bmatrix} 0 & I \\ -J^{-1}K & -J^{-1}D \end{bmatrix}, \\
B_u &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{1}{m} & 0 & 0 \end{bmatrix}^T, \qquad B_d = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{m} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T, \\
C_z &= \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad B_w = \begin{bmatrix} 0 & 0 & 0 & 0 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}^T, \\
C_e &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0.1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad D_{eu} = \begin{bmatrix} 0 \\ \rho^{\frac{1}{2}} \end{bmatrix}, \\
C_y &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad D_{yd} = \begin{bmatrix} 0 & \rho^{\frac{1}{2}} \end{bmatrix},
\end{aligned} \tag{4.5}
$$

where

$$
J = m \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad K = k \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}, \quad D = d \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}. \tag{4.6}
$$

Moreover, $m = k = 1$, the damping factor is $d = 0.01$, and $\rho = 0.005$. The two spring stiffnesses, $k_1$ and $k_3$, are assumed to be independent.

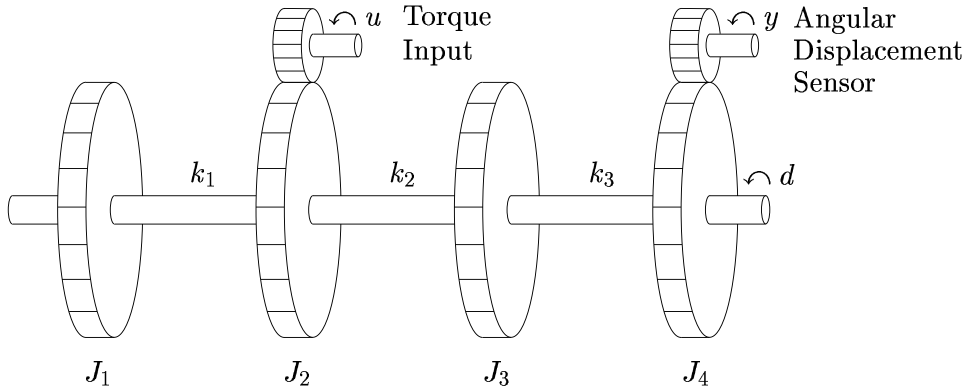



**Figure 4-8:** Coupled rotating four disk system. The control input $u$ is used to counteract the disturbance input $d$. The system is assumed to be uncertain in the spring stiffnesses $k_1$ and $k_3$.

The importance of considering this system lies in the fact that slight perturbations in the uncertain stiffnesses cause large phase variations in the system dynamics. In the case of lightly damped systems, such variations may lead to system instability. Fig. 4-9 presents the variation in the frequency characteristics of the four disk problem for 5% perturbations in the spring stiffnesses. These slight perturbations introduce 100 deg phase variations in the plant.

For comparison, the Popov controller designs were performed with perturbation bounds identical to the ones used for the designs in Refs. 24 and 25. The presentation of the results is similar to that of the previous section, with the exception that optimal $\mathcal{H}_2$/Popov controllers were generated only for Popov multipliers of the form $W = H + Ns$. Tables 4.10 and 4.11 present the robustness ranges and the performance bounds obtained for the different optimal $\mathcal{H}_2$/Popov controller designs. Table 4.12 includes the pole-zero representations of the optimal $\mathcal{H}_2$/Popov controller designs and Table 4.13 presents the optimal Popov multipliers obtained. A comparison between the transfer functions of the LQG compensator and one of the optimal $\mathcal{H}_2$/Popov designs is presented in Fig. 4-10.

The performance buckets for two controller designs are present in Fig. 4-11. As in case of the mass-spring system, the disk system is harder to robustify for negative perturbations. Once again, this may be inferred from the fact that the $\mathcal{H}_2$ norm buckets are closest to the curves of guaranteed performance in the negative perturbation region.

The computational time requirements for the various designs, present in Table 4.14, indicate that larger systems will require lengthy iterations. However, since the controller

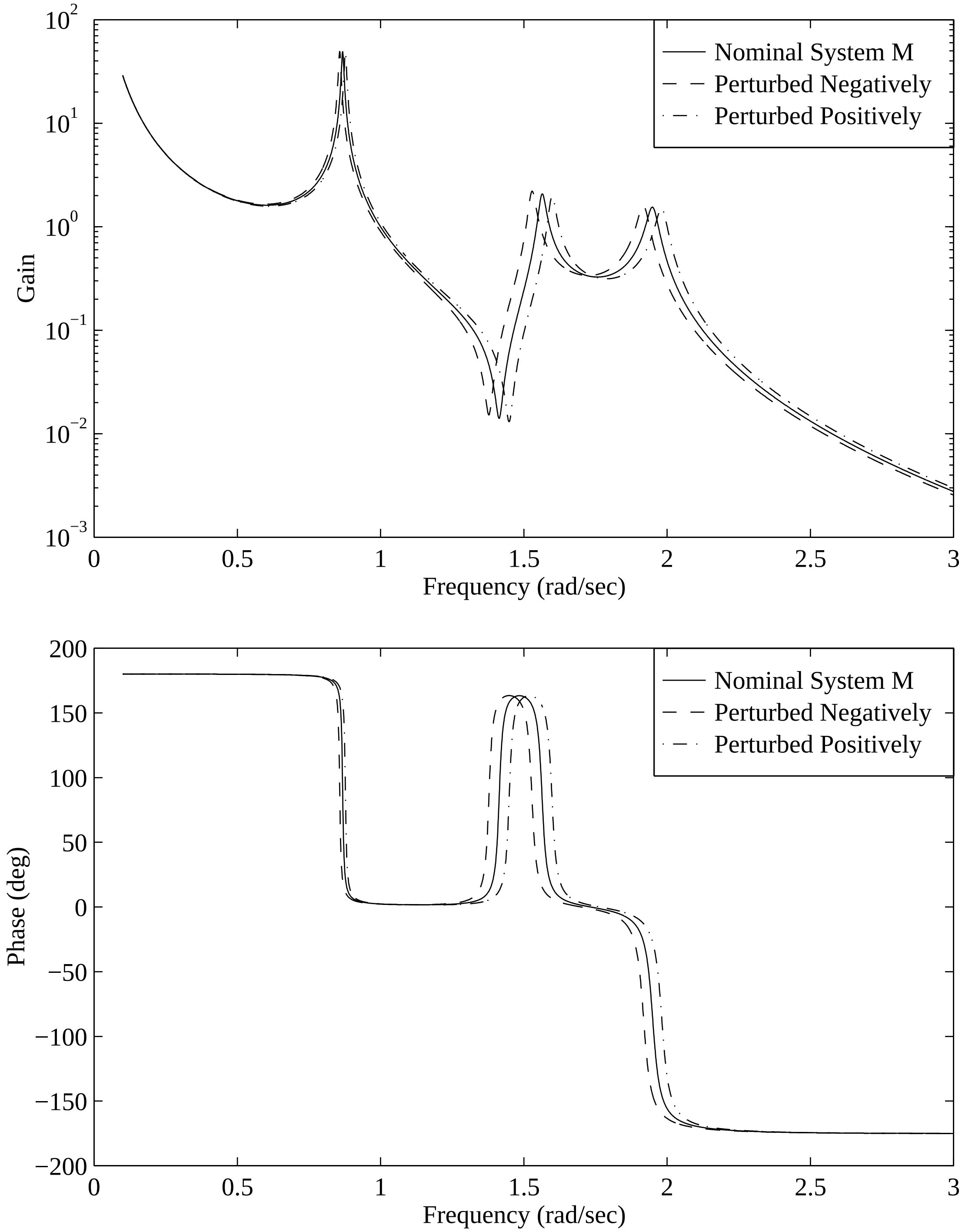



**Figure 4-9:** Effects of slight spring stiffness perturbations to the open loop system dynamics. The sector bound for the system perturbations in this case is given by $M_2 = -M_1 = 0.05\, I$ and corresponds to a 5% uncertainty in each of the spring stiffnesses $k_1$ and $k_3$. Such perturbations actually result in phase uncertainty of more than 100 deg.

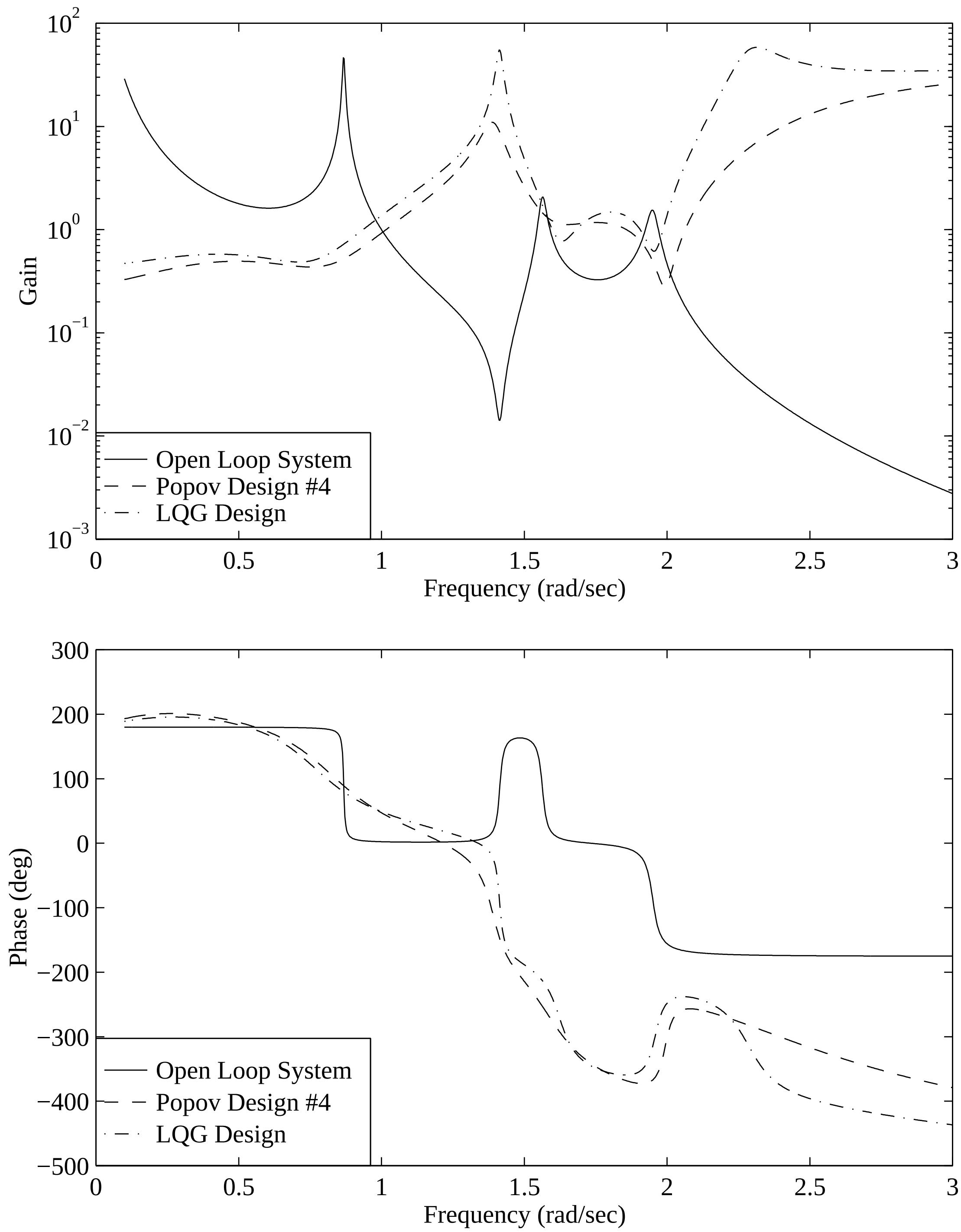



**Figure 4-10:** Comparison of the Bode plots corresponding to the open loop system $M$, the LQG controller and the Popov controller # 4. The stability multiplier used is of the form $W = H + Ns$.

**Table 4.10:** Actual and guaranteed stability robustness bounds for optimal $\mathcal{H}_2$/Popov controller designs with stability multipliers of the form $W = H + Ns$.

| Controller Design | Actual Negative Stability Margin | Guaranteed Stab. Margins (Negative) | Guaranteed Stab. Margins (Positive) | Actual Positive Stability Margin |
|---|---|---|---|---|
| Popov # 1 | -0.0256 | -0.0190 | 0.0190 | 0.0581 |
| Popov # 2 | -0.0311 | -0.0240 | 0.0240 | 0.0719 |
| Popov # 3 | -0.0432 | -0.0350 | 0.0350 | 0.0987 |
| Popov # 4 | -0.0609 | -0.0510 | 0.0510 | 0.1400 |
| Popov # 1 (Ref. 24) | -0.025 | -0.019 | 0.019 | 0.060 |
| Popov # 2 (Ref. 24) | -0.033 | -0.024 | 0.024 | 0.075 |
| Popov # 3 (Ref. 24) | -0.045 | -0.035 | 0.035 | 0.100 |
| Popov # 4 (Ref. 24) | -0.063 | -0.051 | 0.051 | 0.140 |

**Table 4.11:** $\mathcal{H}_2$ norms and guaranteed $\mathcal{H}_2$ norm bounds for optimal $\mathcal{H}_2$/Popov controller designs with stability multipliers of the form $W = H + Ns$.

| Controller Design | Nominal $\mathcal{H}_2$ Norm Actual | Nominal $\mathcal{H}_2$ Norm Normalized | Bound on $\mathcal{H}_2$ Norm Actual | Bound on $\mathcal{H}_2$ Norm Normalized |
|---|---|---|---|---|
| LQG | 2.2062 | 1 | — | — |
| Popov # 1 | 2.4878 | 1.1277 | 2.8363 | 1.2856 |
| Popov # 2 | 2.5697 | 1.1648 | 2.9925 | 1.3564 |
| Popov # 3 | 2.7488 | 1.2460 | 3.3372 | 1.5126 |
| Popov # 4 | 3.0150 | 1.3666 | 3.8555 | 1.7476 |
| Popov # 1 (Ref. 24) | — | 1.12 | — | — |
| Popov # 2 (Ref. 24) | — | 1.17 | — | — |
| Popov # 3 (Ref. 24) | — | 1.25 | — | — |
| Popov # 4 (Ref. 24) | — | 1.38 | — | — |

designs at each iteration step satisfy the robustness requirement, it is possible to continuously update the optimal controller characteristics as the $W$-$K$ iteration progresses.

Fig. 4-12 presents the fractional change of the bound on the $\mathcal{H}_2$ norm with respect to iteration step. As a measure of the optimality of the intermediate controllers, the error in the bound to the $\mathcal{H}_2$ norm is also plotted. In particular, the plot indicates that the error in the $\mathcal{H}_2$ bound is below 1% after half of the iteration steps.

The sharp jumps that are present in Fig. 4-12 indicate that some of the intermediate iterations do not optimize the cost nearly as much as their neighboring iterations. This should be taken under consideration when choosing the stopping accuracy. What should be avoided is the premature termination of the iteration and subsequently the generation of a

**Table 4.12:** Pole-zero characteristics of optimal $\mathcal{H}_2$/Popov controller designs with stability multipliers of the form $W = H + Ns$.

| Controller Design | Gain | Poles | Zeros |
|---|---|---|---|
| LQG | -0.4544 | $-0.0110 \pm 1.4138\,j$<br>$-0.0781 \pm 2.2860\,j$<br>$-2.0071 \pm 1.4774\,j$<br>$-2.4419 \pm 2.7548\,j$ | $-0.0283 \pm 1.9588\,j$<br>$0.0465 \pm 1.6231\,j$<br>$0.1626 \pm 0.7677\,j$<br>$-0.3062$ |
| Popov # 1 | -0.3787 | $-2.4145 \pm 2.7874\,j$<br>$-0.2487 \pm 2.3472\,j$<br>$-0.0203 \pm 1.4065\,j$<br>$-2.1756 \pm 1.5202\,j$ | $-0.0296 \pm 1.9679\,j$<br>$0.0778 \pm 1.6010\,j$<br>$0.1674 \pm 0.7892\,j$<br>$-0.2710$ |
| Popov # 2 | -0.3648 | $-2.4122 \pm 2.7938\,j$<br>$-0.2939 \pm 2.3556\,j$<br>$-0.0230 \pm 1.4042\,j$<br>$-2.2136 \pm 1.5293\,j$ | $-0.0285 \pm 1.9719\,j$<br>$0.0833 \pm 1.5980\,j$<br>$0.1686 \pm 0.7940\,j$<br>$-0.2645$ |
| Popov # 3 | -0.3385 | $-2.4040 \pm 2.8247\,j$<br>$-0.3968 \pm 2.3614\,j$<br>$-0.0288 \pm 1.3988\,j$<br>$-2.2958 \pm 1.5405\,j$ | $-0.0261 \pm 1.9807\,j$<br>$0.0949 \pm 1.5932\,j$<br>$0.1709 \pm 0.8042\,j$<br>$-0.2521$ |
| Popov # 4 | -0.3065 | $-2.4138 \pm 2.8774\,j$<br>$-2.4445 \pm 1.5455\,j$<br>$-0.5541 \pm 2.3227\,j$<br>$-0.0382 \pm 1.3904\,j$ | $-0.0252 \pm 1.9936\,j$<br>$0.1112 \pm 1.5899\,j$<br>$0.1742 \pm 0.8198\,j$<br>$-0.2370$ |
| Popov # 2 (Ref. 24) | -0.36 | $-0.02 \pm 1.40\,j$<br>$-0.31 \pm 2.35\,j$<br>$-2.18 \pm 1.51\,j$<br>$-2.41 \pm 2.85\,j$ | $0.17 \pm 0.80\,j$<br>$0.08 \pm 1.60\,j$<br>$-0.03 \pm 1.97\,j$<br>$-0.26$ |

**Table 4.13:** Optimal multipliers of the form $W = H + Ns$ for the optimal $\mathcal{H}_2$/Popov controller designs.

| Controller Design | Optimal Multiplier $W = H + Ns$ |
|---|---|
| Popov # 1 | $\begin{bmatrix} 0.8388 + 0.1561\,s & 0 \\ 0 & 2.8186 + 0.7524\,s \end{bmatrix}$ |
| Popov # 2 | $\begin{bmatrix} 0.9161 + 0.2010\,s & 0 \\ 0 & 2.7718 + 0.8182\,s \end{bmatrix}$ |
| Popov # 3 | $\begin{bmatrix} 1.0891 + 0.3149\,s & 0 \\ 0 & 2.7311 + 0.9668\,s \end{bmatrix}$ |
| Popov # 4 | $\begin{bmatrix} 1.3583 + 0.5428\,s & 0 \\ 0 & 2.7544 + 1.2231\,s \end{bmatrix}$ |
| Popov # 2 (Ref. 24) | $\begin{bmatrix} 1 + 0.16\,s & 0 \\ 0 & 2.9(1 + 0.27\,s) \end{bmatrix}$ |

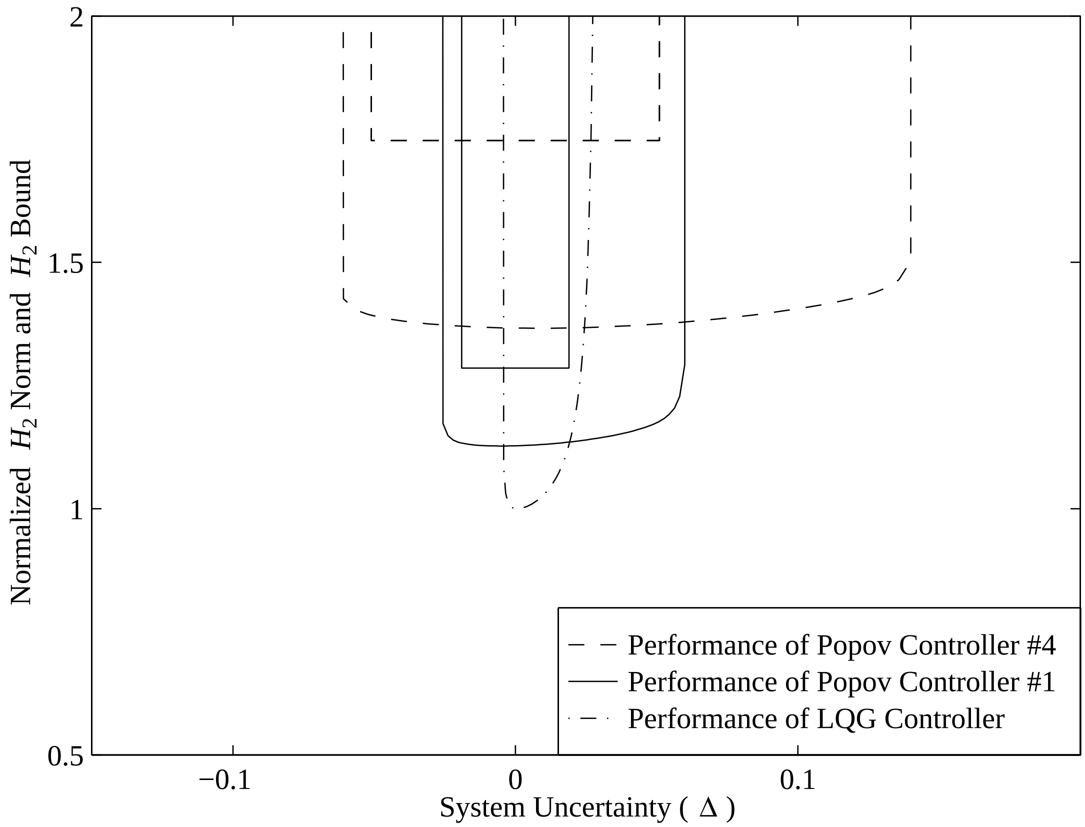



**Figure 4-11:** Performance of two optimal $\mathcal{H}_2$/Popov controllers. Each design is represented by two curves. The box corresponds to the closed loop $\mathcal{H}_2$ performance guaranteed by the controller and the bucket comprises the actual closed loop $\mathcal{H}_2$ performance attained by the use of the respective controller. Recall that all curves are normalized by the nominal LQG $\mathcal{H}_2$ cost.

**Table 4.14:** Computational time required for the design of optimal $\mathcal{H}_2$/Popov controllers with stability multipliers of the form $W = H + Ns$.

| Controller Design | Number of Iterations | Comp. Time (hr:min:sec) | Comp. Time/Iteration (sec) |
|---|---|---|---|
| Popov # 1 | 41 | 01:45:11 | 154 |
| Popov # 2 | 62 | 03:06:59 | 181 |
| Popov # 3 | 79 | 02:57:17 | 135 |
| Popov # 4 | 73 | 02:36:01 | 129 |

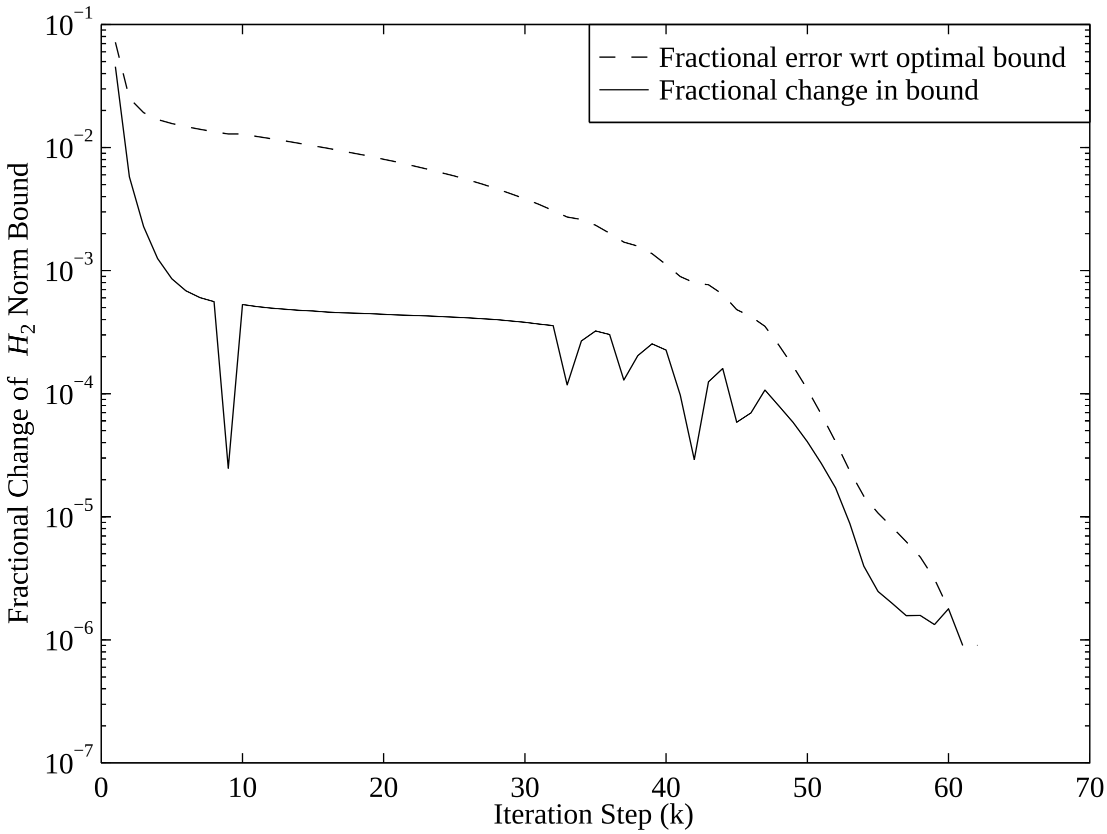



**Figure 4-12:** Fractional change in the $\mathcal{H}_2$/Popov performance bound with respect to a $W$-$K$ optimization pair. This figure is very important because it shows that the choice of a stopping accuracy for the $W$-$K$ iteration is crucial in avoiding a premature termination. It is evident that the sharp spikes may cause the iteration to terminate early and therefore yield a suboptimal $\mathcal{H}_2$/Popov controller design.

suboptimal controller.

## 4.3 Summary

The comparison of the LMI based Popov controller synthesis procedure with its gradient search counterpart introduced in Refs. 24 and 25 shows that the two techniques generate practically identical controllers. However, the $W$-$K$ iteration scheme overcomes the numerical sensitivity that apparently haunted the gradient search approach. Moreover, even though the compensators at each iteration step are not optimal with respect to the multiplier weights or the compensator dynamics, they satisfy the robustness requirements. This is very important because intermediate suboptimal controllers may be used prior to the determination of the local optimal at the end of the $W$-$K$ iteration. On the other hand, the gradient search method of Refs. 24 and 25 requires the use of a loop to iteratively increase the robustness bounds until the required stability bounds are achieved, and therefore the robustness requirements are satisfied solely upon termination of the design process. The capability of using intermediate compensators in the $W$-$K$ iteration scheme favors its use in designing Popov controllers for considerably larger systems.

# Chapter 5

# Conclusion

The contribution of this thesis was to formulate robustness criteria for systems involving linear or nonlinear real parametric uncertainties in terms of standard LMI problems. In addition, the LMI-based robust performance criteria were used to develop an iterative approach to robust controller synthesis that was shown to be computationally attractive.

## 5.1 Summary

Dissipation considerations involving supply rates and storage functions were used to obtain parameter-dependent Lyapunov functions that restrict the magnitude and the time variation of the uncertainties and reduce the conservatism of the robustness criteria. The robust stability criteria were formulated as standard LMI problems, *i.e.*, convex constraints, involving the state space matrices of the system and the Lyapunov function matrices. The robust performance problem was subsequently formulated by introducing a bound on the closed-loop $\mathcal{H}_2$ cost of the system. Due to the coupling of the scaling matrices and the compensator dynamics in both the robustness constraints and the performance metric, the resulting optimization problem, although in matrix inequality form, was nonlinear. This problem was overcome by splitting the robust performance problem into two distinct optimizations each of which comprised a standard LMI problem. The performance metric was optimized with respect to the scaling matrices and the compensator dynamics separately. These two optimizations were denoted the $W$ and $K$ optimizations respectively. The iteration involving the $W$ and $K$ optimizations resulted in the $W$-$K$ iterative design scheme. The final controller guarantees a locally optimal bound on the $\mathcal{H}_2$ cost of the closed-loop system. Since each of the optimizations involve standard LMI problems, they can be solved in polynomial time. The $W$-$K$ iterative design approach was shown to be equivalent to the gradient search approach presented in Refs. 24 and 25. More importantly, it seems that the $W$-$K$ iterative scheme overcomes the numerical problems haunting its gradient search counterpart.

The effectiveness of the $W$-$K$ iterative scheme was evaluated through the design of optimal $\mathcal{H}_2$/Popov controllers for a couple of benchmark robust control systems. The computational efficiency of the $W$-$K$ iteration indicates that expressing the performance criteria in terms of standard LMI problems is computationally advantageous. The intermediate controllers, although suboptimal, fulfill the robustness requirements and the bound on the $\mathcal{H}_2$ norm of the closed-loop system practically converges within a few iteration steps.

## 5.2 Future Work

The iterative approach for designing optimal $\mathcal{H}_2$/Popov controllers that was developed in this thesis was shown to be very effective for some benchmark test systems. However, the benchmark systems that were considered were of low order and may not represent the true performance characteristics of the robust control design technique for real-life systems. However, the inherent advantages of numerical stability and simplicity of the actual design procedure are sufficient reasons for further research on the $W$-$K$ iteration scheme. More specifically it appears that a closed form solution may be obtained that will eliminate the LMIP used to solve for the compensator dynamics in the $K$ optimization. In addition, there are indications that one of the two LMI constraints again in the $K$ optimization may be replaced by a Riccati type equality. It is hoped that this thesis will form a basis for future developments in this LMI-based approach to robust $\mathcal{H}_2$ controller synthesis for systems involving real parametric uncertainties.

# Appendix A

# Matlab Routines

This appendix contains the Matlab routines that comprise the $W$-$K$ iteration scheme for the design of optimal $\mathcal{H}_2$/Popov controllers. Each of the Matlab routines contains comments on its usage and effects. The top level routine, popov.m, must be called with the appropriate values and requires the existence of a Matlab routine named after the specific system at hand, for example spring.m, that may be executed. This routine must result in the definition of the state space matrices $A$, $B_u$, $B_d$, $C_y$, $D_{yd}$, $C_e$, $D_{eu}$, $C_z$, $B_w$, $sys$ is the name of the system at hand, $R_{xx} = C_e^T C_e$, $R_{uu} = D_{eu}^T D_{eu}$, $V_{xx} = B_d B_d^T$, and $V_{yy} = D_{yd} D_{yd}^T$.

The Matlab routines used to generate the above matrices for the mass-spring system and the disk system considered in Chapter 4 are also included.

## popov.m

```
function [P,Ac,Bc,Cc,H,N,M1,M2,bound,valid,delta_t] = popov(sys,stop_accuracy,se
t_H,set_N,M1,M2,P,Ac,Bc,Cc,H,N)

% [P,Ac,Bc,Cc,H,N,M1,M2,bound,valid,delta_t]
%   = popov(sys,stop_accuracy,set_H,set_N,M1,M2,P,Ac,Bc,Cc,H,N)
%
% Function:     popov           This function uses a W-K iterative scheme
%                               for the design of the optimal H2/popov
%                               controller for the system defined by
%                               the matlab script sys.m
%
% Inputs:       sys             System name.
%               stop_accuracy   Accuracy of H2 performance objective bound.
%               set_H           Boolean for setting multiplier scaling H.
%               set_N           Boolean for setting multiplier scaling N.
%  (optional)   M1              Low uncertainty sector bound.
%  (optional)   M2              High uncertainty sector bound.
%  (optional)   P               Inintial Lyapunov matrix value.
%  (optional)   Ac,Bc,Cc        Initial controller state space matrices.
%  (optional)   H,N             Initial multiplier scaling matrices.
%
% Outputs:      P               Optimal Lyapunov matrix.
%               Ac, Bc, Cc      Optimal H2/Popov controller dynamics.
%               H, N            Optimal Popov multiplier scaling matrices.
%               M1,M2           Uncertainty sector bound.
%               bound           Minimum upper bound to H2 performance metric.
%               valid           Boolean specifying validity of the iteration.
%               delta_t         CPU time required for the W-K iteration.
%
% Files         sys_popov.diary Optimization diary file.
% Generated:    sys.mat         System dynamics for system sys.
%               sys_lqg.mat     LQG compensator dynamics for system sys.
%               sys_popov.mat   Optimal H2/Popov controller dynamics.

% Comments      This matlab routine is part of the W-K irterative scheme
%               used for the design of optimal H2/Popov controllers.
%
% Author        Carl Livadas
%               September 1995
%
% Copyright (c) Massachusetts Institute of Technology 1995

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the output in the diary file: sys_popov.diary

format short e
diary_file = concat(sys,'_popov.diary');
delete(diary_file)
diary(diary_file)
clc
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disp('================================================================');
disp('=================    POPOV CONTROLLER DESIGN    ================');
disp('================================================================');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global A Bd Bw Bwtil Bu Bd Ce Cz Cztil Cp Cy Deu Dyd
global Rtil Ruu Rxx Vxx Vyy Vtil
global n nc nd ne nu nw ny nz
global accuracy hide_trace cutoff
global M1 M2 Md Ahat

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Loading or generating the system called 'sys'

if ~exist(sys)==2
  error('Incorrect file name!');
else
  sm = abs(sys);
  sl = length(sm);
  m_file   = concat(sys,'.m');
  mat_file = concat(sys,'.mat');
  if exist(mat_file)==2
    disp(concat('Loading the state-space characteristics for the system: ',sys))
;
    load(sys);
  else
    disp(concat('Generating the state-space characteristics for the system:',sys
));
    eval(sys);
  end;
end;

% Loading or generating the LQG compensator for 'sys'

lqg_file = concat(sys,'_lqg');
if exist(concat(lqg_file,'.mat'))==2
  disp(concat('Loading the LQG compensator for the system: ',sys));
  load(lqg_file);
else
  disp(concat('Generating the LQG compensator for the system: ',sys));
  [P_lqg,Ac_lqg,Bc_lqg,Cc_lqg,cost_lqg] = lqg(A,Bu,Bd,Cy,Dyd,Ce,Deu,sys);
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Setting some of the iteration parameters

accuracy      = stop_accuracy;
accuracy_K    = accuracy/10;    % K-iteration accuracy
accuracy_D    = accuracy/10;    % D-iteration accuracy

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Choosing which iteration to compute.  If only one of the two is
% chosen the loop only gets executed once

perf_K     = 1;                 % Perform the K optimization
perf_D     = 1;                 % Perform the D optimization

% Note: that the iteration will only take place if we perform both
% K and D optimizations.

% Choosing some other iteration parameters

hide_trace = 1;                 % Hiding the LMI solver traces
cutoff     = 0;                 % Allowing some positive definiteness
                                % in the LMIs
debug      = 0;                 % Using debug mode

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% CALCULATING SOME USEFUL MATRICES THAT DO NOT DEPEND ON THE DYNAMIC
% COMPENSATOR SYNTHESIS PROCEDURE

% System Dimensions:

n  = length(A);                 % System size
nu = cols(Bu);
ny = rows(Cy);                  % Measurement size
ne = rows(Ce);                  % Output error size
nw = cols(Bw);
nz = rows(Cz);
nc = n;                         % Controller size
nd = rows(Cz);                  % Uncertainty block size

%  CL uncertainty matrices

Cztil = [Cz zeros(nd,nc)];
Bwtil = [Bw; zeros(nc,nd)];

% Useful pseudo inverses

Cp    = pinv(Cy');

% If not already defined, prompt the user to define the sector bound of the
% uncertainty block

if ~exist('M1') | ~exist('M2')
  M1 = input('Enter the perturbation bound M1: ');
  M2 = input('Enter the perturbation bound M2: ');
end;

% Printing the sector bound matrices

M1
M2
```

```
Md    = inv(M2-M1);
Ahat  = A+Bw*M1*Cz;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Initializing the compensator dynamics, the multipliers and the
% Lyapunov matrix P

if ~exist('P') | ~exist('Ac') | ~exist('Bc') | ~exist('Cc') | ~exist('H') | ~exi
st('N')
  P  = P_lqg;
  Ac = Ac_lqg;
  Bc = Bc_lqg;
  Cc = Cc_lqg;

  if set_H
    H = input('Enter the popov multiplier matrix H: ');
  else
    H = eye(nd);
  end;
  if set_N
    N = input('Enter the popov multiplier matrix N: ');
  else
    N = eye(nd);
  end;

end;

% Retaining the initial Popov multiplier scaling matrices.

H_init = H
N_init = N

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Some useful CL matrices

Cetil_lqg = [Ce Deu*Cc_lqg];
Bdtil_lqg = [Bd;Bc_lqg*Dyd];
Rtil_lqg  = blkdiag(Rxx,Cc_lqg'*Ruu*Cc_lqg);    % Cetil_lqg' * Cetil_lqg
Vtil_lqg  = blkdiag(Vxx,Bc_lqg*Vyy*Bc_lqg');    % Bdtil_lqg  * Bdtil_lqg'

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% INITIALIZING SOME OF THE ITERATION PARAMETERS

bound   = 1e20;
valid_K = 0; valid_D = 0; converged = 0;
stop    = 0;
index   = 0;
bnds_K  = []; bnds_D  = []; bnds      = []; bnds_i    = 0;
time    = []; time_K  = []; dt_K      = []; time_D    = []; dt_D    = [];
Ac_iter = []; Cc_iter = []; Ac_iter_K = []; Cc_iter_K = [];
H_iter  = H;
```

```
N_iter  = N;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Ti = cputime;   % Initial Time

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% STARTING THE D-K ITERATION

stop_accuracy

while ~stop

  Ti_iter = cputime;              % Starting the iteration timer

  index = index + 1;
  temp_bound = bound;

  if perf_K

    disp('================================================================');
    disp('=================          K-ITERATION!         ================');
    disp(sprintf('=================\t\t %d \t\t================',index));
    disp('================================================================');

    Ti_K = cputime;

    accuracy = accuracy_K;
    [P_K,Ac_K,Bc_K,Cc_K,cost_K,bound_K,valid_K] = k_iter(P,Ac,Bc,Cc,H,N,M1,M2);

    if valid_K & (bound_K < temp_bound)
      P      = P_K;
      Ac     = Ac_K;
      Bc     = Bc_K;
      Cc     = Cc_K;
      bound_K
      cost_K

      % Setting the current loop bound

      delta_bound   = bound_K-temp_bound
      temp_bound    = bound_K;

      if debug

        save k_iter P Ac Bc Cc cost_K bound_K

        % The actual cost is given by:

        Bdtil = [Bd;Bc*Dyd];
        Cetil = [Ce Deu*Cc];
        Rtil  = blkdiag(Rxx,Cc'*Ruu*Cc);                % Cetil' * Cetil
        Vtil  = blkdiag(Vxx,Bc*Vyy*Bc');                % Bdtil  * Bdtil'
```

```
        Ptil  = P;
        J_K   = trace((Ptil+Cztil'*(M2-M1)*N*Cztil)*Vtil)

      end

    else

      if ~valid_K
        disp('This K-iteration is not valid! [omitted]');
      else
        disp('This K-iteration step has not optimized the cost function! [omitte
d]');
      end;

      if index>1
        cost_K = cost_iter(index-1);
      else
        cost_K = bound;
      end;

    end;

    % Adjusting the cost

    cost_iter(index) = cost_K;

    % Adjusting the bounds

    bnds_K(:,index) = [temp_bound; bound_K];
    bnds_i          = bnds_i+1;
    bnds(bnds_i)    = temp_bound;

    % Adjusting the times

    curr_time     = cputime-Ti
    dur_K         = cputime-Ti_K
    time(bnds_i)  = curr_time;          % Time of K iteration end
    time_K(index) = curr_time;          % Time of K iteration end
    dt_K(index)   = dur_K;              % Duration of K iteration

    if n<5

      % Saving the compensator dynamics:

      Ac_iter = [Ac_iter; Ac];
      Cc_iter = [Cc_iter; Cc];

    end;

  else

    valid_K = 0;
    bound_K = temp_bound;
```

```
  end;

  if perf_D

    disp('================================================================');
    disp('=================          D-ITERATION!        ================');
    disp(sprintf('=================\t\t %d \t\t================',index));
    disp('================================================================');

    Ti_D = cputime;

    accuracy = accuracy_D;
    [P_D,H_D,N_D,bound_D,valid_D] = d_iter(P,Ac,Bc,Cc,H,N,M1,M2,set_H,set_N);

    % Saving the old bound

    if valid_D & (bound_D < temp_bound)
      P  = P_D;
      H = H_D
      N = N_D
      bound_D

      % Setting the current loop bound

      delta_bound = bound_D-temp_bound
      temp_bound = bound_D;

      if debug

        save d_iter P_D H_D N_D bound_D

        % The actual cost is given by:

        Bdtil = [Bd;Bc*Dyd];
        Cetil = [Ce Deu*Cc];
        Rtil  = blkdiag(Rxx,Cc'*Ruu*Cc);               % Cetil' * Cetil
        Vtil  = blkdiag(Vxx,Bc*Vyy*Bc');               % Bdtil  * Bdtil'
        Ptil  = P;
        J_D   = trace((Ptil+Cztil'*(M2-M1)*N*Cztil)*Vtil)

      end;

    elseif ~valid_D
      disp('This D-iteration is not valid! [omitted]');
    else
      disp('This D-iteration step has not optimized the cost function! [omitted]
');
    end;

    % Adjusting the bounds

    bnds_D(index) = temp_bound;
    bnds_i        = bnds_i+1;
    bnds(bnds_i)  = temp_bound;
```

```
  % Adjusting the times

  curr_time     = cputime-Ti
  dur_D         = cputime-Ti_D
  time(bnds_i)  = curr_time;          % Time of D iteration end
  time_D(index) = curr_time;          % Time of D iteration end
  dt_D(index)   = dur_D;              % Duration of D iteration


  if n<5

    % Saving the multiplier matrices

    H_iter = [H_iter H];
    N_iter = [N_iter N];

  end;

else
  valid_D = 0;
  bound_D = temp_bound;
end;

% Determining whether the iteration must continue or not

stop = 1;

if valid_K | valid_D
  if (abs((bound - temp_bound)/temp_bound) < stop_accuracy)
    disp('This iteration has CONVERGED !!!');
  else
    bound  = temp_bound;
    stop = ~(perf_D & perf_K);
  end;
else
  disp('This iteration can NOT be further optimized!');
end;

% Calculating the time required for this iteration step.

Tf_iter = cputime;
dt_iter = Tf_iter-Ti_iter;

% Displaying the actual time used for a complete W-K iteration step and
% also the elapsed time from the beginning of the W-K iteration

disp(sprintf('\nThe iteration took %d sec. \n',ceil(dt_iter)));
disp(concat('Elapsed time: ',pptime(cputime-Ti), ' [hr:min:sec]'))


% Saves the intermediate results in the file: 'sys'_popov.mat

save(concat(sys,'_popov'))
```

```
  disp(concat('Results saved in the file: ',sys,'_popov.mat'))

end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Displaying the final bound:

disp(sprintf('The final bound is: \t %f',bound))

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Calculating the nominal cost:

Cetil = [Ce Deu*Cc];
Bdtil = [Bd;Bc*Dyd];
Rtil  = blkdiag(Rxx,Cc'*Ruu*Cc);        % Cetil' * Cetil
Vtil  = blkdiag(Vxx,Bc*Vyy*Bc');        % Bdtil  * Bdtil'
Atil  = [A Bu*Cc;Bc*Cy Ac];

Atil_nom = Atil;
Ptil_nom = lyap(Atil_nom',Rtil);
cost_nom = trace(Ptil_nom*Vtil)
cost_lqg
factor_nom = cost_nom/cost_lqg

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Setting a boolean so that I know that the output has converged:

converged = 1;

% Calculating the duration of the whole iteration

Tf = cputime;           % Final Time
delta_t = Tf-Ti         % Duration of iteration in seconds

% Pretty printing the duration of the iteration.

disp(concat('The K-D iteration had a duration of: ',pptime(delta_t), ' [hr:min:s
ec]'))

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Saving the final results in 'sys'_popov.mat

save(concat(sys,'_popov'))
disp(concat('Results saved in the file: ',sys,'_popov.mat'))
diary off
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%            END OF CODE           %%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

## k_iter.m

```
function [P,Ac,Bc,Cc,cost,bound,valid] = k_iter(P,Ac,Bc,Cc,H,N,M1,M2)

% [P,Ac,Bc,Cc,cost,bound,valid] = k_iter(P,Ac,Bc,Cc,H,N,M1,M2)
%
% Overview      Given some Popov multiplier scalings H and N, optimizes
%               the H2 norm bound of the closed loop system with respect to
%               the compensator dynamics.
%
% Inputs        P               Current Lyapunov matrix.
%               Ac, Bc, Cc      Current H2/Popov controller dynamics.
%               H, N            Current Popov multiplier scaling matrices.
%               M1,M2           Uncertainty sector bound.
%
% Outputs       P               Optimal Lyapunov matrix.
%               Ac, Bc, Cc      Optimal H2/Popov controller dynamics.
%               cost            Nominal closed loop H2 norm.
%               bound           Bound on worst-case closed loop H2 norm.
%               valid           boolean specifying whether this optimization
%                               was well behaved and the results satisfied
%                               the constraints.

% Comments      This matlab routine is part of the W-K irterative scheme
%               used for the design of optimal H2/Popov controllers.
%
% Author        Carl Livadas
%               September 1995
%
% Copyright (c) Massachusetts Institute of Technology 1995


% Setting some useful constants and initial values:

global A Ahat Bd Bw Bwtil Bu Bd Ce Cz Cztil Cy Cz Deu Dyd Dyd Dzu Md Ro
global Rtil Ruu Rxx Vxx Vyy Vtil
global cost_lqg
global n nc nd ne nu nw ny nz
global accuracy hide_trace cutoff

% Defining some useful quantities

Ahat  = A+Bw*M1*Cz;
Md    = inv(M2-M1);
temp  = H*Md-N*Cz*Bw;
Ro    = temp + temp';

% The solution parameters are:

mintol  = 1e-10;
tol     = 0;
valid   = 0;

while ~valid
```

```
  [P,R,bound,valid] = elim_theta(P,Ac,Bc,Cc,H,N,M1,M2,Ro,tol);
  if valid

    valid = 0;

    % Splitting up the matrices

    P11 = P(1:n,1:n);
    P21 = P(n+1:n+nc,1:n);
    P12 = P21';
    P22 = P(n+1:n+nc,n+1:n+nc);

    R11 = R(1:n,1:n);
    R21 = R(n+1:n+nc,1:n);
    R12 = R21';
    R22 = R(n+1:n+nc,n+1:n+nc);

    % Setting the Bc matrix to be Cy'

    Bc = Cy';

    % Defining some useful matrices:

    A1 = [A+Bw*M1*Cz    zeros(n);
          Bc*Cy         zeros(n)];

    A2 = [zeros(n)      Bu;
          eye(n)        zeros(n,nu)];

    A3 = [zeros(nc)     eye(nc)];

    C1 = [Ce            zeros(ne,nc);
          zeros(ne,n)   zeros(ne,nc)];

    C2 = [zeros(ne,n)   zeros(ne,nu);
          zeros(ne,n)   Deu];

    C3 = A3;

    Psi = [A1'*P+(A1'*P)'              (H*Cztil+N*Cztil*A1+Bwtil'*P)'  C1';
           H*Cztil+N*Cztil*A1+Bwtil'*P  -Ro                             zeros(nd
,ne+ne)
           C1                          zeros(ne+ne,nd)                 -eye(ne+
ne,ne+ne)];

    U   = [A2'*P                A2'*Cztil'*N    C2'];

    V   = [A3                   zeros(nc,nd)    zeros(nc,ne+ne)];

    % Calculating the closed form solution if Cz * Bu = 0

    if (find((Cz*Bu)~=0) == [])
      disp('K-iteration solving for the controller dynamics using');
      disp('the closed-form solution.');
```

```
      Cc = -inv(Deu'*Deu)*Bu'*R21'*inv(R21*R21');
      Ac = -inv(P12'*P12)*P12'*(Ce'*Ce*R11+Ahat'+P11*Ahat*R11+P12*Bc*Cy*R11+P11*
Bu*Cc*R21+(H*Cz+N*Cz*Ahat+Bw'*P11)'*inv(Ro)*(H*Cz*R11+N*Cz*Ahat*R11+N*Cz*Bu*Cc*R
21+Bw'))*R21'*inv(R21*R21');
      theta = [Ac;Cc];
      M = Psi+V'*theta'*U+U'*theta*V;
      maxeigofM = max(real(eig(M)))
      valid = (maxeigofM < 0);
    end;

    % If the closed form solution was not used or if its result violates the
    % negative definiteness constraint, use the basiclmi.m function built in
    % LMILAB.

    if ~valid
      disp('K-iteration solving for the controller dynamics algebraically...');
      theta = basiclmi(Psi,U,V);
      M = Psi+V'*theta'*U+U'*theta*V;
      maxeigofM = max(real(eig(M)))
      valid = (theta~=[]);
      % the alternative form would be
      % [theta,valid]=calc_theta(Psi,U,V);
    end;
  end;

  if ~valid
    tol = 10*tol+(tol==0)*mintol;
    disp(sprintf('The tolerance for the k_iteration is set to: %f',tol));
  end;
end;

% Performing the nominal CL calculations...

if valid

  Ac   = theta(1:nc,1:nc);
  Cc   = theta(nc+1:nc+nu,1:nc);
  K    = [Ac Bc;Cc zeros(cols(Bu),rows(Cy))];

  Atil = [A Bu*Cc;Bc*Cy Ac];
  Btil = [Bd;Bc*Dyd];
  Ctil = [Ce Deu*Cc];
  G    = [Atil Btil; Ctil zeros(rows(Ce),cols(Bd))];

  % The actual Htwo cost is:

  Qpost    = lyap(Atil,Btil*Btil');
  cost     = trace(Ctil*Qpost*Ctil');

else

  % The LMIs were found infeasible and one must set the
  % return variables
```

```
  Ac    = [];
  Bc    = [];
  Cc    = [];
  cost  = [];
  bound = [];

end;

save after_K
```

## d_iter.m

```
function [P,H,N,bound,valid] = d_iter(P,Ac,Bc,Cc,H,N,M1,M2,set_H,set_N)

% [P,H,N,bound,valid] = d_iter(P,Ac,Bc,Cc,H,N,M1,M2,set_H,set_N)
%
% Overview      Given some H2/Popov controller dynamics, optimizes the H2
%               norm bound of the closed loop system with respect to the
%               Popov multiplier scalings.
%
% Inputs        P               Current Lyapunov matrix.
%               Ac, Bc, Cc      Current H2/Popov controller dynamics.
%               H, N            Current Popov multiplier scaling matrices.
%               M1,M2           Uncertainty sector bound.
%               set_H           Boolean specifying whether the H scale of
%                               the Popov multiplier must be kept constant.
%                               The H scale is kept constant at its current
%                               value.
%               set_N           Boolean specifying whether the N scale of
%                               the Popov multiplier must be kept constant.
%                               The N scale is kept constant at its current
%                               value.
%
% Outputs       P               Optimal Lyapunov matrix.
%               H, N            Optimal Popov multiplier scaling matrices.
%               bound           Bound on worst-case closed loop H2 norm.
%               valid           Boolean specifying whether this optimization
%                               was well behaved and the results satisfied
%                               the constraints.

% Comments      This matlab routine is part of the W-K irterative scheme
%               used for the design of optimal H2/Popov controllers.
%
% Author        Carl Livadas
%               September 1995
%
% Copyright (c) Massachusetts Institute of Technology 1995

% Setting some useful constants and initial values:

global A Bd Bw Bwtil Bu Bd Ce Cz Cztil Cy Deu Dyd
global Rtil Ruu Rxx Vxx Vyy Vtil
global cost_lqg
global n nc nd ne nu nw ny nz
global accuracy hide_trace cutoff

num_iters = 100;
rad_soln  = 1e9;
end_iters = 5;

tol       = 0;
eq_tol    = 1e-10;

use_tol   = (tol~=0);
```

```
use_init  = 0;
disp(sprintf('USE_VARS \t Tol \t Init'));
disp(sprintf('\t \t %d \t %d \n',use_tol,use_init));

options  = [accuracy num_iters rad_soln end_iters hide_trace];

% Set some useful matrices:

P11  = P(1:n,1:n);
P21  = P(n+1:n+nc,1:n);
P22  = P(n+1:n+nc,n+1:n+nc);

Md    = inv(M2-M1);
Ahat  = A+Bw*M1*Cz;

% Initializing the LMI

lmi = [];

% Initializing the counter variables

lmi_num  = 0;  % LMI number
lmi_size = 0;  % LMI size

% Declaring the LMI variables...
% The notation is the following:
%   n_VARNAME = Variable size

n_P11  = n;
n_P22  = nc;
n_P21  = [n_P22 n_P11];
n_P    = n_P11 + n_P22;

n_H   = length(Md);
n_N   = n_H;

[lmi,tagP11]  = addvar(lmi,1,[n_P11 1]);                 % P11
[lmi,tagP21]  = addvar(lmi,2,[n_P22 n_P11]);             % P21
[lmi,tagP22]  = addvar(lmi,1,[n_P22 1]);                 % P22
[lmi,tagH]   = addvar(lmi,1,[ones(n_H,1) zeros(n_H,1)]); % H
[lmi,tagN]   = addvar(lmi,1,[ones(n_N,1) zeros(n_N,1)]); % N

% LMI #1: CL Stability LMI

lmi_num = lmi_num+1;
[lmi,lmi1] = addlmi(lmi);
lmi_size = lmi_size + n_P11 + n_P21(1) + n_H;

% Block (1,1)
lmi=addterm(lmi,[lmi_num 1 1 tagP11],1,Ahat,'s');       % P11*Ahat + Ahat'*P11
lmi=addterm(lmi,[lmi_num 1 1 tagP21],Cy'*Bc',1,'s');     % Cy'*Bc'*P21 + P12*Bc*
Cy
lmi=addterm(lmi,[lmi_num 1 1 0],Ce'*Ce);                 % Ce'*Ce
```

```
% Block (2,1)
lmi=addterm(lmi,[lmi_num 2 1 tagP11],Cc'*Bu',1);         % Cc'*Bu'*P11
lmi=addterm(lmi,[lmi_num 2 1 tagP21],Ac',1);             % Ac'*P21
lmi=addterm(lmi,[lmi_num 2 1 tagP21],1,Ahat);            % P21*Ahat
lmi=addterm(lmi,[lmi_num 2 1 tagP22],1,Bc*Cy);           % P22*Bc*Cy

% Block (2,2)
lmi=addterm(lmi,[lmi_num 2 2 tagP21],1,Bu*Cc,'s');       % P21*Bu*Cc + Cc'*Bu'*P
12
lmi=addterm(lmi,[lmi_num 2 2 tagP22],Ac',1,'s');         % Ac'*P22 + P22*Ac
lmi=addterm(lmi,[lmi_num 2 2 0],Cc'*Deu'*Deu*Cc);        % Cc'*Deu'*Deu*Cc

% Block (3,1)
lmi=addterm(lmi,[lmi_num 3 1 tagH],1,Cz);
lmi=addterm(lmi,[lmi_num 3 1 tagN],1,Cz*Ahat);
lmi=addterm(lmi,[lmi_num 3 1 tagP11],Bw',1);

% Block (3,2)
lmi=addterm(lmi,[lmi_num 3 2 tagN],1,Cz*Bu*Cc);
lmi=addterm(lmi,[lmi_num 3 2 -tagP21],Bw',1);

% Block (3,3)
lmi=addterm(lmi,[lmi_num 3 3 tagH],-1,Md,'s');
lmi=addterm(lmi,[lmi_num 3 3 tagN],1,Cz*Bw,'s');

if use_tol
  lmi=addterm(lmi,[-lmi_num 1 1 0],-tol);
  lmi=addterm(lmi,[-lmi_num 2 2 0],-tol);
  lmi=addterm(lmi,[-lmi_num 3 3 0],-tol);
end;

% LMI #2: P > 0, H > 0, N > 0

lmi_num = lmi_num+1;
[lmi,lmi2] = addlmi(lmi);
lmi_size = lmi_size + n_P + n_H + n_N;

% P > 0
lmi=addterm(lmi,[-lmi_num 1 1 tagP11],1,1);
lmi=addterm(lmi,[-lmi_num 2 1 tagP21],1,1);
lmi=addterm(lmi,[-lmi_num 2 2 tagP22],1,1);

% H > 0
lmi=addterm(lmi,[-lmi_num 3 3 tagH],1,1);

% N > 0
lmi=addterm(lmi,[-lmi_num 4 4 tagN],1,1);
lmi=addterm(lmi,[-lmi_num 4 4 0],eq_tol);

% Ro_til > 0 (this lmi is given by the (3,3) block of the first lmi above

if use_tol
  lmi=addterm(lmi,[-lmi_num 1 1 0],-tol);
  lmi=addterm(lmi,[-lmi_num 2 2 0],-tol);
```

```
  lmi=addterm(lmi,[-lmi_num 3 3 0],-tol);
%  lmi=addterm(lmi,[-lmi_num 4 4 0],-tol);
end;

% Now lets calculate the bound to the actual cost.  The bound is
% given by:
%
%   J = Trace( Bd' * P11 * Bd + Dyd' * Bc' * P22 * Bc * Dyd + Bd' * Cz' * (M2-M1
) * N * Cz * Bd)
%

if set_H
  disp('Letting');
  H
  lmi = setvar(lmi,tagH,H);
end;
if set_N
  disp('Letting');
  N
  lmi = setvar(lmi,tagN,N);
end;

% Calculating the LMI solver weights for the variables P and R.

lmi_name = 'lmi';
c_name = 'cvar';
if ~set_N
  c_formula = 'sum(diag(Bd'' * #tagP11# * Bd)) + sum(diag(Dyd'' * Bc'' * #tagP22
# * Bc * Dyd)) + sum(diag(Bd'' * Cz'' * (M2-M1) * #tagN# * Cz * Bd))';
else
  c_formula = 'sum(diag(Bd'' * #tagP11# * Bd)) + sum(diag(Dyd'' * Bc'' * #tagP22
# * Bc * Dyd))';
end;
makeobj

disp('Calculating P, H and N...')
disp(sprintf('The lmi size is %d.',lmi_size));
disp(sprintf('The variable size is %d.',length(cvar)));

if use_init

  if set_H
    xinit = mat2dec(lmi,P11,P21,P22,N);
  elseif set_N
    xinit = mat2dec(lmi,P11,P21,P22,H);
  else
    xinit = mat2dec(lmi,P11,P21,P22,H,N);
  end;
  [bound, xopt] = linobj(lmi, cvar, options, xinit);
else
  [bound, xopt] = linobj(lmi, cvar, options);
end;

if xopt == []
```

```
  disp('The LMI for the D-iteration is INFEASIBLE!');

  H = [];
  N = [];
  bound = [];
  valid =0;

else
  lmis        = evallmi(lmi,xopt);
  [lhs1 rhs1] = showlmi(lmis,1);
  [lhs2 rhs2] = showlmi(lmis,2);

  P11 = dec2mat(lmi,xopt,tagP11);
  P21 = dec2mat(lmi,xopt,tagP21);
  P12 = P21';
  P22 = dec2mat(lmi,xopt,tagP22);

  if ~set_H
    H   = dec2mat(lmi,xopt,tagH);
  end;
  if ~set_N
    N   = dec2mat(lmi,xopt,tagN);
  end;

  % The final P matrix is:

  P = [P11 P21';P21 P22];

  % Now lets check the results...

  eig_lmi1  = real(eig(lhs1-rhs1));
  eig_lmi2  = real(eig(lhs2-rhs2));

  valid = all([eig_lmi1; eig_lmi2]<cutoff);

  M = lhs1;
  S = M(1:8,1:8)-M(1:8,9)*inv(M(9,9))*M(9,1:8);

  % If LMIs solved incorrectly, print an error message

  if ~valid
    disp('The LMI for the D-iteration is INFEASIBLE!');
  else
    if all([eig_lmi1; eig_lmi2]<0)
      disp('The LMI for the D-iteration is FEASIBLE!');
    else
      disp(sprintf('The LMI for the D-iteration is not well behaved, i.e. INFEAS
IBLE! [%1.4f]',max(real([eig(lhs1-rhs1); eig(lhs2-rhs2)]))));
    end
  end;
end;

save after_D
```

## elim_theta.m

```
function [P,R,bound,valid] = elim_theta(P,Ac,Bc,Cc,H,N,M1,M2,Ro,tol)

% [P,R,bound,valid] = elim_theta(P,Ac,Bc,Cc,H,N,M1,M2,Ro,tol)
%
% Overview
%
% Inputs        P               Current Lyapunov matrix.
%               Ac, Bc, Cc      Current H2/Popov controller dynamics.
%               H, N            Current Popov multiplier scaling matrices.
%               M1,M2           Uncertainty sector bound.
%               Ro              Current value for Ro.
%               tol             Tolerance introduced in order to get a
%                               conservative solution so that the
%                               subsequent calculation of the compensator
%                               dynamics using the basic LMI is feasible.
%
% Outputs       P               Optimal Lyapunov matrix.
%               R               Inverse of the optimal Lyapunov matrix P.
%               bound           Bound on worst-case closed loop H2 norm.
%               valid           Boolean specifying whether this optimization
%                               was well behaved and the results satisfied
%                               the constraints.

% Comments      This matlab routine is part of the W-K irterative scheme
%               used for the design of optimal H2/Popov controllers.
%
% Author        Carl Livadas
%               September 1995
%
% Copyright (c) Massachusetts Institute of Technology 1995

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Setting some useful constants and initial values:

global A Ahat Bd Bw Bwtil Bu Bd Ce Cztil Cy Cz Deu Dyd
global Rtil Ruu Rxx Vxx Vyy Vtil
global cost_lqg
global n nc nd ne nu nw ny nz
global accuracy hide_trace cutoff

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if ~exist('tol') tol = 0; end;
use_tol   = (tol~=0);
use_init  = 0;
disp(sprintf('USE_VARS \t Tol \t Init'));
disp(sprintf('\t \t %d \t %d \n',use_tol,use_init));

options  = [accuracy 0 0 0 hide_trace];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Set some useful matrices:

Cp    = pinv(Cy');
Bc    = Cy';
Md    = inv(M2-M1);
Ahat  = A+Bw*M1*Cz;
Ro    = H*Md+Md*H-N*Cz*Bw-Bw'*Cz'*N;

% Retrieving the old values for the lmi variables:

P11  = P(1:n,1:n);
P21  = P(n+1:n+nc,1:n);
P22  = P(n+1:n+nc,n+1:n+nc);
Pbar = Bc'*(P22-(eye(nc)-Cy'*Cp))*Bc;
R    = inv(P);
R11  = R(1:n,1:n);

% Generating the new P21 and P22 matrices:

P22  = Cp'*Pbar*Cp+eye(nc)-Cy'*Cp;
P21  = diag(Bc)./(diag(Bc)+(diag(Bc)==0))*P21;
R11  = R11+R11*1e-10;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Initializing the LMI P and R solution

lmi = [];

% Initializing the counter variables

lmi_num  = 0;  % LMI number
lmi_size = 0;  % LMI size

% Declaring the LMI variables...
% The notation is the following:
%   n_VARNAME = Variable size

n_P11  = n;
n_P22  = nc;
n_Pbar = rows(Cp);
n_P21  = [n_P22 n_P11];
n_P    = n_P11 + n_P22;

n_R11  = n;
n_R22  = nc;
n_R21  = [n_R22 n_R11];
n_R    = n_R11 + n_R22;

[lmi,tagP11]  = addvar(lmi,1,[n_P11 1]);               % P11
[lmi,tagP21]  = addvar(lmi,2,[n_P22 n_P11]);           % P21
[lmi,tagPbar] = addvar(lmi,1,[n_Pbar 1]);              % Pbar
[lmi,tagR11]  = addvar(lmi,1,[n_R11 1]);               % R11
```

```
% LMI #1: Wu'*Psi*Wu < 0

lmi_num = lmi_num+1;
[lmi,lmi1] = addlmi(lmi);
lmi_size = lmi_size + n + length(H) + ne;

% Row 1
lmi=addterm(lmi,[lmi_num 1 1 tagR11],1,Ahat','s');
lmi=addterm(lmi,[lmi_num 1 1 0],-Bu*inv(Deu'*Deu)*Bu');

% Row 2
lmi=addterm(lmi,[lmi_num 2 1 tagR11],H*Cz,1);
lmi=addterm(lmi,[lmi_num 2 1 tagR11],N*Cz*Ahat,1);
lmi=addterm(lmi,[lmi_num 2 1 0],Bw'-N*Cz*Bu*inv(Deu'*Deu)*Bu');

lmi=addterm(lmi,[lmi_num 2 2 0],-Ro-N*Cz*Bu*inv(Deu'*Deu)*Bu'*Cz'*N);

% Row 3
lmi=addterm(lmi,[lmi_num 3 1 tagR11],Ce,1);

lmi=addterm(lmi,[lmi_num 3 3 0],-1);

if use_tol
  lmi=addterm(lmi,[-lmi_num 1 1 0],-tol);
  lmi=addterm(lmi,[-lmi_num 2 2 0],-tol);
  lmi=addterm(lmi,[-lmi_num 3 3 0],-tol);
end;

% LMI #2: Wv'*Psi*Wv < 0

lmi_num = lmi_num+1;
[lmi,lmi2] = addlmi(lmi);
lmi_size = lmi_size + n + length(H);

% Block (1,1)
lmi=addterm(lmi,[lmi_num 1 1 tagP11],1,Ahat,'s');
lmi=addterm(lmi,[lmi_num 1 1 tagP21],Cy'*Cy,1,'s');
lmi=addterm(lmi,[lmi_num 1 1 0],Ce'*Ce);

% Block (2,1)
lmi=addterm(lmi,[lmi_num 2 1 tagP11],Bw',1);
lmi=addterm(lmi,[lmi_num 2 1 0],H*Cz+N*Cz*Ahat);

% Block (2,2)
lmi=addterm(lmi,[lmi_num 2 2 0],-Ro);

if use_tol
  lmi=addterm(lmi,[-lmi_num 1 1 0],-tol);
  lmi=addterm(lmi,[-lmi_num 2 2 0],-tol);
end;

% LMI #3: P > inv(R)
```

```
lmi_num = lmi_num+1;
[lmi,posdef_lmi] = addlmi(lmi);
lmi_size = lmi_size + n + nc + n;

lmi=addterm(lmi,[-lmi_num 1 1 tagP11],1,1);
lmi=addterm(lmi,[-lmi_num 2 1 tagP21],1,1);
lmi=addterm(lmi,[-lmi_num 2 2 tagPbar],Cp',Cp);
lmi=addterm(lmi,[-lmi_num 2 2 0],eye(n_P22)-Cy'*Cp);
lmi=addterm(lmi,[-lmi_num 3 1 0],1);
lmi=addterm(lmi,[-lmi_num 3 3 tagR11],1,1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Now lets calculate the bound to the actual cost.  The bound is
% given by:
%
%   J = Trace( Bd' * P11 * Bd + Dyd' * Pbar * Dyd + Bd' * Cz' * (M2-M1) * N * Cz
 * Bd)
%

% Calculating the LMI solver weights for the variables P and R.

lmi_name  = 'lmi';
c_name    = 'cvar';
c_formula = 'sum(diag(Bd'' * #tagP11# * Bd)) + sum(diag(Dyd'' * #tagPbar# * Dyd)
) + sum(diag(Bd'' * Cz'' * (M2-M1) * N * Cz * Bd))';
makeobj

disp('Calculating P and R...')
disp(sprintf('The lmi size is %d.',lmi_size));
disp(sprintf('The variable size is %d.',length(cvar)));

if use_init

  xinit       = mat2dec(lmi,P11,P21,Pbar,R11);
  lmis        = evallmi(lmi,xinit);
  [lhs1 rhs1] = showlmi(lmis,1);
  [lhs2 rhs2] = showlmi(lmis,2);
  [lhs3 rhs3] = showlmi(lmis,3);

  save before_K

  [bound, xopt] = linobj(lmi, cvar, options, xinit);
else
  [bound, xopt] = linobj(lmi, cvar, options);
%  [tmin, xfeas] = feasp(lmi, options);
end;

if xopt == []
  disp('The LMI for the K-iteration is INFEASIBLE!');
  valid =0;
else
  lmis        = evallmi(lmi,xopt);
  [lhs1 rhs1] = showlmi(lmis,1);
```

```
  [lhs2 rhs2] = showlmi(lmis,2);
  [lhs3 rhs3] = showlmi(lmis,3);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  P11  = dec2mat(lmi,xopt,tagP11);
  P21  = dec2mat(lmi,xopt,tagP21);
  P12  = P21';
  Pbar = dec2mat(lmi,xopt,tagPbar);
  P22  = Cp' * Pbar * Cp + eye(n_P22)-Cy'*Cp;

  R11  = dec2mat(lmi,xopt,tagR11);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  % Now lets calculate the full P21 block:

  M        = sqrtm(P11-inv(R11));
  Upartial = (sqrtm(inv(P22)) * P21 * inv(M))';
  [u,s,v]  = svd(Upartial);
  U        = u*v';
  P21old   = P21;
  P21      = sqrtm(P22)*U'*M;
  P12      = P21';

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  % The final P and R matrix values are:

  P = [P11 P21';P21 P22];
  R11old = R11;
  R = inv(P);
  R11 = R(1:n,1:n);
  R21 = R(n+1:n+nc,1:n);
  R12 = R21';
  R22 = R(n+1:n+nc,n+1:n+nc);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

  % Now lets check the results...

  eig_lmi1  = real(eig(lhs1-rhs1));
  eig_lmi2  = real(eig(lhs2-rhs2));
  eig_lmi3  = real(eig(lhs3-rhs3));
  valid = all([eig_lmi1; eig_lmi2; eig_lmi3] < cutoff);

  % If LMIs solved incorrectly, print an error message

  if ~valid
    disp('The LMI for the K-iteration is INFEASIBLE!');
  else
    if (max(real([eig(lhs1-rhs1); eig(lhs2-rhs2); eig(lhs3-rhs3)])) < 0)
      disp('The LMI for the K-iteration is FEASIBLE!');
    else
```

```
      disp(sprintf('The LMI for the K-iteration is not well behaved, i.e. INFEAS
IBLE! [%1.4f]',max(real([eig(lhs1-rhs1); eig(lhs2-rhs2); eig(lhs3-rhs3)]))));
    end;
  end;
end;
```

## calc_theta.m

```
function [theta,valid] = calc_theta(Psi,U,V)

% [theta,valid] = calc_theta(Psi,U,V)
%
% Overview      Solves the basic LMI problem of the form
%
%               Psi + U' theta V + V' theta' U < = 0
%
% Inputs        Psi, U and V
%
% Outputs       theta
%               valid           boolean specifying whether this optimization
%                               was well behaved and the results satisfied
%                               the constraints.

% Comments      This matlab routine is part of the W-K irterative scheme
%               used for the design of optimal H2/Popov controllers.
%
% Author        Carl Livadas
%               September 1995
%
% Copyright (c) Massachusetts Institute of Technology 1995

global accuracy hide_trace cutoff

accuracy = 1e-6;
options = [accuracy 0 0 0 1];

theta_lmi = [];

% Initializing the counter variables

lmi_num  = 0;  % LMI number
lmi_size = 0;  % Size of the LMI to be solved

% Declaring the LMI variable Theta

size_theta  = [cols(U') rows(V)];

[theta_lmi,tagtheta] = addvar(theta_lmi,2,size_theta); % Theta

% Recall this is a feasibility problem:

lmi_num = lmi_num+1;
[theta_lmi,lmi1] = addlmi(theta_lmi);
lmi_size = lmi_size + length(Psi);

theta_lmi=addterm(theta_lmi,[lmi_num 1 1 0],Psi);
theta_lmi=addterm(theta_lmi,[lmi_num 1 1 tagtheta],U',V,'s');

disp('Calculating the controller parameters Theta')
disp(sprintf('The lmi size is %d.',lmi_size));
```

```
[tmin, xopt] = feasp(theta_lmi, options);

tmin

if xopt == [] | tmin > 0
  theta = [];
  valid = 0;
  disp('The controller parameter LMI is INFEASIBLE!');
else
  lmis      = evallmi(theta_lmi,xopt);
  [lhs rhs] = showlmi(lmis,1);

  theta = dec2mat(theta_lmi,xopt,tagtheta);

  % Now lets check the results...

  valid  = all(real(eig(lhs-rhs))<cutoff);

  if ~valid
    disp('The controller parameter LMI is INFEASIBLE!');
  else
    if all(real(eig(lhs-rhs))<0)
      disp('The controller parameter LMI is FEASIBLE!');
    else
      disp(sprintf('The controller parameter LMI solution is not well behaved, i
.e. INFEASIBLE! [%1.4f]',max(real(eig(lhs-rhs)))));
    end;
%    max(real(eig(lhs-rhs)))
  end;
end;
```

## spring.m

```
% Comments      This matlab routine is part of the W-K irterative scheme
%               used for the design of optimal H2/Popov controllers.
%
% Author        Carl Livadas
%               September 1995
%
% Copyright (c) Massachusetts Institute of Technology 1995

clear all;
format short e

accuracy = 1e-6;

% This is a matlab script that generates the state space matrices for
% the first example of Jon How's thesis:

k   = 1;

A   = [0 0 1 0; 0 0 0 1; -k k 0 0; k -k 0 0];
Bu  = [0 0 1 0]';
Bd  = [0 0; 0 0; 0 0; 1 0];

% Measurements:

Cy  = [0 1 0 0];
Dyd = [0 sqrt(0.001)];  % This is what is used in Jon's thesis

% Performance Outputs:

Ce  = [0 1 0 0; 0 0 0 0];
Deu = [0; sqrt(0.001)];

% Perturbation matrices:

Cz = [1 -1 0 0];
Bw = [0 0 -1 1]';

% Some weighting matrices

Rxx = Ce'*Ce;
Ruu = Deu'*Deu;
Vxx = Bd*Bd';
Vyy = Dyd*Dyd';

% Setting the plot axes

x_min = -1;
x_max = 1.5;
y_min = 10^(-1);
y_max = 10^(2);

% Defining the system name and saving the info
```

```
sys = 'spring';
save(sys);
disp(concat('Results saved in the file: ',sys,'.mat'));

% Calculating the lqg controller

lqg(A,Bu,Bd,Cy,Dyd,Ce,Deu,sys);
```

## disk2.m

```
% Comments      This matlab routine is part of the W-K irterative scheme
%               used for the design of optimal H2/Popov controllers.
%
% Author        Carl Livadas
%               September 1995
%
% Copyright (c) Massachusetts Institute of Technology 1995

clear all;
format short e

accuracy = 1e-6;

% This is a matlab script that generates the state space matrices for
% the second example of Jon How's thesis:

k   = 1;
m   = 1;
d   = 0.01;
rho = 0.005;

J   = m*diag([0.5,1,1,1]);
T   = [1 -1 0 0;-1 2 -1 0;0 -1 2 -1;0 0 -1 1];
K   = k*T;
D   = d*T;

A   = [zeros(4) eye(4);-inv(J)*K -inv(J)*D];
Bu  = [0 0 0 0 0 1/m 0 0]';
Bd  = [0 0; 0 0; 0 0; 0 0; 0 0; 0 0; 0 0; 1/m 0];

% Measurements:

Cy  = [0 0 0 1 0 0 0 0];
Dyd = [0 sqrt(rho)];

% Performance Outputs:

Ce  = [0 0 0 1 0 0 0 0.1
       0 0 0 0 0 0 0 0];
Deu = [0;sqrt(rho)];

% Perturbation matrices:

Cz = [1 -1 0 0 0 0 0 0;0 0 1 -1 0 0 0 0];
Bw = [0 0 0 0 -2 1 0 0;0 0 0 0 0 0 -1 1]';

% The perturbation bounds are

p = input('Enter the uncertainty magnitude ');

if p ==[]
  disp('Using the default uncertainty magnitude.');
```

```
  p = 0.024;
end;

M1 = -p*eye(2)
M2 = p*eye(2)

% Some weighting matrices

Rxx = Ce'*Ce;
Ruu = Deu'*Deu;
Vxx = Bd*Bd';
Vyy = Dyd*Dyd';

% Setting the plot axes

x_min = -0.15;
x_max = 0.2;
y_min = 0.5;
y_max = 2;

% Defining the system name and saving the info

sys = 'disk2';
save(sys);
disp(concat('Results saved in the file: ',sys,'.mat'));

% Calculating the lqg controller

lqg(A,Bu,Bd,Cy,Dyd,Ce,Deu,sys);
```

# References


[1] Aizerman, M. A. and Gantmacher, F. R., *Absolute Stability of Regulator Systems*, Holden-Day, Inc., San Francisco, 1964.

[2] Athans, M., "The Matrix Minimum Principle," *Information and Control*, vol. 11, pp. 592–606, 1968.

[3] Balas, G. J., Young, P. M., and Doyle, J. C., "$\mu$-Based Control Design as Applied to Large Space Structure: Control Design for the Minimast Facility," *Tech. Rep., NASA CSI/GI Final Report*, August 1992.

[4] Barmish, R. B., "New Tools for Robustness Analysis," *Proc. IEEE Conference on Decision and Control*, pp. 1–6, December 1988.

[5] Bernstein, D. S. and Haddad, W. M., "The Optimal Projection Equations with Petersen-Hollot Bounds: Robust stability and Performance via Fixed-Order Dynamic Compensation for Systems with Structured Real-Valued Parameter Uncertainty," *IEEE Transactions on Automatic Control*, vol. AC–33, pp. 578–582, 1988.

[6] Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V., *Linear Matrix Inequalities in System and Control Theory*, Prentice Hall, 1994.

[7] Braatz, R. D., Young, P. M., Doyle, J. C., and Morari, M., "Computational Complexity of $\mu$ Calculation," *Proc. of American Control Conference*, pp. 1682–1683, June 1993.

[8] Brockett, R. W. and Willems, J. C., "Frequency Domain Stability Criteria – Parts I and II," *IEEE Transactions on Automatic Control*, vol. AC–10, no. 3 and 4, pp. 255–261 and 407–412, 1965.

[9] 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.

[10] Chiang, R. Y and Safonov, M. G., "Real $K_m$-Synthesis via Generalized Popov Multipliers," *Proc. of American Control Conference*, June 1992.

[11] Doyle, J. C., "Analysis of Feedback Systems with Structured Uncertainties," *IEE Proceedings*, vol. 129, no. 6, pp. 242–250, November 1982.

[12] Doyle, J. C., "Structured Uncertainty in Control System Design," *Proc. IEEE Conference on Decision and Control*, pp. 260–265, 1985.

[13] Doyle, J. C., Francis, B. A., and Tannenbaum, A. R., *Feedback Control Theory*, Macmillan Publishing Company, New York, 1992.

[14] Doyle, J. C., Glover, K., Khargonekar, P. P., and Francis, B. A., "State-Space Solutions to Standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ Control Problems," *IEEE Transactions on Automatic Control*, vol. AC–34, no. 8, pp. 831–847, August 1989.

[15] Fan, M. K. H., Tits, A. L., and Doyle, J. C., "Robustness in the Presence of Fixed Parametric Uncertainty and Unmodelled Dynamics," *IEEE Transactions on Automatic Control*, vol. AC–36, no. 1, pp. 25–38, January 1991.

[16] Francis, B. A., *A Course in $\mathcal{H}_\infty$ Control Theory*, Springer-Verlag, New York, 1987.

[17] Gahinet, P. and Nemirovskii, A., "General-Purpose LMI Solvers with Benchmarks," *Proc. IEEE Conference on Decision and Control*, pp. 3162–3165, 1993.

[18] Gahinet, P. and Nemirovskii, A., "The Projective Method for Solving Linear Matrix Inequalities," *Proc. American Control Conference*, pp. 840–844, 1994.

[19] Gahinet, P., Nemirovskii, A., Laub, A. J., and Chilali, M., "The LMI Control Toolbox," *Proc. IEEE Conference on Decision and Control*, 1994.

[20] 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.

[21] Haddad, W. M. and Bernstein, D. S., "Robust Stabilization with Positive Real Uncertainty: Beyond the Small Gain Theorem," *Systems and Control Letters*, vol. 17, pp. 191–208, 1991.

[22] Haddad, W. M. and Bernstein, D. S., "Explicit Construction of Quadratic Lyapunov Functions for the Small Gain, Positivity, Circle, and Popov Theorems and their Application to Robust Stability Parts I and II," *International Journal on Robust and Nonlinear Control*, 1992.

[23] Hall, S. R. and How, J. P., "Mixed $\mathcal{H}_2/\mu$ Performance Bounds using Dissipation Theory," *Proc. Conference on Decision and Control*, December 1993.

[24] How, J. P., "Robust Control Design with Real Parameter Uncertainty using Absolute Stability Theory," PhD thesis, Massachusetts Institute of Technology, January 1993.

[25] 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.

[26] How, J. P. and Hall, S. R., "Connections between Absolute Stability Theory and Bounds for the Structured Singular Value," *IEEE Transactions on Automatic Control*, 1992.

[27] How, J. P. and Hall, S. R., "Connections between the Popov Stability Criterion and Bounds for Real Parameter Uncertainty," *Proc. American Control Conference*, pp. 1084–1089, June 1993.

[28] Khargonekar, P. P., Petersen, I. R., and Zhou, K., "Robust Stabilization of Uncertain Linear Systems: Quadratic Stabilizability and $\mathcal{H}_\infty$ Control Theory," *IEEE Transactions on Automatic Control*, vol. AC–35, pp. 356–361, 1990.

[29] Leitmann, G., "Guaranteed Asymptotic Stability for some Linear Systems with Bounded Uncertainties," *Transactions of the ASME*, vol. 101, pp. 212–216, September 1979.

[30] Morton, B. G. and McAfoos, R. M., "A Mu-Test for Robustness Analysis of Real Parameter Variation Problem," *Proc. American Control Conference*, pp. 135–138, May 1985.

[31] MuSyn, Inc., *$\mu$-Analysis and Synthesis Toolbox: User's Guide*, The Math Works, Inc., Natick, MA, 1991.

[32] Narendra, K. S. and Taylor, J. H., *Frequency Domain Criteria for Absolute Stability*, Academic Press, New York, 1973.

[33] Packard, A. and Doyle, J. C., "Quadratic Stability with Real and Complex Perturbations," *IEEE Transactions on Automatic Control*, vol. AC–35, pp. 198–201, 1990.

[34] Petersen, I. R. and Hollot, C. V., "A Riccati Equation Approach to the Stabilization of Uncertain Systems," *Systems and Control Letters*, vol. 8, pp. 351–357, 1987.

[35] Pinzoni, S. and Willems, J. C., "The Dissipation Inequality for Systems Described by High-order Differential Equations," *Proc. of IEEE Conference on Decision and Control*, pp. 1439–1440, December 1992.

[36] Popov, V. M., "On Absolute Stability of Nonlinear Automatic Control Systems," *Automatika I Telemekhanika*, vol. 22, no. 8, pp. 961–979, 1961.

[37] Safonov, M. G., *Stability and Robustness of Multivariable Feedback Systems*, M.I.T. Press, Cambridge, MA, 1980.

[38] Safonov, M. G. and Chiang, R. Y, "Real/Complex $K_m$-Synthesis without Curve Fitting," in *Control and Dynamic Systems,* Academic Press, New York, 1993.

[39] Safonov, M. G. and Lee, P. H., "A Multiplier Method for Computing Real Multivariable Stability Margin," *Proc.,* submitted to *the IFAC World Congress,* Sydney, Australia, July 1993.

[40] Safonov, M. G. and Wyetizner, G., "Computer-Aided Stability Analysis Renders Popov Criterion Obsolete," *IEEE Transactions on Automatic Control*, vol. AC–32, pp. 1128–1131, 1987.

[41] Sparks, A. G. and Bernstein, D. S., "Reliable State Space Upper Bounds for the Peak Structured Singular Value," *Proc. American Control Conference*, pp. 2419–2423, June 1995.

[42] 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, September 1993.

[43] Tahk, M. and Speyer, J., "Modeling of Parameter Variations and Asymptotic LQG Synthesis," *IEEE Transactions on Automatic Control*, pp. 793–801, September 1987.

[44] Tekawy, J. A., Safonov, M. G., and Chiang, R. Y., "Convexity Property of One-Sided Multivariable Stability Margin," *IEEE Transactions on Automatic Control*, vol. AC–37, pp. 496–498, 1992.

[45] Thompson, P. M., "Classical/$\mathcal{H}_2$ Solution for a Robust Control Design Benchmark Problem," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 18, no. 1, pp. 160–169, Jan.–Feb. 1995.

[46] Wie, B. and Bernstein, D. S., "Benchmark Problems for Robust Control Design," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 15, no. 5, pp. 1057–1059, Sept.–Oct. 1992.

[47] Willems, J. C., "Dissipative Dynamical Systems Part I: General Theory," *Archive Rational Mechanics Analysis*, vol. 45, pp. 321–351, 1972.

[48] Willems, J. C., "Dissipative Dynamical Systems Part II: Linear Systems with Quadratic Supply Rates," *Archive Rational Mechanics Analysis*, vol. 45, pp. 352–393, 1972.

[49] Young, P. M. and Doyle, J. C., "Computation of $\mu$ with Real and Complex Uncertainties," *Proc. IEEE Conference on Decision and Control*, pp. 1230–1235, December 1990.

[50] Young, P. M. and Doyle, J. C., "Properties of the Mixed-$\mu$ Problem and its Bounds," *IEEE Transactions on Automatic Control*, October 1992.

[51] Zames, G., "On the Input-Output Stability of Time-Varying Nonlinear Systems Parts I and II," *IEEE Transactions of Automatic Control*, vol. AC–11, no. 2 and 3, pp. 228–238 and 465–476, 1966.