The Distance to Uncontrollability via Linear Matrix Inequalities

Steven J. Boyce

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Mathematics

Lizette Zietsman, Chair
Anderson H. Norton
Jeffrey T. Borggaard
Martin V. Day

December 3, 2010
Blacksburg, Virginia

Keywords: Optimization, Systems, Control, LMI, SDP, Uncontrollability

The Distance to Uncontrollability via Linear Matrix Inequalities

Steven J. Boyce

(ABSTRACT)

The distance to uncontrollability of a controllable linear system is a measure of the degree of perturbation a system can undergo and remain controllable. The definition of the distance to uncontrollability leads to a non-convex optimization problem in two variables. In 2000 Gu proposed the first polynomial time algorithm to compute this distance. This algorithm relies heavily on efficient eigenvalue solvers.

In this work we examine two alternative algorithms that result in linear matrix inequalities. For the first algorithm, proposed by Ebihara et. al., a semidefinite programming problem is derived via the Kalman-Yakubovich-Popov (KYP) lemma. The dual formulation is also considered and leads to rank conditions for exactness verification of the approximation. For the second algorithm, by Dumitrescu, Şicleru and Ştefan, a semidefinite programming problem is derived using a sum-of-squares relaxation of an associated matrix-polynomial and the associated Gram matrix parameterization. In both cases the optimization problems are solved using primal-dual-interior point methods that retain positive semidefiniteness at each iteration.

Numerical results are presented to compare the three algorithms for a number of benchmark examples. In addition, we also consider a system that results from a finite element discretization of the one-dimensional advection-diffusion equation. Here our objective is to test these algorithms for larger problems that originate in PDE-control.

Acknowledgments

I wish to thank the many people who made this thesis possible. First and foremost, it would be nearly impossible to overstate my gratitude to my advisor, Dr. Lizette Zietsman. Her support, enthusiasm, and willingness to teach me about the array of the topics that I encountered kept me motivated and eager to learn more. I would also like to thank Dr. Borggaard for the finite-element code used in the control application. I appreciate the input from graduate students Adam Bowman and Kasie Farlow, who helped with preparation for presenting portions of this thesis. Lastly, I would like to acknowledge the encouragement and patience demonstrated by my wife, Kendra Atkins-Boyce.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The motivation for the problem, its history, and recent developments are explored.

1.1 Distance to Uncontrollability

Consider the first order continuous-time system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{1.1}$$

where $A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{p\times n}$, and $D \in \mathbb{R}^{p\times m}$.

Two fundamental concepts in control theory are those of controllability and observability.

Definition 1.1.1 *The pair $[A\ B]$ is said to be* ***controllable*** *if starting at any initial state $x(0) = x_0$, the system* (1.1) *can be driven to any final state $x_1 = x(t_1)$ in some finite time $t_1 > 0$ choosing the continuous input $u(t)$, for $0 \leq t \leq t_1$, appropriately.*

Definition 1.1.2 *The pair $[C\ D]$ is said to be* ***observable*** *if there exists $t_1 > 0$ such that the initial state $x(0)$ can be uniquely determined from the knowledge of $u(t)$ and $y(t)$, for $0 \leq t \leq t_1$.*

There are several equivalent mathematical characterizations of (1.1), see for example, [33]. We list three characterizations each of controllability and observability.

C1. The system (1.1) is controllable if and only if the controllability matrix has full rank. That is,

$$\operatorname{rank}\left(B, AB,\ \ldots,\ A^{n-1}B\right) = n. \quad (1.2)$$

C2. The system (1.1) is controllable if and only if,

$$\operatorname{rank}\left(B, A - \lambda I\right) = n, \quad (1.3)$$

for each eigenvalue λ of A.

C3. The system (1.1) is controllable if and only if there exists a matrix K such that set of eigenvalues of

$$A - BK \quad (1.4)$$

and the set of eigenvalues of A are mutually exclusive.

O1. The system (1.1) is observable if and only if the observability matrix has full rank. That is,

$$\operatorname{rank}\left((C, CA,\ \ldots, CA^{n-1})^T\right) = n. \quad (1.5)$$

O2. The system (1.1) is observable if and only if

$$\operatorname{rank}\left((\lambda I - A, C)^T\right) = n, \quad (1.6)$$

for each eigenvalue λ of A. See [20].

O3. The system (1.1) is observable if and only if there exists a matrix L such that the set of eigenvalues of

$$A + LC \quad (1.7)$$

and the set of eigenvalues of A are mutually exclusive.

In [33] Paige describe three direct approaches to verify the controllability and observability of a system:

(i) Form the matrices in (1.2) and (1.5) and compute their ranks.

(ii) Compute all the eigenvalues of A and compute ranks of the matrices in (1.3) and (1.6).

(iii) Compute all the eigenvalues of A, perform the comparison from (1.4) and (1.7) for a random matrix K.

Each of the methods has its own challenges when executed using finite precision arithmetic. For example, numerical error can play a role in determining the rank of a matrix (as in the first two conditions) and in determining the equality of two eigenvalues (as in the last condition).

Apart from the difficulty in verifying exact controllability of a matrix pair, Example 1.1.1 demonstrates how a small perturbation can affect the exact controllability of a matrix.

Example 1.1.1 *(Eising, 1984) The Matrix pair* $[A \;\; B]$ *is controllable where*

$$A = \begin{bmatrix} -1 & -1 & \dots & \dots & -1 & -1 \\ 1 & \ddots & & & & -1 \\ & 1 & \ddots & & & -1 \\ & & \ddots & \ddots & & \vdots \\ 0 & & & \ddots & \ddots & \vdots \\ & & & & 1 & 1 \end{bmatrix}, \; B = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{bmatrix}.$$

If the last row is perturbed by 2^{1-n}, where $A \in \mathbb{R}^{n \times n}$, the system becomes uncontrollable.

Table 1.1: Examples of perturbations resulting in uncontrollability

n	20	40	60	80	100
2^{1-n}	1.9×10^{-6}	1.8×10^{-12}	1.7×10^{-18}	1.7×10^{-24}	1.6×10^{-30}.

□

Clearly, using computer arithmetic and numerical algorithms, can lead to a controllable system becoming numerically uncontrollable despite a high degree of precision.

Considering that a matrix has full-rank if its singular values are non-zero, it follows that an uncontrollable matrix is arbitrary close to some controllable system. Singular values are well-conditioned with respect to perturbations in the system, and in fact using the singular value decomposition, we can find how far a matrix is from the nearest matrix of a given lower rank, [22].

Since an uncontrollable system is arbitrarily close to some controllable system, but a controllable system may or may not be close to an uncontrollable system, the measure of the distance from a controllable system to an uncontrollable system is of far greater use than knowing that a system is exactly controllable. This metric is formally defined by Paige.

Definition 1.1.3 *Distance to uncontrollability (Paige [33])*

Assume [A B] is controllable. Then the distance to uncontrollability, τ, is defined as

$$\tau(A, B) = \min \|(\delta A, \delta B)\| \text{ such that the system defined by } [A + \delta A \;\; B + \delta B] \text{ is uncontrollable.}$$

Here either the 2-norm or Frobenius norm is used.

Paige [33] outlined the motivation for this definition by theorizing that **if there were a numerically stable algorithm** that could compute the exact distance to uncontrollability τ for a nearby system:

$$\dot{x} = (A + \delta_A)x + (B + \delta_B)u,$$
$$\text{where } \|\delta_A\| \leq CTOL\|A\|;\ \|\delta_B\| \leq CTOL\|B\|.$$

Then if

$$\tau(A, B) > (ATOL + CTOL)\|A\| + (BTOL + CTOL)\|B\|$$

a designer would have confidence that the system was controllable. Here $ATOL$ and $BTOL$ are the precision for matrices A and B, δ_A and δ_B are perturbations, and $CTOL$ is the tolerance for the computation. A large distance to uncontrollability on a nearby system leads to a large distance to uncontrollability on the system of interest.

Eising, see [16], use the relationship between rank and singular value decomposition to write the distance to uncontrollability as

$$\tau(A, B) = \min_{\lambda \in \mathbb{C}} \sigma_n([A - \lambda I, B]), \tag{1.8}$$

where $\sigma_n(G)$ denotes the n^{th} largest singular value of $G \in \mathbb{C}^{n \times (n+m)}$.

The following example illustrates some of the difficulties in approximating the distance to uncontrollability via optimization, such as multiple local minima.

Example 1.1.2 *(Ebhiara, [15])*

Consider the problem to compute the distance to uncontrollability of the controllable pair $[A\ B]$ where

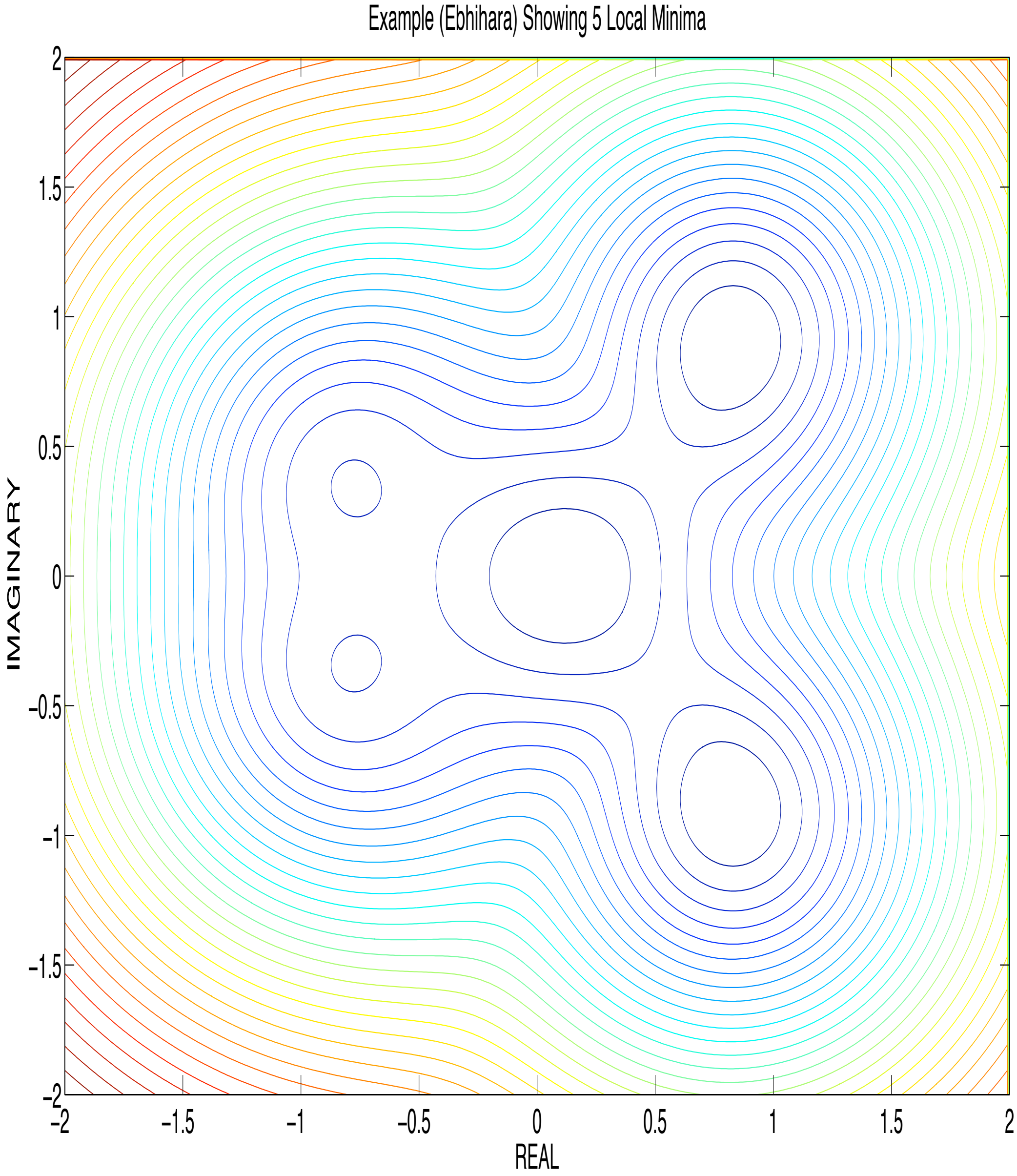
$$A = \begin{bmatrix} 0.0738 & 0.0407 & 0.9088 & -0.4485 & 0.8856 \\ 0.4370 & -0.2746 & -0.0579 & -0.2505 & -0.1580 \\ 0.1166 & 0.0472 & -0.5416 & -0.3650 & 0.3760 \\ -0.8070 & 0.9024 & -0.0012 & 0.1459 & -0.4057 \\ -0.3297 & -0.7345 & 0.2330 & 0.7524 & 0.6038 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 0.7135 & -0.2087 & -0.0804 & -0.1726 & -0.8912 \\ -0.6224 & -0.0742 & -0.1083 & -0.3099 & -0.6416 \\ 0.1434 & -0.7873 & 0.4139 & 0.2392 & 0.0859 \\ -0.1091 & 0.1916 & 0.3438 & 0.3059 & -0.1606 \\ 0.0150 & -0.1913 & -0.0885 & 0.2258 & 0.0718 \end{bmatrix}.$$

Figure 1.1 shows the contour plot of $\log_{10}(\sigma_{\min}([A - \lambda I B]))$ where the horizontal and vertical axes show the real and imaginary parts of λ. It appears that there are five local minima.

Figure 1.1: Contour plot of $\log_{10}(\sigma_{\min}([A - \lambda IB]))$



1.2 Computing the Distance to Uncontrollability

In this section, we present a brief history of the development of algorithms prior to the first polynomial-time algorithm by Gu [18]. We also give a short summary of Gu's algorithm and different linear matrix inequality (LMI) approaches. These algorithms are discussed in more detail in Sections 3, 4, and 5, respectively.

1.2.1 History of Calculating the Distance to Uncontrollability

Eising's formulation (1.8) is a difficult problem to solve. It is a nonsmooth global optimization problem in two real variables, α and β, where $\lambda = \alpha + i\beta$. In addition, $\sigma_n([A - \lambda I, B])$ is nonconvex. While some authors [4, 33] proposed heuristic algorithms to compute perturbation matrices to approximate τ using Definition 1.1.3, this method could not accurately compute the distance to uncontrollability itself, and was unreliable as there are examples in which the method fail to show a nearly uncontrollable pair [8]. Using Definition 1.1.3 and (1.3), the problem was recast as the quest for λ for which

$$\text{Rank}\ ([A - \lambda I\ \ B]) = n \quad \text{for all } \lambda \in \mathbb{C}. \tag{1.9}$$

Since λ varies over all complex numbers it is unclear which values of λ for which singular values might be computed to verify the full rank condition of the matrix $[A - \lambda I\ B]$.

Byers [8] suggested a reliable but impractical "brute force" method in which the singular values are evaluated only at mesh points dependent upon the norm of A and B and some tolerance ϵ. Modifications to reduce the size of the mesh were explored in [8].

Boley [4] and Miminis [28] developed more efficient algorithms for this optimization, but it is a non-convex minimization problem, so the number of local minima varies from problem to problem. Such minimization requires expensive singular value decompositions of both $h(\lambda) = \sigma_{\min}([A - \lambda I\ B])$ and its derivative [8]. Algorithms prior to the development of Gu's scheme that do find the global minimum were prohibitively expensive, requiring computing time inversely proportional to τ and thus prohibitive for nearly uncontrollable systems [18].

1.2.2 Gu's Scheme

Gu algebraically manipulated Eising's formula into a generalized eigenvalue problem of size $\mathcal{O}(n^2)$ using a verification scheme of order $\mathcal{O}(n^6)$, see [18]. Starting with a given possible minimum singular value, σ, a test is constructed using a search for a real eigenvalue. If one exists, then the global minimum singular value was smaller than σ, and one uses the bisection method to reduce, ending with a range of values for τ. While Gu's verification scheme has remained the central idea of this algorithm, improvements in the iterative process have been made. The algorithm was improved upon first by Burke, Lewis, and Overton, who replaced the bisection step with a trisection [7] which allows you to appraximate the distance to any desired accuracy. Mengi improved upon the verification scheme using Sylvester Equation solvers to reduce complexity to average $\mathcal{O}(n^4)$ and in the worst case $\mathcal{O}(n^5)$, see [19].

1.2.3 LMI Approach

In their discussion of improvements to Gu's methods, Burke et al. [7, p 8] noted

> "No other polynomial-time algorithm for estimating (the distance to uncontrollability) within a constant factor seems to be known; in particular, it does not seem to be known whether Gu's test could be replaced by an LMI (linear matrix inequality)-based test."

This is a natural question because convex optimization problems with LMI constraints can be solved very efficiently using recent interior-point methods.

The original problem is approximated by a semidefinite programming (SDP) problem with LMI constraints. This is achieved by utilizing the Kalman-Yakubovich-Popov (KYP) lemma [3], (D, G)-scaling, see [25], and Lagrange duality theory, [36]. Primal-dual interior point methods are then used to solve the SDP.

The computation involves $2n^2 + n$ scalar variables, while the size of the LMI is $5n$, see [13] The computational complexity to solve the SDP is represented by $\mathcal{O}(K^2R^{2.5} + R^{3.5})$, where K denotes the number of scalar variables involved and R the size of the underlying LMI, see [13, 38]. Thus this method is of order $\mathcal{O}(n^{6.5})$.

1.2.4 Sum-of-Square Approach

Dumitrescu, Sicleru, and Sefan, see [12], independently derived another way to use semidefinite programming to find the controllability radius. Their approach is to exploit the relationship between singular values and eigenvalues; that is, the minimum singular value of a matrix M is the square root of the minimum eigenvalue of the matrix MM^*. In this way, they recast (1.8) as the computation of the square root of the smallest eigenvalue of

$$P(\lambda) = |\lambda|^2 I = \lambda A^* - \overline{\lambda} A + AA^* + BB^* \quad \text{for} \quad \lambda \in \mathbb{C}. \tag{1.10}$$

The problem, (1.8), can then be expressed as an LMI, searching for

$$\tau_0 = \max_{\tau \geq 0} \tau \text{ such that } P(\lambda) - \tau I \geq 0 \text{ for all } \quad \lambda \in \mathbb{C} \tag{1.11}$$

so that $\tau = \sqrt{\tau_0}$. For ease of implementation, they compute an estimate using a sum-of-squares relaxation, so that the problem can be converted into SDP by parametrizing with the Gram matrix, see [34] for more detail. They estimate their approach using SeDuMi [38] to have a complexity of $\mathcal{O}(n^6)$, see [12] .

1.3 Main Findings

This thesis is devoted to algorithms that compute the distance to uncontrollability via SDP with LMI constraints and the numerical study of these algorithms which consider systems in literature. These systems are relatively small, of size $n = 10$ or smaller. The results of Gu are considered to be accurate, and we compared the results of the two new methods with those published in [19]. Of the 39 matrix pairs tested, the results of Dumetriescu agreed with Gu's method on all but two pairs, while the results of Ebihara's method agreed with Gu's method on only 20 pairs.

Our main interest is the efficiency and accuracy of large systems generated by PDE based optimal control parameters. For a first study we consider the 1-D advection-diffusion equation

$$z_t(t,x) = -v \cdot \nabla z(t,x) + \mu\Delta^2 z(t,x) + b(x)u(t), \quad 0 < x < 1,$$

where μ represents the diffusion coefficient and v represents the advection velocity.

Using a finite (quadratic) element method with various mesh sizes and (single) observer locations, the distance to observability for the system was calculated using each of the three methods to determine the location of the optimal sensor location. This optimal sensor location should be placed where the distance to unobservability is largest. The method of Ebihara quickly was eliminated as a practical choice, for due to lack of convergence in its call to SeDuMi it often "timed out" without yielding a solution. This resulted in slower speed and inaccurate results for this method. The sum-of-squares method performed much better, with results much closer to those of Mengi, although the sum-of-squares method produced, in general, a slightly larger distance to unobservability and was less consistent in its suggested placement location. In terms of performance, the sum-of-squares method outperformed that of Mengi when the mesh size was smaller, but the advantage was reversed as soon as the mesh size grew into the twenties. Thus for smaller sized systems, Dumitrescu's method works well, but for large systems Mengi's method is preferable. The reliability and correspondence of the results from the two methods is encouraging, and perhaps there is a place for both methods in systems control analysis.

Chapter 2

Background on Semidefinite Programming

Brief History of SDP

There exist many important applications of semidefinite programming in control theory, combinatorial optimization and mechanical and electrical engineering, see for example [10] and [41]. The use of linear matrix inequalities in control theory began with Lyapunov toward the end of the nineteenth century, who developed and solved what is now called the Lyapunov inequality condition: a differential equation $x'(t) - Ax(t)$ is stable if and only if there is is a positive definite matrix P such that $A^TP + PA \succeq 0$. Among others, Lur'e used LMIs to solve practical control problems by hand in the 1950s. Yakubavich and colleagues developed graphical criteria for linear matrix inequalities that were later found related to Algebraic Ricatti Equations. In the early 1980s researchers observed that the LMIs from control theory can be formulated as convex optimization problems. The development of interior point methods for linear or quadratic programming, beginning with Karmakar, led to Nesterov and Nemirovskii's development of interior-point methods for more general convex problems involving LMIs, including SDP. Algorithms and software for solving such problems continue to be developed and improved. For more details on the development of SDP in control theory, see [5].

The main focus of this study is algorithms where Equation (1.8)

$$\tau(A, B) = \min_{\lambda \in \mathbb{C}} \sigma_n([A - \lambda I, B])$$

is approximated by a semidefinite programming (SDP) problem. In this chapter we provide a brief background on primal and dual SDP problems. In addition, we describe primal-dual interior point algorithms used to solve the SDPs. These algorithms are implemented in the software package SeDuMi, see [39].

2.1 Problem Description

A *semidefinite programming* (SDP) problem is of the form

$$\begin{aligned}&\text{minimize } c^t x\\ &\text{subject to } F(x) \geq 0, \text{where } \ F(x) = F_0 + \sum_{i=1}^{m} x_i F_i.\end{aligned} \tag{2.1}$$

The vector, $c \in \mathbb{R}^m$, as well as the $m+1$ symmetric matrices, $F_i \in \mathbb{R}^{n\times n}$, with $i = 0,\ 1,\ \ldots,\ m$ are known. Here $c^t x$ is called the *objective function* and the inequality $F(x) \geq 0$, represents the *constraints*. This inequality constraint, $F(x) \geq 0$ implies that $F(x)$ is positive semidefinite, that is, $z^T F(x) z \geq 0$ for all $z \in \mathbb{R}^n$.

The inequality $F(x) \geq 0$ is called a *linear matrix inequality* (LMI) and can be rewritten using the Löwner partial ordering, $\succeq$, on symmetric matrices $\mathcal{S}^n$. That is, if A and B are $n \times n$ symmetric matrices we write $A \succeq B$ if and only if $A - B$ is positive semidefinite. This ordering is reflexive, antisymmetric and transitive. The inequality constraint $F(x) \geq 0$ is equivalent to $F(x) \succeq 0_n$ where 0_n denotes the $n \times n$ zero matrix. Thus (2.1) can be written as

$$\begin{aligned}&\text{minimize } c^t x\\ &\text{subject to } F(x) \succeq 0_n, \ \text{where } F(x) = F_0 + \sum_{i=1}^{m} x_i F_i.\end{aligned} \tag{2.2}$$

Note that SDPs are convex optimization problems with a linear objective function and linear matrix inequality constraints and include many well-known convex optimization problems as special cases. Consider for example, the linear program (LP)

$$\begin{aligned}&\min c^t x\\ &\text{such that } Ax + b \geq 0,\end{aligned} \tag{2.3}$$

where $c \in \mathbb{R}^m$, $A \in \mathbb{R}^{n\times m}$, and $b \in \mathbb{R}^n$ are known. The inequality, $Ax + b \geq 0$ denotes *component* wise inequality. Note that a vector $z \geq 0$ if and only if the matrix $\operatorname{diag}(z)$ (the diagonal matrix with the components of the vector z on the diagonal) is positive semidefinite. The matrix problem (2.3) can thus be expressed as a semidefinite program where

$$F(x) = \operatorname{diag}(Ax + b) = \operatorname{diag}(b) + \sum_{i=1}^{m} x_i \operatorname{diag}(a_i),$$

where a_i denote the i-th column in the matrix $A = [a_1,\ \ldots,\ a_m]$.

The semidefinite program can also be thought of as an extension of an LP problem where the component wise inequalities are replaced by matrix inequalities. The inequality $F(x) \succeq 0_n$ is equivalent to an infinite set of linear constraints on x since $z^T F(x) z \geq 0$ for all $z \in \mathbb{R}^n$.

2.2 SDP Optimization Using Duality

2.2.1 Introduction to Lagrange Multipliers and Duality

The theory of semidefinite programming problems is very similar to that of linear programming problems but there are important differences. For example, duality results are weaker for semidefinite programming problems than for linear programming problems. In this section we provide the necessary background to describe the primal-dual interior point methods that are used to solve these type of problems.

We start this discussion by reviewing the standard linear problem with both equality and inequality constraints. In standard form

$$\begin{aligned} \text{minimize } & f_0(x) \\ \text{such that } & f_i(x) \leq 0, i = 1, \ \ldots, \ m, \\ & h_i(x) = 0, i = 1, \ \ldots, \ p, \end{aligned} \tag{2.4}$$

where $x \in \mathbb{R}^n$. Denote the feasible set by $\mathcal{D} \subseteq \mathbb{R}^n$ and the optimal value of the objective function by q.

The *Lagrangian* function, $\mathcal{L} \colon \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ is defined as

$$\mathcal{L}(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \tag{2.5}$$

where $\lambda \in \mathbb{R}^m$ denote the Lagrange multiplier vector associated with the inequality constraints, and $\nu \in \mathbb{R}^p$ denote the Lagrange multiplier vector associated with the equality constraints. The Lagrangian function is a sum of the objective function and weighted constraint functions.

The *Lagrangian dual* function, $g \colon \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ is defined as

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} \mathcal{L}(x, \lambda, \nu). \tag{2.6}$$

If we add an additional constraint, $\lambda \geq 0$, then $g(\lambda, \nu) \leq q$ for all $x \in \mathcal{D}$. One can see this because for any $\hat{x} \in \mathcal{D}$, if $\lambda \geq 0$, then $\sum_{i=1}^{m} \lambda_i f_i(\hat{x}) \leq 0$ and $\sum_{i=1}^{p} \nu_i h_i(\hat{x}) = 0$, so that $f_0(\hat{x}) \geq \mathcal{L}(\hat{x}, \lambda, \nu)$. From the definition of the dual function follows $\mathcal{L}(\hat{x}, \lambda, \nu) \geq \inf_{x \in \mathcal{D}} \mathcal{L}(x, \lambda, \nu) = g(\lambda, \nu)$, consequently $q \geq g(\lambda, \nu)$.

The *Lagrange Dual Problem* is defined as

$$\max_{\lambda \geq 0} g(\lambda, \nu). \tag{2.7}$$

In other words, the Lagrangian Dual Problem finds the best lower bound on q obtained from g. This is a convex optimization problem, and thus has a single maximum, denoted by d.

The relationship between the optimal values d and q are characterized by either *weak duality* or *strong duality.* Weak duality means $d \leq q$, and always holds. Strong duality, $d = q$, does not hold in general, but holds under certain constraint qualifications.

2.2.2 Lagrange Multipliers and Duality for the Semidefinite Program

The semidefinite programming problem (2.1) is not a standard linear problem and the strategy described above can not be applied directly. We follow the global theory for constrained optimization described by Luenberger in [24].

Consider the primal problem defined in (2.1):

$$\begin{aligned} &\text{minimize } c^t x \\ &\text{subject to } F(x) \geq 0, \text{ where } F(x) = F_0 + \sum_{i=1}^{m} x_i F_i. \end{aligned}$$

The inequality constraint $F(x) \geq 0$, implies that $F(x) \in \mathcal{S}^n$ is positive semidefinite. This inequality can be rewritten using the Löwner partial ordering, $\succeq$, on symmetric matrices $\mathcal{S}^n$. That is, if A and B are $n \times n$ symmetric matrices we write $A \succeq B$ if and only if $A - B$ is positive semidefinite. This ordering is reflexive, antisymmetric and transitive. The inequality constraint $F(x) \geq 0$ is equivalent to $F(x) \succeq 0_n$ where 0_n denotes the $n \times n$ zero matrix. Thus (2.1) can be written as

$$\begin{aligned} &\text{minimize } c^t x \\ &\text{subject to } F(x) \succeq 0_n, \text{ where } F(x) = F_0 + \sum_{i=1}^{m} x_i F_i. \end{aligned} \tag{2.8}$$

The function $f(x) = c^T x$ is a mapping from $\mathbb{R}^m$ into $\mathbb{R}$ and F maps $\mathbb{R}^m$ into $\mathcal{S}^n$.

Due to the inequality in (2.2) it is necessary to define the positive cone in the vector space $\mathcal{S}^n$. The cone of positive semidefinite matrices is denoted by $\mathcal{S}^n_+$ and defined as matrices

$$\mathcal{S}^n_+ = \{A \in \mathcal{S}^n \colon x^t A x \geq 0 \;\text{ for all }\; x \in \mathbb{R}^n\}$$

or equivalently

$$\mathcal{S}^n_+ = \{A \in \mathcal{S}^n \colon A \succeq 0_n\}.$$

The positive cone $\mathcal{S}^n_+$ has the following properties that enable its usefulness, as listed in [37]:

1. $\mathcal{S}_+^n$ is closed, the limit of a sequence of positive semidefinite matrices is positive semidefinite,
2. $\mathcal{S}_+^n$ is full dimensional with a nonempty interior,
3. $\mathcal{S}_+^n$ is self-dual, which means that the optimization of the *dual problem* is over the same positive semidefinite cone.

Remarks: The first two conditions justify using iterative methods (restricted to $\mathcal{S}_+^n$) to perform the optimization. The fact that the cone is self-dual enables the optimization of both problems to be performed simultaneously.

The *Lagrangian* of (2.2) is defined as

$$\mathcal{L}(x, Z) = c^T x - \langle F(x), Z \rangle \tag{2.9}$$

where Z is in the dual of $\mathcal{S}_+^n$, that is $Z \in (\mathcal{S}_+^n)^*$. In functional notation $\langle F(x), Z \rangle$ can be expressed as $ZF(x)$.

The set of symmetric matrices S^n is isomorphic to $\mathbb{R}^{\frac{n(n+1)}{2}}$. This can be seen by writing the entries of the matrix as a vector with the columns stacked end to end. Since the matrix is symmetric, only the entries in the upper triangular portion is considered, and consequently every $n \times n$ symmetric matrix, say A, is isomorphic to a vector $\mathbf{vec}(A) \in \mathbb{R}^{\frac{n(n+1)}{2}}$. This isomorphism also implies that $(\mathcal{S}^n)^* = \mathcal{S}^n$. We now use this isomorphism to express $\langle F(x), Z \rangle$ in terms of the trace of the product, $F(x)Z$. We write $\mathrm{Tr}\left(F(x)Z\right)$.

For any A and any $B \in S^n$ we have

$$\langle A, B \rangle = \mathbf{vec}(B)^T \mathbf{vec}(A) = \mathrm{Tr}\left(BA\right).$$

This inner product introduces the Frobenius-norm

$$\|A\|^2 = \langle A, A \rangle = \mathrm{Tr}\left(AA^T\right) = \sum_{i,j} A_{ij}^2.$$

If both A and B are symmetric,

$$\mathrm{Tr}\left(AB\right) = \mathrm{Tr}\left(BA\right).$$

Using the isomorphism described above, we can rewrite the term $\langle F(x), Z \rangle$ in the Lagrangian as

$$\begin{aligned} \langle F(x), Z \rangle &= \langle F_0, Z \rangle + \sum_{i=1}^{m} x_i \langle F_i, Z \rangle \\ &= \mathrm{Tr}\left(F_0 Z\right) + \sum_{i=1}^{m} x_i \mathrm{Tr}\left(F_i Z\right). \end{aligned}$$

The Lagrangian (2.9) now takes the form

$$\mathcal{L}(x, Z) = c^T x - \mathrm{Tr}\,(F_0 Z) - \sum_{i=1}^{m} x_i \mathrm{Tr}\,(F_i Z) \tag{2.10}$$

and the *Lagrangian dual functional* is defined on the positive cone in the dual space $(\mathcal{S}^n)^*$ as

$$\psi(Z) = \inf_{x \in \mathbb{R}^m} \left[c^T x - \langle F(x), Z \rangle \right] = \inf_{x \in \mathbb{R}^m} \left[c^T x - \mathrm{Tr}\,(F_0 Z) - \sum_{i=1}^{m} x_i \mathrm{Tr}\,(F_i Z) \right].$$

In general, ψ is not finite throughout the positive cone but the region where it is, is convex.

As for the linear case one can show that the dual functional always serves as a lower bound to the value of the primal problem provided that $\psi(Z)$ is finite for some $Z \in \mathcal{S}^n_+$. See the discussion in [24, page 225].

The *Lagrange Dual Semidefinite Problem* is then defined as

$$\max_{Z \in \mathcal{S}^n_+} \psi(Z). \tag{2.11}$$

Here the variable is the matrix $Z \in \mathcal{S}^n_+$.

The Wolfe dual [42], is more convenient for computations and can be stated as follows

$$\begin{aligned} \max_{x,\ Z} \ & \mathcal{L}(x, Z) \\ \text{subject to } & \nabla_x \mathcal{L}(x, Z) = 0, \ Z \succeq 0_n. \end{aligned} \tag{2.12}$$

Using the notation in (2.10) the dual problem is equivalent to

$$\begin{aligned} \max \ & -\mathrm{Tr}\,(F_0 Z) \\ \text{subject to } & \mathrm{Tr}\,(F_i Z) = c_i, \quad i = 1,\ 2,\ \ldots,\ m, \\ & Z \succeq 0_n. \end{aligned} \tag{2.13}$$

where x is a solution to (2.1).

The dual semidefinite program yields a bound on the primal semidefinite program and vice versa. To see this, assume that Z is dual feasible and x is primal feasible. Then, since Z and $F(x)$ are symmetric, positive semidefinite,

$$c^T x + \mathrm{Tr}\,(F_0 Z) = \sum_{i=1}^{m} \mathrm{Tr}\,(F_i Z)\, x_i + \mathrm{Tr}\,(F_0 Z) = \mathrm{Tr}\,(F(x) Z) \geq 0. \tag{2.14}$$

Consequently,

$$-\mathrm{Tr}\,(F_0 Z) \leq c^T x. \tag{2.15}$$

The *duality gap* η is defined as

$$\eta = c^T x + \mathrm{Tr}\,(F_0 Z) = \mathrm{Tr}\,(F(x)Z)$$

which is a linear function of x and Z.

Let p^* denote the optimal value of the primal problem (2.1) and d^* the optimal value of the dual problem (2.13). Then (2.15) implies that $d^* \leq p^*$ which is referred to as *weak duality.* The case where $d^* = p^*$ is called *strong duality.*

In linear programming one has strong duality if both the primal and dual problems are feasible. However in semidefinite programming (and conic programming in general) this is not always the case. There are examples in [37] and [41] in which either the primal or the dual problem has a finite optimal value but the other does not, as well as cases in which both the primal and dual have finite optimal values, but there is a finite duality gap (weak duality) since the problem is not strictly feasible.

Nesterov and Nemirovsky, [30] used duality in convex analysis to prove conditions for strong duality; $d^* = p^*$.

Theorem 2.2.1 *The duality gap is zero, $d^* = p^*$, if either of the following conditions holds.*

1. *The primal problem (2.1) is strictly feasible, that is, there exists an x with $F(x) \succeq 0_n$.*
2. *The dual problem (2.13) is strictly feasible, that is, there exists a Z with $Z \in \mathcal{S}_+^n$, $\mathrm{Tr}\,(F_i Z) = c_i,\ i = 1,\ 2,\ \ldots,\ m$.*

If both conditions hold, the primal and dual optimal sets are nonempty.

Note that the respective primal and dual optimal sets are

$$P_{opt} = \Big\{x \ : \ F(x) \succeq 0_n \ \text{ and } \ c^T x = p^*\Big\}$$

and

$$D_{opt} = \{Z \ : \ Z \succeq 0_n,\ \mathrm{Tr}\,(F_i Z) = c_i, \ \ i = 1,\ 2,\ \ldots,\ m \ \text{ and } \ -\mathrm{Tr}\,(F_0 Z) = d^*\}\,.$$

If the two optimal sets are nonempty, there exist x and Z such that

$$p^* = c^T x = -\mathrm{Tr}\,(F_0 Z) = d^*.$$

From (2.14) it then follows that $\mathrm{Tr}\,(F(x)Z) = 0$. Since both $F(x) \succeq 0_n$ and $Z \succeq 0_n$, we conclude that $F(x)Z = 0_n$. This condition is called the *complementary slackness* condition.

If we assume that both the primal (2.1) and dual (2.13) problems are strictly feasible, then by Theorem 2.2.1 the optimal values in both problems are attained and the solutions are characterized by the optimality conditions

$$\begin{aligned} &\mathrm{Tr}\,(F_i Z) = c_i, \quad i = 1,\ \ldots,\ m, \\ &F(x) \succeq 0_n \ \text{ and } \ Z \succeq 0_n, \\ &F(x)Z = 0_n. \end{aligned} \tag{2.16}$$

This is an example of the Karush-Kuhn-Tucker (KKT) necessary conditions for an optimal solution, see [6].

2.2.3 Primal-Dual Method of Solving SDP

Interior-point methods for linear programming problems became well known with the ellipsoid algorithm of Khacijan in 1979. This approach allowed a polynomial bound on the worst case iteration count but the actual application of this method was disappointing. In 1984 Karmarkar introduced an algorithm with an improved complexity bound. This was followed in 1988 by Nesterov and Nemirovsky, see [29], that showed that these interior-point methods can be generalized to all convex optimization problems. The critical element was the knowledge of a *barrier function* with a property called self-concordance in which case Newton methods are very efficient in minimizing the function over its domain. Linear matrix inequalities are convex optimization problems for which computable self-concordant barrier functions are known which makes interior-point methods particularly applicable in these cases.

One can think of f being a self-concordant function if $f \in C^3(dom(f))$, this domain is closed and convex and for every $x \in dom(f)$ and $h \in \mathbb{R}^n$ one has

$$|D^3 f(x)[h,h,h]| \leq 2 \left(D^2 f(x)[h,h]\right)^{3/2}.$$

Here $D^k f(x)[h_1,\ \ldots h_k]$ denotes the k-th differential of a smooth function f taken at a point x along the directions $h_1,\ \ldots, h_k$. This concept of self-concordance is discussed in detail in [21].

In [40] it is stated that the most promising methods for semidefinite programming solve the primal and dual problems simultaneously. The basic idea is as follows:

The *primal-dual central path* is followed, which generates a sequence of primal and dual feasible points, $x^{(k)}$ and $Z^{(k)}$ where k denotes the iteration number. The point $x^{(k)}$ is suboptimal and provides an upper bound $p^* \leq c^T x^{(k)}$ and $Z^{(k)}$ proves the lower bound $p^* \geq -\mathrm{Tr}\left(F_0 Z^{(k)}\right)$. The suboptimality of the current points are bounded in terms of the duality gap $\eta^{(k)}$,

$$c^T x^{(k)} - p^* \leq \eta^{(k)} = c^T x^{(k)} + \mathrm{Tr}\left(F_0 Z^{(k)}\right).$$

This leads to the development of a stopping criterion for which the duality gap $c^T x^{(k)} + \mathrm{Tr}\left(F_0 Z^{(k)}\right) \leq \epsilon$, i.e., when the duality gap between the primal and dual solutions is nearly zero. One can think of this as solving the primal-dual optimization problem where the duality gap is minimized over all primal-dual feasible points. This is again a semidefinite program in x and Z.

$$\begin{aligned} \text{minimize} \quad & c^T x + \mathrm{Tr}\,(F_0 Z) \\ \text{subject to} \quad & F(x) \succeq 0_n, \\ & Z \succeq 0_n, \\ & F(x)Z = 0_n. \end{aligned} \tag{2.17}$$

Rather than solving the primal and dual problems separately, the dual information in each step, $Z^{(k)}$, is used to find a good update for the primal variable $x^{(k)}$ and vice versa.

The central path is an arc of strictly feasible points that is parametrized by a scalar μ. That is, we assume that there exists an x with $F(x) \succ 0_n,\ Z = Z^T \succ 0_n$ with $\mathrm{Tr}\,(F_i Z) = c_i$ for $i = 1, \ldots, m$. The primal-dual central path using the duality gap parametrization, see [41, Section 4.5], is defined as the set of solutions $x(\mu)$ and $Z(\mu)$ of the nonlinear equations

$$\begin{aligned} & \mathrm{Tr}\,(F_i Z(\mu)) = c_i, \quad i = 1, \ldots, m, \\ & F(x(\mu)) \succeq 0_n \ \text{ and } \ Z(\mu) \succeq 0_n, \\ & F(x(\mu))Z(\mu) = \mu I_n. \end{aligned} \tag{2.18}$$

The pair $x^{(k)}$ and $Z^{(k)}$ converges to a primal and dual optimal pair as $\mu \to 0$, see [40]. Here path following algorithms restrict the iterates to a neighborhood of the central path. Thus by generating sequences $x^{(k)}(\mu)$ and $Z^{(k)}(\mu)$ that follow the central path for decreasing values of μ, one obtain convergence to an optimal solution. This involves computing primal and dual *search directions*, $\delta x, \delta F$ and δZ. These directions can be interpreted as Newton search directions for solving the set of nonlinear equations (2.18). The search directions are computed by linearizing about a current iterate and solving a set a linear equations. There are several choices of linearization which determines the different types of interior-point methods.

To simplify the notation for the discussion that follows, let F, Z and x denote the current iterate, $F(x^{(k)}),\ Z^{(k)}$ and $x^{(k)}$.

The first approach involves either primal or dual scaling (in which after linearization either δF or δZ is eliminated). A primal-dual symmetric scaling, in which $FZ = \mu I$ is linearized as $F\delta Z + \delta F Z = \mu I - FZ$, was used for linear problems, but its solution is not generally symmetric. One method attributed to Alizadeh, Haeberly, and Overton of resolving this issue was to first write $FZ = \mu I$ as $FZ + ZF = 2\mu I$ and linearize this equation as

$$F\delta Z + \delta F Z + Z\delta F + \delta Z F = 2\mu I - FZ - ZF$$

which results in symmetric δF and δZ. See [40] for the detail. Other researchers, including Nesterov and Todd, and Sturm and Zhang, defined new matrices R and Λ such that

$$R^t Z R = \Lambda^{1/2} = R^t F^{-1} R, \text{ where } \Lambda \text{ is diagonal with the same eigenvalues as } FZ.$$

The search directions are then obtained by solving the system

$$\begin{aligned} \operatorname{Tr}(F_i \delta Z) &= 0, \quad i = 1, \ \ldots, \ m \\ -RR^t \delta Z R R^t + \sum_{i=}^{m} \delta x_i F_i &= F - \mu X^{-1}. \end{aligned} \tag{2.19}$$

Practical primal-dual algorithms retain the positive-semidefinite characteristics at each iteration [31], but most implementations work with an infeasible starting point and infeasible iterations. Suggestions for choosing the starting point are described in [6, 31].

Practical algorithms typically use corrector steps that compensate for the linearization error made during the scaling step described above. After the step direction is computed, a maximum *step length* and recentering constant are computed so that the iterates do not get too close to the boundary of the positive semidefinite cone.

2.3 SeDuMi Implementation

For our calculations we used open source software SeDuMi, written in Matlab and C for optimization over symmetric cones and can perform optimization with linear, quadratic, or semidefinite constraints, see [38]. Some of relevant advantages to SeDuMi are

- high numerical accuracy and robustness,
- takes full advantage of sparsity, leading to significant speed benefits,
- has a theoretically proven $O(\sqrt{n}\log(1/\epsilon)$ worst-case iteration bound.

SeDuMi deals with semidefinite constraints by converting $n \times n$ symmetric matrices into vectors of length $(n(n+1)/2)$. Checking positive semidefiniteness of constraints is executed very efficiently using the `eigK` command.

There are several options in SeDuMi which can be specified the accuracy. These settings affect the number of iterations necessary to obtain the desired accuracy. The first option is the step-length. The default option is to perform the step with a second-order corrector step (as opposed to the longest possible step). The centering of the step can be changed, as well as the weight of the corrector. The error, denoted by ϵ, takes the default value 10^{-9}. SeDuMi quits when it finds a solution that violates feasibility and optimality requirements by no more than epsilon.

In preliminary numerical experimentation, we experimented with different values for the options described above. We discovered quickly in the initial phase with sample data from Mengi that the longest step algorithm resulted in poor performance and the maximum number of iterations was exceeded. We also changed the step parameters, changing the centering and amount of weight placed on the "corrector". In general, this did not seem to affect the result very much in terms of accuracy or the number of iterations (unless the parameters were changed so drastically that it resulted in the longest-step algorithm). Altering the default value of ϵ did not have an effect on the result, in general, although for well-conditioned matrices a larger ϵ gave the correct result in fewer iterations, while for ill-conditioned matrices a smaller ϵ resulted in values closer to the correct value.

In subsequent numerical experiments we used the default values in SeDuMi. Note that, for a given problem in which the solution is known and the default settings give an incorrect result, it may be possible to alter the centering parameters so that the correct solution is obtained. While decreasing the value of ϵ results in a more accurate result, but it also increases the number of iterations required. We left ϵ as the default for more direct comparison between the different methods.

Chapter 3

Gu's Bisection Algorithm

This chapter is devoted to the algorithm by Gu which is the first algorithm that accurately estimates the distance to uncontrollability in polynomial time, see [18]. This algorithm requires $\mathcal{O}(n^6)$ floating point operations and the main cost is the result of eigenvalue calculations of sparse generalized eigenvalue problems of size $\mathcal{O}(n^2)$. Section 3.1 gives an overview of Gu's bisection method, while the details of Gu's novel scheme are discussed in Section 3.2. Recent improvements to his algorithm by Burke, Lewis and Overton [7], as well as Mengi, [19, 27], are discussed in Section 3.3.

3.1 Introduction to Gu's Bisection Method

In Equation (1.8) the distance to the nearest uncontrollable system is defined as

$$\tau(A,B) = \min_{\lambda \in \mathbb{C}} \sigma_n([A - \lambda I, B]). \tag{3.1}$$

This is a non-convex optimization problem in two variables, the real and imaginary parts of λ and may have multiple local minima as illustrated in Example 1.1.2. There is a vast literature on algorithms that have been designed to compute $\tau(A,B)$ and we provide a brief summary. Algorithms that search for a local minimum does not provide a guarantee that the minimum is also the global minimum. The algorithms designed to search for the global minimum require computing time that is inverse proportional to $\tau^2(A,B)$. In the case of a system that is nearly uncontrollable this cost is excessively expensive. For example, in Example 1.1.1 with $n = 40$ the distance is $\tau(A,B) = 1.53 \times 10^{-17}$ and the computational cost is proportional to 10^{34}. Lastly, the backward stable algorithms often fail to detect near uncontrollability. For more detail and references see [18].

Gu's algorithm computes the *global* minimum within a factor of two and is based on the following bisection method:

Algorithm 3.1.1 *(see Gu [18]).*

```
1. Set $\delta = \sigma_{min}([A, B])/2$
2. While $\delta \geq \tau(A, B)$,
   $\delta = \delta/2$
3. End
```

Past algorithms that was based on Algorithm 3.1.1 were excessively expensive, see [8, 17]. These methods are based on the minimization of the objective function in (1.8) where λ is restricted to a straight line in the complex plane. The algorithm proposed by Gu minimizes over the entire complex plane. The key to Gu's algorithm is the verification scheme that he developed, see [18]. Gu's scheme involves an intermediary test at each iteration: given two real numbers δ_1 and δ_2, the scheme returns information about the validity of one of the inequalities $\tau(A, B) \leq \delta_1$ or $\tau(A, B) > \delta_2$. In the case where both inequalities are satisfied, the scheme returns information about only one.

This scheme is based on the following theorem by Gu:

Theorem 3.1.1 *(see Gu [18]). Assume that $\delta > \tau(A, B)$. Given an $\eta \in [0, 2(\delta - \tau(A, B))]$, there exist at least two pairs of real number α and β such that*

$$\delta \in \sigma([A - (\alpha + \beta i)I, B]) \quad \textit{and} \quad \delta \in \sigma([A - (\alpha + \nu + \beta i)I, B] \tag{3.2}$$

where $\sigma(\cdot)$ denotes the set of singular values of its argument and $0 < \nu \leq 2(\delta - \tau(A, B))$.

Note the transfer of parameters involved. Instead of searching over all complex values for the existence of a λ which corresponds to a singular value smaller than δ, one requires the existence of *two* pairs of *real* numbers that allow δ to be in the corresponding set of singular values for some ν that decreases as δ decreases.

Using this theorem, the proof of which is detailed in [18], Gu transformed the test $\delta \geq \tau(A, B)$ into a two-part test whose main computation is the search for real eigenvalues of a $2n^2 \times 2n^2$ generalized eigenvalue problem. The details are presented in Section 3.2 but the two steps can be described as follows:

Firstly, for given $\delta > 0$ and $\nu > 0$ the numerical verification of a real solution to (3.2) is equivalent to the existence of a real eigenvalue to an associated $2n^2 \times 2n^2$ generalized eigenvalue problem. For each bisection step, set $\delta = \nu$ and verify if the eigenvalue problem has any real eigenvalues. Secondly, for each real eigenvalue, say α, one has to check if two matrix pencils share a common pure imaginary eigenvalue, say βi. If this is the case for at least one α, then one has found a pair α and β such that

$$\delta = \sigma([A - (\alpha + \beta i)I, B]) \quad \text{and hence} \quad \delta \geq \tau(A, B).$$

If no such real eigenvalue exists it follows by Theorem 3.1.1 that

$$\delta = \nu > 2\left(\delta - \tau(A, B)\right).$$

From the previous bisection step Algorithm 3.1.1 it follows that $2\delta \geq \tau(A, B)$ thus the value of δ after the execution of Algorithm 3.1.1 satisfies

$$\frac{\tau(A, B)}{2} \leq \delta \leq 2\tau(A, B).$$

For a given $\delta > 0$ the following algorithm is used to verify whether $\delta > \tau(A, B)$.

Algorithm 3.1.2 *Gu's Scheme(see [18])*

1. *Check for real eigenvalues in the generalized eigenvalue problem.*
2. *For each real eigenvalue α obtained in the first step, check for a common pure imaginary eigenvalue β_i*
3. *If either steps (2) or (3) fail, then by Theorem 3.1.1, $\delta = \nu > 2(\delta - \tau(A, B))$.*

The efficiency of Gu's scheme relies on the availability of efficient and accurate algorithms to compute the eigenvalues. The development of the scheme and idea to separate the real and imaginary parts of the search was truly novel.

3.2 Gu's Scheme

In this section we consider the numerical verification of a real solution to Equation 3.2. This discussion closely follows Gu's description of the scheme from [18] with additional explanation where beneficial.

Let $\delta > 0$, $\nu > 0$ and $A \in \mathbb{C}^{n\times n}$ and $B \in \mathbb{C}^{n\times m}$. The existence of nonzero vectors $\begin{pmatrix} x \\ y \end{pmatrix}$, z, $\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix}$ and $\hat{z}$ that satisfy

$$[A - (\alpha + \beta i)IB] \begin{pmatrix} x \\ y \end{pmatrix} = \delta z, \qquad \begin{pmatrix} A^* - (\alpha - \beta i)I \\ B^* \end{pmatrix} z = \delta \begin{pmatrix} x \\ y \end{pmatrix} \tag{3.3}$$

and

$$[A - (\alpha + \nu + \beta i)IB] \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \delta \hat{z}, \qquad \begin{pmatrix} A^* - (\alpha + \nu - \beta i)I \\ B^* \end{pmatrix} \hat{z} = \delta \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} \tag{3.4}$$

follow by the definition of singular values and Theorem 3.1.1. Note that A^* denotes the conjugate transpose of A.

These equations can be rearranged as

$$\begin{pmatrix} -\delta I & A-\alpha I & B \\ A^*-\alpha I & -\alpha I & 0 \\ B^* & 0 & -\delta I \end{pmatrix} \begin{pmatrix} z \\ x \\ y \end{pmatrix} = \beta i \begin{pmatrix} 0 & I & 0 \\ -I & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} z \\ x \\ y \end{pmatrix} \tag{3.5}$$

and

$$\begin{pmatrix} -\delta I & A-(\alpha+\nu I & B \\ A^*-(\alpha+\nu)I & -\alpha I & 0 \\ B^* & 0 & -\delta I \end{pmatrix} \begin{pmatrix} \hat{z} \\ \hat{x} \\ \hat{y} \end{pmatrix} = \beta i \begin{pmatrix} 0 & I & 0 \\ -I & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{z} \\ \hat{x} \\ \hat{y} \end{pmatrix}. \tag{3.6}$$

Using the QR factorization, with a focus on the $(1,3)$ and $(3,1)$ blocks of the first matrix, we can write

$$\begin{pmatrix} B \\ -\delta I \end{pmatrix} = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix} \begin{pmatrix} R \\ 0 \end{pmatrix}.$$

Since B is $n \times m$ and $\delta \neq 0$, $\begin{pmatrix} B \\ -\delta I_m \end{pmatrix}$ must have rank m, since the last m rows are linearly independent. The QR factorization preserves rank, thus we conclude that the $m \times m$ matrix R must also have rank m and is consequently nonsingular. Let

$$\begin{pmatrix} z_1 \\ y_1 \end{pmatrix} = Q^* \begin{pmatrix} z \\ y \end{pmatrix} \text{ and } \begin{pmatrix} \hat{z}_1 \\ \hat{y}_1 \end{pmatrix} = Q^* \begin{pmatrix} \hat{z} \\ \hat{y} \end{pmatrix}. \tag{3.7}$$

Since R is nonsingular, both systems $R^* z_1 = 0$ and $R^* \hat{z}_1 = 0$ each have unique solutions, namely $z_1 = 0$ and $\hat{z}_1 = 0$.

From the first equation in (3.7)

$$\begin{pmatrix} z_1 \\ y_1 \end{pmatrix} = Q^* \begin{pmatrix} z \\ y \end{pmatrix} \quad \text{it follows that} \quad \begin{pmatrix} z \\ y \end{pmatrix} = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix} \begin{pmatrix} z_1 \\ y_1 \end{pmatrix}.$$

Since $z_1 = 0$, this means that

$$\begin{pmatrix} z \\ y \end{pmatrix} = \begin{pmatrix} Q_{12}\ y_1 \\ Q_{22}\ y_1 \end{pmatrix}. \tag{3.8}$$

Substituting (3.8) into the first equation in (3.5),

$$\delta I z + (A-\alpha I)x + By = \beta i x \tag{3.9}$$

results in

$$(A-\alpha I)x + (BQ_{22} - \delta Q_{12})y_1 = \beta i x. \tag{3.10}$$

Similarly, the first equation in (3.6) becomes

$$(A-(\alpha+\nu)I)\hat{x}+(BQ_{22}-\delta Q_{12})\hat{y}_1=\beta i\hat{x}. \tag{3.11}$$

Note that both z and $\hat{z}$ are eliminated. In matrix form, (3.5) and (3.6) reduce to

$$\begin{pmatrix} A-\alpha I & BQ_{22}-\delta Q_{12} \\ -\delta I & (A^*-\alpha I)\,Q_{12} \end{pmatrix}\begin{pmatrix} x \\ y_1 \end{pmatrix}=\beta_i\begin{pmatrix} I & 0 \\ 0 & -Q_{12} \end{pmatrix}\begin{pmatrix} x \\ y_1 \end{pmatrix} \tag{3.12}$$

and

$$\begin{pmatrix} A-(\alpha+\nu)I & BQ_{22}-\delta Q_{12} \\ -\delta I & (A^*-(\alpha+\nu)I)\,Q_{12} \end{pmatrix}\begin{pmatrix} \hat{x} \\ \hat{y}_1 \end{pmatrix}=\beta_i\begin{pmatrix} I & 0 \\ 0 & -Q_{12} \end{pmatrix}\begin{pmatrix} \hat{x} \\ \hat{y}_1 \end{pmatrix}. \tag{3.13}$$

Since Q_{12} is part of the Q in the QR factorization, it must be nonsingular for each $\delta > 0$. Thus the matrix $\begin{pmatrix} I & 0 \\ 0 & -Q_{12} \end{pmatrix}$ from the right side of (3.12) and (3.13) must be nonsingular, implying that the matrices on the left side of the equations are nonsingular as well. If β_i is to be an eigenvalue of each matrix pencil, then there exists a non-zero matrix $X \in \mathbb{R}^{2n\times 2n}$ such that

$$\begin{aligned} \begin{pmatrix} A-\alpha I & BQ_{22}-\delta Q_{12} \\ -\delta I & (A^*-\alpha I)Q_{12} \end{pmatrix} X \begin{pmatrix} I & 0 \\ 0 & -Q_{12} \end{pmatrix}^* &= \\ \begin{pmatrix} I & 0 \\ 0 & -Q_{12} \end{pmatrix} X &\begin{pmatrix} A-(\alpha+\nu)I & BQ_{22}-\delta Q_{12} \\ -\delta I & (A^*-(\alpha+\nu)I)\,Q_{12} \end{pmatrix}^*. \end{aligned} \tag{3.14}$$

Partitioning X into $\begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix}$, (3.14) can be written as

$$\mathcal{H}u=0 \text{ and } \mathcal{A}u=2\alpha\mathcal{B}u \tag{3.15}$$

where $\mathcal{B}=\begin{pmatrix} 0 & 0 & -Q_{12}\otimes I & 0 \\ 0 & 0 & 0 & I\otimes Q_{12} \end{pmatrix}$; $u=\begin{pmatrix} vec(X_{11}) \\ vec(X_{22}) \\ vec(X_{12}) \\ vec(X_{21}) \end{pmatrix}$;

$$\mathcal{A}=\begin{pmatrix} \delta I & -Q_{12}\otimes\hat{B} & Q_{12}\otimes A-((A^*-\nu)Q_{12})\otimes I & 0 \\ -\delta I & \hat{B}\otimes Q_{12} & 0 & (A-\nu)\otimes Q_{12}-I\otimes(A^*Q_{12}) \end{pmatrix};$$

$$\mathcal{H}=\begin{pmatrix} I\otimes A-(A-\nu I)\otimes I & 0 & -\hat{B}\otimes I & I\otimes\hat{B} \\ 0 & ((A^*-\nu)Q_{12}\otimes Q_{12}-Q_{12}\otimes(A^*Q_{12}) & \delta Q_{12}\otimes I & -\delta I\otimes Q_{12} \end{pmatrix}$$

where $\hat{B}=BQ_{22}-\delta Q_{12}$. Here $\otimes$ is the Kronecker product, and $vec(C)$ is a vector formed by stacking the column vectors of matrix C. More information on Kronecker products and *vec* can be found in [1].

Using the RQ factorization,

$$\mathcal{H} = \begin{pmatrix} \mathcal{R} & 0 \end{pmatrix} \begin{pmatrix} \mathcal{Q}_{11} & \mathcal{Q}_{12} \\ \mathcal{Q}_{21} & \mathcal{Q}_{22} \end{pmatrix}$$

where $\mathcal{Q}_{ij} \in \mathbb{C}^{2n^2 \times 2n^2}$, and define

$$\begin{pmatrix} w \\ v \end{pmatrix} = \begin{pmatrix} \mathcal{Q}_{11} & \mathcal{Q}_{12} \\ \mathcal{Q}_{21} & \mathcal{Q}_{22} \end{pmatrix} u.$$

Then the following modifications can be made to the equation (3.15):

$\mathcal{H} = 0$ reduces to $\mathcal{R}w = 0$. Setting $w = 0$, $\mathcal{A}u = 2\alpha\beta u$ becomes

$$\mathcal{A}_1 v = 2\alpha\mathcal{B}_1 v \tag{3.16}$$

where

$$\mathcal{A}_1 = \mathcal{A}\begin{pmatrix} \mathcal{Q}_{21}^* \\ \mathcal{Q}_{22}^* \end{pmatrix} \quad \text{and} \quad \mathcal{B}_1 = \begin{pmatrix} -Q_{12} \otimes I & 0 \\ 0 & I \otimes Q_{12} \end{pmatrix} \mathcal{Q}_{22}^*.$$

The equation (3.16) is now a $2n^2 \times 2n^2$ generalized eigenvalue problem that can be solved efficiently for α using the QZ algorithm or its variants.

3.3 Modifications on Gu's Scheme

The first improvement on Gu's method was due to Burke, Lewis, and Overton [7]. In Gu's algorithm, they replaced Gu's bisection step with a trisection step. At each iteration, both a lower (L) and upper (U) bound on τ are updated. Depending on the result of Gu's test, either L is updated to δ_2 or U is updated to δ_1. This enables the reduction of $\frac{2}{3}$ at each iteration instead of a reduction of $\frac{1}{2}$.

Mengi, in his doctoral dissertation [27], presented an improvement to Gu's algorithm that reduces the complexity from $\mathcal{O}(n^6)$ to $\mathcal{O}(n^4)$ on average and $\mathcal{O}(n^5)$ in the worst case. In the real-eigenvalue search of the generalized eigenvalue problem (3.16), Mengi noticed that the fact that the search was for only real eigenvalues was not exploited. Predecessors used standard algorithms (QZ) to find all eigenvalues and then tested for the existence of real-eigenvalues. Mengi showed that the generalized eigenvalue problem (3.16) can be replaced by standard eigenvalue problem using vectorization. The inverse of the corresponding new matrix times a vector can be computed by solving a Sylvester equation of size $2n$. Using standard Sylvester equations solvers, the closest eigenvalue to a given real number can thus be computed efficiently by performing $\mathcal{O}(n^3)$ operations. Since only real eigenvalues are needed, a divide-and-conquer algorithm scanning only the real axis was used to find the existence of real eigenvalues at a computational cost of $\mathcal{O}(n^4)$. Gu, Mengi, Overton, Xia and Zhu [19] also published an alternative "adaptive progress " algorithm to scan for the real eigenvalues that was found to be inferior to the divide-and-conquer method.

t

Chapter 4

Linear Matrix Inequality Approach

The distance to uncontrollability can be regarded as a special case of the structured singular value computation, see [32]. Since linear matrix inequalities are very useful for the structured singular value computation, Ebihara expected that one can construct linear matrix inequality based algorithms to compute the distance to uncontrollability, see [15, 13]. The remark in [7] by Burke *et. al.* that it is not known if Gu's algorithm can be replaced by an LMI-based algorithm, motivated Ebihara to develop an LMI-based algorithm to compute upper and lower bounds on the distance to uncontrollability as well as conditions for exactness verification of the lower bound. In this discussion we only focus on the lower bound, for more information on the upper bound, see [15].

As with Gu's method, the LMI approach begins with Eising's representation as the search for a minimal singular value:

$$\tau(A,B) = \min_{\lambda \in \mathbb{C}} \sigma_n([A - \lambda I, B]). \tag{4.1}$$

Unlike in Gu's method, where the structure of the matrices was used to re-formulate the problem into a search for real eigenvalues, Ebihara instead considers at a more general case:

$$\tau^* = \min_{z \in \mathbb{C}} \sigma_{min}(P + zQ) \tag{4.2}$$

where $P,\ Q \in \mathbb{C}^{n \times (n+m)}$ and we assume that $\mathrm{Rank}(Q) = n$ throughout the discussion. The distance to uncontrollability is simply a special case of (4.2) with $P = [A\ B]$ and $Q = [-I_n\ 0_{n,m}]$.

Ebihara transforms this problem, (4.2), into an equivalent optimization problem of a single variable over a compact set. This process is discussed in more detail in Section 4.1 but a brief summary is given here. The first step is to express the variable z in terms of polar coordinates and then determining a bound on the radius r. The dependence on the angular component θ is removed via the Kalman-Yakubovič-Popov (KYP) lemma. This step introduces a matrix valued function depending on the radius $X : \mathbb{R} \to \mathbf{H}_n$. The Hermitian matrix is approximated by a finite power series and the approximate SDP is obtained using (D, G)-scaling, see [14], The solution to the SDP yields a lower bound on the distance to uncontrollability. The dual SDP yields a rank condition under which the lower bound coincides with the exact distance to uncontrollability. More detail on the dual SDP and the rank condition is provided in Sections 4.2 and 4.3

Ebihara specifically states that his primary interest is the insight arising from convex duality theory and not the development of computationally demanding LMI-based algorithms. Section 4.4 is devoted to discussion about the computational complexity of the method, preliminary numerical results, as well as benefits of the approach that are not directly related to our focus.

4.1 Formulation of Eising's Formula as an LMI

A discussion of how Ebihara determined a bound on the domain of z in

$$\tau^* = \min_{z \in \mathbb{C}} \sigma_{min}(P + zQ).$$

follows. The KYP lemma is applied to the resulting optimization problem to eliminate the dependence on the angular component θ and this optimization problem is approximated using a power series expansion.

Since Q has full-row rank,

$$\lim_{|z| \to \infty} \sigma_{min}(P + zQ) \to \infty$$

and $\sigma_{min}(P + zQ)$ takes the value $\sigma_{min}(P)$ when $z = 0$. Thus for a given Q and P and any real $M > 0$, there exists a real $\gamma > 0$ such that if $|z| > \gamma$, then $\sigma_{min}(P + zQ) > M$. Therefore there must be a compact disk centered at the origin in $\mathbb{C}$ where all global minimizers must lie, namely, $|z| \leq \gamma$.

Equation (4.2) can be expressed as an optimization problem over a compact set

$$\tau^* = \min_{z \in \mathbb{C}} \sigma_{min}(P + zQ) = \min_{|z| \leq \gamma} \sigma_{min}(P + zQ).$$

Using Schur complements and the matrix inversion formula, Ebihara derives a bound on γ. More detail about Schur complements and the matrix inversion formula is presented in the Appendix.

Proposition 4.1.1 *(Ebihara, [13]) The equation* (4.2) *for the lower bound on the distance to uncontrollability can be characterized by*

$$\tau^* = \min_{(r,\theta) \in [-\gamma, \gamma] \times [0, \pi)} \sigma_{min}(P + re^{i\theta}Q) \tag{4.3}$$

where

$$\gamma = \sqrt{\frac{\sigma_{min}(P)^2 + 1}{\lambda_{\min}(Q(I + P^*P)^{-1}Q^*)}}. \tag{4.4}$$

Proof Since Q has full rank and $I + P^*P$ is positive definite it follows that $\lambda_{min}(Q(I + P^*P)^{-1}Q^*) > 0$ and consequently γ is well-defined.

To prove (4.4), it is sufficient to show that the optimizer can not be greater than γ, that is $|r| \not> \gamma$.

Applying the matrix inversion formula (7.5), to $(I + P^*P)^{-1}$ results in

$$\begin{aligned}(I + P^*P)^{-1} &= I^{-1} - I^{-1}P^*(I + PI^{-1}P^*)^{-1}PI^{-1} \\ &= I - P^*(I + PP^*)^{-1}P.\end{aligned}$$

Thus, $Q(I + P^*P)^{-1}Q^* = QQ^* - QP^*(PP^* + I)^{-1}PQ^*$ and squaring (4.4) yields

$$r^2 > \frac{(\sigma_{min}(P))^2 + 1}{\lambda_{min}(QQ^* - QP^*(PP^* + I)^{-1}PQ^*)} \quad \text{for all} \quad |r| > \gamma$$

and thus

$$r^2\lambda_{min}(QQ^* - QP^*(PP^* + I)^{-1}PQ^*) - (\sigma_{min}(P))^2 + 1) > 0 \quad \text{for all} \quad |r| > \gamma.$$

Since $(\sigma_{\text{min}}(P))^2 + 1 = \lambda_{min}((\sigma_{\text{min}}(P))^2)I - I)$, and symmetric matrices have positive eigenvalues if and only if they are positive definite, this is equivalent to

$$r^2(QQ^* - QP^*(PP^* + I)^{-1}PQ^*) - \sigma_{\text{min}}(P)^2 I - I \succ 0. \tag{4.5}$$

Since, for any non-zero P, the matrix PP^* is positive definite, we also have

$$PP^* + I \succ 0. \tag{4.6}$$

Using the Schur complement characterization, (7.1.2) these two matrix inequalities are equivalent to

$$\begin{bmatrix} PP^* + I & rQP^* \\ rPQ^* & r^2QQ^* - (\sigma_{min}(P))^2 I - I \end{bmatrix} \succ 0 \quad \text{for all} \quad |r| > \gamma. \tag{4.7}$$

since the Schur complement of the first block,

$$r^2QQ^* - (\sigma_{min}(P))^2 I - I - rQP^*(PP^* + I)^{-1}rPQ^*$$

is equivalent to (4.5), so that (4.5) and (4.6) are true if and only if (4.7) holds. Pre-multiplying (4.7) by $\begin{bmatrix} e^{-i\theta}I & I \end{bmatrix}$ results in

$$\begin{bmatrix} e^{-i\theta}PP^* + e^{-i\theta}I + rQP^* & re^{-i\theta}PQ^* + r^2QQ^* - (\sigma(P))^2I - I \end{bmatrix} \succ 0$$

and post-multiplying (4.7) by $\begin{bmatrix} e^{i\theta}I \\ I \end{bmatrix}$ yields

$$\begin{bmatrix} PP^* + I + re^{i\theta}QP^* + re^{-i\theta}PQ^* + r^2QQ^* - (\sigma(P))^2I - I \end{bmatrix} \succ 0$$

which simplifies to

$$(P + re^{i\theta}Q)(P + re^{i\theta}Q)^* \succ (\sigma_{min}(P))^2I \quad \text{for all} \quad |r| > \gamma \quad \text{and} \quad \theta \in [0, \pi). \tag{4.8}$$

From (4.8) it follows that

$$\sigma_{\min}(P + re^{i\theta}Q) > \sigma_{min}(P) \quad \text{for all} \quad |r| > \gamma \quad \text{and} \quad \theta \in [0, \pi)$$

and consequently

$$\tau^*_\gamma = \min_{|r|>\gamma, \theta\in[0,\pi)} \sigma_{min}(P + re^{i\theta}) > \sigma_{min}(P).$$

This shows that the minimum, τ^* in (4.3), will occur for $|r| < \gamma$ since $\tau^* \leq \sigma_{\min}(P) < \tau^*_\gamma$. ∎

Remark Clearly this step is focused on theory and not computational application. The calculation of the value of γ requires calculating a matrix inverse of size $(n + m) \times (n + m)$, followed by the singular value and eigenvalue computations with different matrices of the same order. Hence the search for the compact set is in itself an intractable problem. However, such a reduction is theoretically necessary in order to justify, using the Kalman-Yakubovič-Popov (KYP) lemma, the elimination of the dependence on the angular component θ in (4.3).

The detail of the KYP lemma is beyond the scope of this thesis but we include some background. The KYP lemma establishes equivalence between a frequency domain inequality for a transfer function and the linear matrix inequality for its state space realization.

The lemma was generalized by Anderson [2] to a condition for positive realness of a multivariable transfer function. More recently, see [5], the KYP lemma has been used to replace a frequency dependence with a new matrix valued decision variable. The resulting inequality is then solved numerically. This takes the opposite direction of the initial applications of the lemma where the search for a matrix variable was converted to a frequency inequality.

Consider the following result form Rantzer in [35].

Theorem 4.1.1 *Given A, B, M with $\det(e^{j\omega}I - A) \neq 0$ for $\omega \in \mathbb{R}$ and (A, B) controllable. Then the following two statements are equivalent:*

(i)

$$\begin{bmatrix} (e^{j\omega}I - A)^{-1}B \\ I \end{bmatrix}^* M \begin{bmatrix} (e^{j\omega}I - A)^{-1}B \\ I \end{bmatrix} \preceq 0 \quad \textit{for all} \quad \omega \in \mathbb{R}.$$

(ii) There exists a matrix $P \in \mathbb{R}^{n\times n}$ such that $P = P^T$ and

$$M + \begin{bmatrix} A^TPA - P & A^TPB \\ B^TPA & B^TPB \end{bmatrix} \prec 0,$$

The corresponding equivalence of strict inequalities holds even if (A, B) is not controllable.

Equation (4.3) can be rewritten as

$$\tau^* = \max \tau \quad \text{subject to}$$
$$\begin{bmatrix} I \\ e^{i\theta} \end{bmatrix}^* \begin{bmatrix} r^2QQ^* - \tau^2 I & rQP^* \\ rPQ^* & PP^* \end{bmatrix} \begin{bmatrix} I \\ e^{i\theta} \end{bmatrix} \succeq 0 \text{ for all } (r,\theta) \in [-\gamma,\gamma] \times [0,\pi). \tag{4.9}$$

Applying the KYP lemma an equivalent formulation is obtained: For each fixed r, the constraint in (4.9) holds for all values of θ if and only if there exists an $n \times n$ Hermitian matrix $X(r)$, such that

$$\begin{bmatrix} r^2QQ^* - \tau^2 I - X(r) & rQP^* \\ rPQ^* & PP^* + X(r) \end{bmatrix} \succeq 0.$$

Problem (4.9) can thus be reformulated as

$$\tau^* = \max_{X(r) \in \mathbf{H}_n} \tau \quad \text{subject to}$$

$$\begin{bmatrix} r^2QQ^* - \tau^2 I - X(r) & rQP^* \\ rPQ^* & PP^* + X(r) \end{bmatrix} \succeq 0 \text{ for all } r \in [-\gamma,\gamma]. \tag{4.10}$$

Thus the original optimization problem has been reduced from a search over the entire complex plane to a search for a single variable over a compact set, but a parameter dependent matrix, $X(r)$, has been introduced. The search for $X(r)$ over an infinite dimensional function space of Hermitian matrices makes that the problem remains intractable, see [13]. To enable LMI optimization to calculate estimates of τ^*, Ebihara considers N^{th} degree polynomial approximations for this parameter dependent matrix, $X(r) = \sum_{i=0}^{N} r^i X_i$ (where X_i are Hermitian). The associated approximation of τ^* is denoted by τ_N^*. It follows that $\tau^* \geq \tau_j^* \geq \tau_k^*$ where $j \geq k$.

Using an order two approximation, $X(r) = \sum_{i=0}^{2} r^i X_i$, and (D, G)-scaling strategies presented in [14], (4.10) takes the form

$$
\begin{aligned}
&\tau_2^* = \max_{X_0,X_1,X_2,D,G} \tau \text{ subject to} \\
&\left[\begin{array}{cc:cc}
-\tau^2 I - X_0 & 0 & -\frac{1}{2}X_1 & QP^* \\
0 & PP^* + X_0 & 0 & \frac{1}{2}X_1 \\
\hdashline
-\frac{1}{2}X_1 & 0 & QQ^* - X_2 & 0 \\
PQ^* & \frac{1}{2}X_1 & 0 & X_2
\end{array}\right] + \begin{bmatrix} -\gamma^2 D & G \\ G^* & D \end{bmatrix} \succeq 0
\end{aligned}
\tag{4.11}
$$

where X_0, X_1, X_2 are Hermitian of size n, D is positive semidefinite of size $2n$, and G is symmetric of size $2n$. This SDP has $(11n^2 + 3n)/2$ scalar variables, while the LMIs are of size $6n$.

According to Ebihara, this second order approximation gave accurate results for most of the test cases that he considered in [15]. Note that we have not considered higher order approximations.

4.2 Formulation of the Dual Problem

Using duality theory, Ebihara was able to reduce (4.11) into an equivalent SDP with $2n^2 + n$ scalar variables and LMIs of size $5n$.

Using duality theory, the estimate for $\tau*$ given by the SDP (4.11) was rewritten as

$$
\begin{aligned}
{\tau_2^*}^2 &= \min_H \ \text{trace} \left(\begin{bmatrix} PP^* & PQ^* \\ QP^* & QQ^* \end{bmatrix} \right) \left(\begin{bmatrix} H_{11} & H_{12}^* \\ H_{12} & H_{22} \end{bmatrix} \right) \\
\text{subject to } H &= \begin{bmatrix} H_{11} & H_{12} \\ H_{12}^* & H_{22} \end{bmatrix} \succeq 0, \\
\widehat{H} &= \begin{bmatrix} H_{11} & H_{12}^* \\ H_{12} & H_{22} \end{bmatrix} \succeq 0, \\
&\gamma^2 H_{11} \succeq H_{22}, \\
&\text{trace}(H_{11}) = 1.
\end{aligned}
\tag{4.12}
$$

A proof of this result is detailed in [13]; key aspects are verification of Slater's constraint qualification and reduction using the properties of the trace operator.

Proof (Ebihara) [13]

Letting $\tau_2^* = \sqrt{-\nu^*}$, (4.11) is equivalent to

$$\min_{X_0,X_1,X_2,D,G} \nu \ \text{ subject to}$$
$$\left[\begin{array}{cc|cc} \nu I - X_0 & 0 & -\frac{1}{2}X_1 & QP^* \\ 0 & PP^* + X_0 & 0 & \frac{1}{2}X_1 \\ \hline -\frac{1}{2}X_1 & 0 & QQ^* - X_2 & 0 \\ PQ^* & \frac{1}{2}X_1 & 0 & X_2 \end{array}\right] + \begin{bmatrix} -\gamma^2 D & G \\ G^* & D \end{bmatrix} \succeq 0 \tag{4.13}$$

where X_0, X_1, X_2 are Hermitian of size n, D is positive semidefinite of size $2n$, and G is symmetric of size $2n$ and ν^* denotes the optimal value of (4.13).

Let $\mathcal{M}(\nu, X_0, X_1, X_2, D, G)$ denote the left-side of (4.13). By setting $\nu = 4\gamma^2$, $X_0 = 2\gamma^2 I_n$, $X_1 = X_2 = 0$, $D = I_{2n}$ and $G = \begin{bmatrix} 0 & -QP^* \\ PQ^* & 0 \end{bmatrix}$ we can confirm that Slater's constraint qualification, see [36], holds for all P and Q:

Substituting the values above yields the matrix

$$\mathcal{M}(4\gamma^2,\ 2\gamma^2 I_n,\ 0,\ 0,\ I_{2n},\ G) = \left[\begin{array}{cc|cc} \gamma^2 I & 0 & 0 & 0 \\ 0 & PP^* + \gamma^2 I & PQ^* & 0 \\ \hline 0 & QP^* & QQ^* + I & 0 \\ 0 & 0 & 0 & I \end{array}\right],$$

which is strictly positive definite for all P and Q, thus (4.13) is strictly feasible. This strict feasibility (Slater's Constraint Qualification) of the resulting SDP for any P and Q allows characterization of the SDP using the dual problem.

Denote the Lagrange multipliers $\mathcal{H}$ and $\mathcal{J}$ (each positive definite of size $4n$ and $2n$, respectively) and define the Lagrangian

$$\mathcal{L}(\nu, X_0, X_1, X_2, D, G, \mathcal{H}, \mathcal{J}) = \nu - \operatorname{Tr}\left(\mathcal{M}(\nu, X_0, X_1, X_2, D, G)\mathcal{H}\right) - \operatorname{Tr}\left(D\mathcal{J}\right).$$

Partitioning $\mathcal{H}$ as

$$\mathcal{H} = \begin{bmatrix} \mathcal{H}_{11} & \mathcal{H}_{12} & \mathcal{H}_{13} & \mathcal{H}_{14} \\ \mathcal{H}_{12}^* & \mathcal{H}_{22} & \mathcal{H}_{23} & \mathcal{H}_{24} \\ \mathcal{H}_{13}^* & \mathcal{H}_{23}^* & \mathcal{H}_{33} & \mathcal{H}_{34} \\ \mathcal{H}_{14}^* & \mathcal{H}_{24}^* & \mathcal{H}_{34}^* & \mathcal{H}_{44} \end{bmatrix}$$

where $\mathcal{H}_{ii},\ i = 1,\ \ldots,\ 4$ is symmetric and of size n.

The Lagrangian, $\mathcal{L}(\nu, X_0, X_1, X_2, D, G, \mathcal{H}, \mathcal{J})$, can be written equivalently as

$$\begin{aligned}(1 - \mathrm{Tr}\,(\mathcal{H}_{11})) + \mathrm{Tr}\,&(X_0(\mathcal{H}_{11} - \mathcal{H}_{22}))\\
&+ \frac{1}{2}\mathrm{Tr}\,(X_1(\mathcal{H}_{13} + \mathcal{H}_{13}^* - \mathcal{H}_{24} - \mathcal{H}_{24}^*)\\
&+ \mathrm{Tr}\,(X_2(\mathcal{H}_{33} - \mathcal{H}_{44}))\\
&+ \mathrm{Tr}\left(D\left(\gamma^2\begin{bmatrix}\mathcal{H}_{11} & \mathcal{H}_{12}\\ \mathcal{H}_{12}^* & \mathcal{H}_{22}\end{bmatrix} - \begin{bmatrix}\mathcal{H}_{33} & \mathcal{H}_{34}\\ \mathcal{H}_{34}^* & \mathcal{H}_{44}\end{bmatrix} - \mathcal{J}\right)\right)\\
&+ \mathrm{Tr}\left(G\left(\begin{bmatrix}\mathcal{H}_{13} & \mathcal{H}_{14}\\ \mathcal{H}_{23} & \mathcal{H}_{24}\end{bmatrix} - \begin{bmatrix}\mathcal{H}_{13}^* & \mathcal{H}_{23}^*\\ \mathcal{H}_{14}^* & \mathcal{H}_{24}^*\end{bmatrix}\right)\right)\\
&- \mathrm{Tr}\left(\begin{bmatrix}PP^* & PQ^*\\ QP^* & QQ^*\end{bmatrix}\begin{bmatrix}\mathcal{H}_{22} & \mathcal{H}_{14}^*\\ \mathcal{H}_{14} & \mathcal{H}_{33}\end{bmatrix}\right). \end{aligned} \tag{4.14}$$

In order for $\mathcal{L}$ to be bounded below for any ν, X_0, X_1, X_2, D and G, all the addends above must be equal to zero at the minimum value. Thus

$$\begin{aligned}\mathrm{Tr}\,(\mathcal{H}_{11}) = 1, \quad \mathcal{H}_{11} = \mathcal{H}_{22},\\
\mathcal{H}_{13} + \mathcal{H}_{13}^* = \mathcal{H}_{24} + \mathcal{H}_{24}^*, \quad \mathcal{H}_{33} = \mathcal{H}_{44},\\
\gamma^2\begin{bmatrix}\mathcal{H}_{11} & \mathcal{H}_{12}\\ \mathcal{H}_{12}^* & \mathcal{H}_{22}\end{bmatrix} - \begin{bmatrix}\mathcal{H}_{33} & \mathcal{H}_{34}\\ \mathcal{H}_{34}^* & \mathcal{H}_{44}\end{bmatrix} - \mathcal{J} = 0,\\
\begin{bmatrix}\mathcal{H}_{13} & \mathcal{H}_{14}\\ \mathcal{H}_{23} & \mathcal{H}_{24}\end{bmatrix} = \begin{bmatrix}\mathcal{H}_{13}^* & \mathcal{H}_{23}^*\\ \mathcal{H}_{14}^* & \mathcal{H}_{24}^*\end{bmatrix}.\end{aligned} \tag{4.15}$$

The final line in (4.14) thus leads to the dual SDP

$$\nu^* = \max_{\mathcal{H}\geq 0,\ \mathcal{J}\geq 0} -\mathrm{Tr}\left(\begin{bmatrix}PP^* & PQ^*\\ QP^* & QQ^*\end{bmatrix}\begin{bmatrix}\mathcal{H}_{22} & \mathcal{H}_{14}^*\\ \mathcal{H}_{14} & \mathcal{H}_{33}\end{bmatrix}\right) \tag{4.16}$$

subject to (4.14).

Rearrangement and substitution using (4.14) results in

$$\tau_2^{*2} = \min_{\mathcal{H}} \mathrm{Tr}\left(\begin{bmatrix}PP^* & PQ^*\\ QP^* & QQ^*\end{bmatrix}\begin{bmatrix}\mathcal{H}_{11} & \mathcal{H}_{14}^*\\ \mathcal{H}_{14} & \mathcal{H}_{33}\end{bmatrix}\right)$$

subject to

$$\begin{aligned}\mathcal{H} = \begin{bmatrix}\mathcal{H}_{11} & \mathcal{H}_{12} & \mathcal{H}_{13} & \mathcal{H}_{14}\\ \mathcal{H}_{12}^* & \mathcal{H}_{11} & \mathcal{H}_{14}^* & \mathcal{H}_{13}\\ \mathcal{H}_{13} & \mathcal{H}_{14} & \mathcal{H}_{33} & \mathcal{H}_{34}\\ \mathcal{H}_{14}^* & \mathcal{H}_{13} & \mathcal{H}_{34}^* & \mathcal{H}_{33}\end{bmatrix} &\succeq 0\\
\gamma^2\begin{bmatrix}\mathcal{H}_{11} & \mathcal{H}_{12}\\ \mathcal{H}_{12}^* & \mathcal{H}_{11}\end{bmatrix} &\succeq \begin{bmatrix}\mathcal{H}_{33} & \mathcal{H}_{34}\\ \mathcal{H}_{34}^* & \mathcal{H}_{33}\end{bmatrix},\ \mathrm{Tr}\,(\mathcal{H}_{11}) = 1.\end{aligned} \tag{4.17}$$

All the blocks of $\mathcal{H}$ except for $\mathcal{H}_{11}$, $\mathcal{H}_{14}^*$, $\mathcal{H}_{33}$ can be taken as zero without loss of generality. This is because these variables are the only ones relevant to the optimal function in SDP (4.17). The positive semi-definite condition $\mathcal{H} \succeq 0$ can be rewritten using a similarity transformation with no other variables on the diagonal blocks. By renaming $\mathcal{H}_{11} = H_{11}$, $\mathcal{H}_{14}^* = H_{12}$, and $\mathcal{H}_{22} = H_{33}$, the SDP (4.12) follows. ∎

4.3 Exactness Verification

Under certain conditions, Ebihara was able to demonstrate theoretically that the estimate τ_2^* is equivalent to τ^*. These conditions rely on the rank of the optimal H and $\widehat{H}$ obtained using Equation (4.12).

Let the following denote the full-rank factorization of such optimal H and $\widehat{H}$:

$$H = \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \end{bmatrix}^*; \quad \widehat{H} = \begin{bmatrix} \widehat{H_1} \\ \widehat{H_2} \end{bmatrix} \begin{bmatrix} \widehat{H_1} \\ \widehat{H_2} \end{bmatrix}^*. \tag{4.18}$$

Theorem 4.3.1 *Exactness Verification (Ebihara [13])*

In Equation (4.18)*, if both H_1 and $\widehat{H_1}$ are full column-rank, then $\tau_2^* = \tau^*$. If both H_1 and $\widehat{H_1}$ are rank-one, then $\tau_2^* = \tau^*$.*

The proof of the equality is detailed in [13]. One direction ($\tau_2^* \leq \tau^*$) follows from the definition of τ_N^*. The other direction relies on the structural relationship between H and $\widehat{H}$, the KYP lemma, and the rank condition to construct a representation for a factorization of H whence τ_2^* and τ^* can be compared.

Thus Ebihara's method can be thought of as having two steps. The first is to find both τ_2^* and the optimal value of H solving SDP (4.12). The second is to verify the exactness of τ_2^* by computing the full rank factorizations of both H and $\widehat{H}$ to see if the column-rank conditions are satisfied. Ebihara makes the statement ... *the exactness test ... has been derived successfully by relying on the particular structure of the distance to uncontrollability computation problem.*, See [13, p.3]. It should be noted that the computation of the rank of a matrix can be ill-conditioned. In numerical experiments with smaller matrices, the rank conditions were always found to hold, but no theoretical justification of the conditions on the matrices in the distance to uncontrollability was given that would imply that the exactness verification should hold, in general.

4.4 Performance of the LMI-based method

A Matlab routine was created in order to test the performance of the approximation of τ^* via the SDP (4.12) using the primal-dual method. A routine was developed to find γ using standard Matlab functions for finding the inverse of $Q(I + P^*P)$, the minimum eigenvalue of $(Q(I + P^*P))^{-1}Q^*)$ and the minimum singular value of P^2. This value was necessary to define the constraints in (4.12).

In order to use SeDuMi to perform the SDP optimization, the Matlab toolbox Yalmip [23] was used to enter the constraints and objective function. Yalmip enables users to enter constraints and matrix variables in an intuitive way and then stores and passes the arguments in the form required for the chosen SDP optimizing software. According to the SeDuMi website [39], it is the most popular way of using SeDuMi.

There were several numerical results published in [15, 13] in which the author specified that SeDuMi was used as the optimization toolbox, but there was neither code available nor mention of a interface for the entering of constraints and objectives. However, for every reproducible numerical result, the identical value for τ_2^* was obtained in testing with similar computation time. Thus we feel confident that the code written to perform the optimization using this method is accurate and no less efficient than the code used in the published numerical results. Specifically, in Table 6.1, Section 6.1, the results from pairs Grcar_104, Markov_Chain104, Hatano52, Gallery52, Skew_Laplacian83, and Twisted104 show agreement with those of [13, 15].

The overall computational performance of the method was poor in comparison to the other methods, as can be seen in Table 6.1. It was hypothesized in [15] that the method was numerically unstable for matrices with large singular values.

The computation to calculate γ is in itself a computationally demanding problem for large matrices, and the fact that the method requires LMI of size $5n$ makes storage an issue for larger matrices as well. While the numerical results were poor, there may be ways of bypassing the computation of γ or making the SeDuMi call more efficient which could make the method more useful for calculation.

Chapter 5

Sum-of-Squares Approach

This chapter focuses on the recent work of Dumitrescu, Şicleru and Ştefan, see [12], who also also responded to the comment in [7] that *"it does not seem to be known whether Gu's test could be replaced by an LMI-based test."*

Dumitrescu *et. al.* transformed the non-convex optimization problem, equation (1.8),

$$\tau(A, B) = \min_{\lambda \in \mathbb{C}} \sigma_n([A - \lambda I \;\; B]),$$

into the minimization of the smallest eigenvalue of a bivariate real polynomial with matrix coefficients. Using a sum-of-squares relaxation and Gram matrix parameterization, the problem are transformed into a semidefinite program.

5.1 Formulation of the Sum-Of-Squares Relaxation

Equation (1.8) is equivalent to the computation of the square root of the smallest eigenvalue of

$$P(\lambda) = |\lambda|^2 I - \lambda A^* - \overline{\lambda} A + AA^* + BB^* \tag{5.1}$$

for $\lambda \in \mathbb{C}$. This is equivalent to the optimization problem

$$\tau_0 = \max_{\tau \geq 0} \tau \quad \text{such that} \quad P(\lambda) - \tau I \geq 0, \qquad \text{for all} \;\; \lambda \in \mathbb{C}. \tag{5.2}$$

The distance to uncontrollability is $\tau^* = \sqrt{\tau_0}$.

Setting $\lambda = x + iy$, and using the real and imaginary parts of λ as real variables, (5.1) takes the form

$$P(x, y) = (x^2 + y^2)I - x(A + A^*) - iy(A^* - A) + AA^* + BB^*. \tag{5.3}$$

Thus the distance to uncontrollability, (5.2), is equivalent to finding the largest value of τ which satisfies the semidefinite constraint $P(x, y) - \tau I \succeq 0$. Note that even though this constraint is convex, it is hard to implement. Dumitrescu *et. al.* proposes a sum-of-squares approximation.

Instead of maximizing (5.2) over the semidefinite constraint, they approximate τ_0 by τ_1 where

$$\tau_1 = \max_{\tau \geq 0} \tau \quad \text{subject to} \quad P(x,y) - \tau I \quad \text{is sum-of-squares} \tag{5.4}$$

where a bivariate polynomial is sum-of-squares if it can be written as

$$\sum_{k=1}^{\nu} \mathbf{F}_k(x,y)\mathbf{F}_k^*(x,y).$$

Since the degree of $P(x,y)$ is two, each polynomial $\mathbf{F}_k(x,y)$ has degree one. It is important to note that not all positive polynomials are sum-of-squares but all sum-of-squares polynomials are positive.

The optimization problem (5.3) can be transformed into an SDP using the Gram-matrix parameterization, see [9, 34].

Theorem 5.1.1 *[12] The matrix polynomial $P(x,y) - \tau I$ is sum-of-squares if and only if there exists a positive semidefinite matrix $Q \in \mathbb{C}^{3n \times 3n}$ such that, for $\Phi(x,y) = \begin{bmatrix} Ix & Iy & I \end{bmatrix}$,*

$$P(x,y) - \tau I = \Phi^T(x,y) \cdot Q \cdot \Phi(x,y). \tag{5.5}$$

Equating the coefficients of x^2, y^2, x and y in (5.3), the matrix Q has the form

$$Q = \begin{bmatrix} AA^* + BB^* - \tau I & -A^* - X & -iA^* - Y \\ -A + X & I & -Z \\ iA + Y & Z & I \end{bmatrix} \tag{5.6}$$

where $X, Y, Z \in \mathbb{C}^{n \times n}$ are each skew-Hermitian ($Z^* = -Z$) to provide the symmetry of Q.

The sum-of-squares relaxation (5.3) can be solved via the equivalent SDP

$$\tau_1 = \max_{\tau \geq 0} \tau \quad \text{subject to} \quad Q \succeq 0 \tag{5.7}$$

whose variables are the elements of the matrices X, Y, and Z.

5.2 Performance of the Sum-Of-Squares Approximation

Since the set of sum-of-squares matrices is a subset of the set of positive semidefinite matrices, the theoretical value of τ_1 in (5.3) should be less than or equal to τ_0 in (5.2). Theoretically, the calculation of τ using the sum-of-squares approximation should result in a radius that is more conservative than necessary for controllability. However, when comparisons are made between τ_1 and the results of Gu's method, the computed values of τ_1 are greater than the maximum of Gu's range [19] for the distance to uncontrollability. In [12], two cases of matrix pairs (Godunov (7,3) and Skew_Laplacian(8,3)) from the data at [26] are listed as such examples.

We also should be concerned with the relative distance between τ_0 and τ_1. If the computed τ_1 is significantly smaller than that of τ_0, then one may assume a problem is ill-conditioned when in fact it might not be. The error the other direction is much more important—if the computed τ_1 is actually significantly greater than the theoretical τ_0, then one might conclude, incorrectly, that a perturbation would not result in an uncontrollable system. The conditioning of the matrix A seems to be correlated with this numerical phenomena. In Section 5 more detailed results are shown.

In terms of complexity, problem (5.7) is $\mathcal{O}(n^6)$. This is because the order of the matrix variable is $\mathcal{O}(n) \times \mathcal{O}(n)$, while the number of free variables in Q is $\mathcal{O}(n^2)$, see [38] .

5.3 Sum-Of-Square Polynomials

This section discusses the definition and characteristics of sum-of-square polynomials, first for the real-valued case and then for the matrix polynomial case. In the case of real-valued polynomials, it can be shown that the set of non-negative valued polynomials and the set of sum-of-squares polynomials are coincident, while in the matrix polynomial case the set of sum-of-squares matrix polynomials is a proper subset of the set of positive semidefinite matrix polynomials. Thus optimizing over the set of sum-of-squares polynomials is a relaxation of an optimization over all positive semidefinite matrices.

Let $\mathbb{R}[x, y]$ denote the set of real-valued polynomials in x and y. A polynomial $F(x, y)$ is sum-of-squares if it can be written as (see [34])

$$F(x, y) = \sum_i f_i^2(x, y) \qquad \text{where} \quad f_i \in \mathbb{R}[x, y]. \tag{5.8}$$

Thus if a polynomial is sum-of-squares, this means that it is automatically non-negative. Hence the set of all sum-of-squares polynomials (of degree n in m real variables) is a subset of the set of all non-negative polynomials (of degree n in m real variables).

In general, a positive semidefinite polynomial need not be sum-of-squares [9]. One example given in [34] is the Motzkin form, $M(x, y, z) = x^4y^2 + x^2y^4z^6 - 3x^2y^2z^2$. Here a *form* is a homogenous polynomial, i.e., one in which the total degree of each term is constant. Since polynomials have a finite degree it implies that for polynomials that have a sum-of-squares representation, the sum must be finite, i.e., there is a v such that $F(x, y) = \sum_i^v f_i^2(x, y)$.

Hilbert proposed several conditions on the degree of the form, m and the number of variables n, where the set of non-negative forms agrees with the set of sum-of-squares forms, see [34]. For example, one of Hilbert's conditions is that if the degree of the forms is two, the set of non-negative forms of degree m in 2 variables is equivalent to the set of forms of degree 2 in two variables that can be written as a sum-of-squares.

Given a polynomial, it can be converted to a form by multiplying each monomial term by powers of a new variable so that the total degree of each monomial term is the same.

Parrilo, see [34], outlined the method for searching for a sum-of-squares polynomial as follows:

Express given polynomial $F(\mathbf{x})$ of degree $2d$ as a quadratic form in all the monomials of degree less than or equal to d given by the products of x and y.

$$F(\mathbf{x}) = \mathbf{z}^T Q \mathbf{z}, \quad \mathbf{z} = [1, x_1, x_2 x_1 x_2, x_1^2, x_2^2, x_1^2 x_2, x_1 x_2^2, x_1^3, x_2^3, \ldots x_m^d, y_m^d]. \tag{5.9}$$

This is called the Gram matrix parameterization of F, see [9]. If Q is positive semidefinite, then $F(x,y)$ is non-negative. Since the elements of $\mathbf{z}$ are not algebraically independent, there can be more than one representation for Q, some of which may not be positive semidefinite. The set of matrices Q that satisfy (5.9) is a linear manifold, since the entries of Q are determined by the coefficients of $\mathbf{x}$. Suppose that the intersection of this subspace and the positive semidefinite matrix cone is nonempty. Then using an eigenvalue factorization of such Q, $Q = T^T D T$, where the elements of diagonal matrix D are the non-negative real eigenvalues of Q and T is triangular, so that $F(x) = \sum_i d_i T(z)_i^2$. Thus F is sum-of-squares and therefore non-negative.

Example 5.3.1 *With a system of size 1 and two variables x and y, we would search for solutions to $F(x,y) = \begin{bmatrix} 1 & x & y \end{bmatrix}^T Q \begin{bmatrix} 1 & x & y \end{bmatrix}$ over positive semidefinite matrices Q of size 3×3.*

For the matrix polynomial case, the polynomial $Q(x,y) = P(x,y) - \tau I$ is of degree 2, since the degree of a polynomial is the largest of the degrees of its monomial terms, but it is not a form since not all the terms have the same degree.

$$Q(x,y) = (x^2 + y^2)I - x(A + A^*) - iy(A^* - A) + AA^* + BB^* - \tau I.$$

In order to change the polynomial into a form, we would have to introduce a new variable, say z, and multiply each term by a factor of z to bring each monomial term up to degree 2.

$$Q(x,y,z) = (x^2 + y^2)I - xz(A + A^*) - iyz(A^* - A) + z^2(AA^* + BB^* - \tau I).$$

Unfortunately, since the polynomial is matrix-valued instead of real-valued, Hilbert's theorem can not be applied.

Chapter 6

Numerical Results

Numerical testing was conducted on a MacBook Pro running OSX Version 10.5.8 with 2.16 GHz Intel Core 2 Duo. Matlab version 7.6(b) was used in all calculations, and SeDuMi version 1.21 was used in SDP optimizations.

6.1 Performance Using Smaller Matrices

The data provided by Mengi in [26] was used as benchmark data for both Dumitrescu [12] and Ebihara [13, 15]. The data consist of real square matrices that have significance or history (as the first matrix in the pair A, B) and a randomly generated rectangular second matrix. The numbers correspond to the size of the system. For example, Demmel104 consists of a square matrix of size 10 that is notoriously ill-conditioned as well as a 10×4 matrix with entries between 0 and 1. While Ebihara and Dumitrescu each gave some numerical results using this data, we repeated the tests for verification purposes as well as to explore in more detail the issues of speed and accuracy.

The code for Dumitrescu's method is available online, see [11], and was used without modification. A brief, simple routine was written using version 3 of Yalmip, see [23], to run Ebihara's code using SeDuMi. In order to compare the methods to each other, and to verify the published results and the validity of the code, the tests were repeated. The results are listed in Table 6.1.

The error columns in Table 6.1 refer to the relative error between the computed distance to uncontrollabilty of the matrix pair. Mengi [19] published an interval for the distance to uncontrollability, and the minimum value for the distance to uncontrollability from Gu's method. The relative error was calculated using the minimum of this interval to ensure that it would not be an underestimate. We can see that the sum of squares method equals or outperforms the method of Ebihara in terms of accuracy. It is also notable that of the three

Table 6.1: Comparison of Dumetriscu and Ebihara Methods with Gu's Method.

Name	Dumetriscu	Dumetriscu Error	Ebihara	Ebihara Error
Airy104	0.16344	0	0.24677	0.50975
Airy52	0.03767	0.0004	0.37729	9.01561
BasorMorrison104	0.60978	0	1.08439	0.77828
BasorMorrison52	0.68928	0	1.33688	0.93950
Chebyshev104	0.82710	0	0.82710	0
Chebyshev52	0.75034	0	0.75034	0
Companion104	0.71779	0.53979	5.57517	10.95977
Companion52	0.42435	0	0.4490	0.05801
ConvectionDiffusion104	1.48586	0	1.48586	0
ConvectionDiffusion52	0.69836	0	0.69836	0
Davies104	0.70012	0	2.21319	2.16116
Davies52	0.23175	0	1.68552	6.27269
Demmel104	0.12011	0.00041	3730.44587	31070.50984
Demmel52	0.09056	0.00004	69569	76820.4505
Frank104	0.67413	0	0.67413	0
Frank52	0.45916	0	0.45916	0
Gallery52	0.28920	0.11574	4.43032	169.92289
GaussSeidel104	0.05067	0.00002	0.05067	0.00002
GaussSeidel52	0.06288	0	0.06288	0
Godunov73	1.38710	0.12076	1029.85331	831.05005
Grcar104	0.44183	0	0.44183	0
Grcar52	0.49579	0	0.49579	0
Hatano104	0.23302	0	0.23302	0
Hatano52	0.39576	0	.39576	0
Kahan104	0.05592	0	0.05592	0
Kahan52	0.18601	0	0.18601	0
Landau104	0.28170	0	0.56381	1.00117
Landau52	0.41773	0	1.27260	2.04647
MarkovChain104	0.07692	0	0.07692	0
MarkovChain62	0.04354	0	0.04354	0
OrrSommerfield104	0.07841	0	0.24493	2.12297
OrrSommerfield52	0.04793	0	0.15570	2.24651
SkewLaplacian83	0.01012	0.00127	0.39799	38.36618
Supg42	0.06552	0	0.06552	0
Transient104	0.13731	0.00003	0.30051	1.18856
Transient52	0.11036	0	0.29699	1.69107
Twisted104	0.77183	0	0.77183	0
Twisted 52	0.14935	0	0.14935	0

matrix pairs in which Dumitrescu's method differs from Gu with a relative error of order greater than 10^{-3} (Gallery52, Godunov73, and Companion104), the ratio of the largest and smallest singular values of the corresponding matrices are large ($2.04 \times 10^{18}, 7.07 \times 10^{16}$, and 8.27×10^{6} respectively).

6.2 Performance with Larger Matrices Stemming from PDE Problem

We consider the 1-D advection-diffusion equation

$$z_t(t,x) = -v \cdot \nabla z(t,x) + \mu \Delta^2 z(t,x) + b(x)u(t), \quad 0 < x < 1,$$

$$\text{with sensed output} \qquad y(t) = \int_0^1 c(x) z(t,x) dx, \tag{6.1}$$

$$\text{and boundary conditions } z(t,0) = 0 \;\; \text{and} \;\; z(t,1) = 0.$$

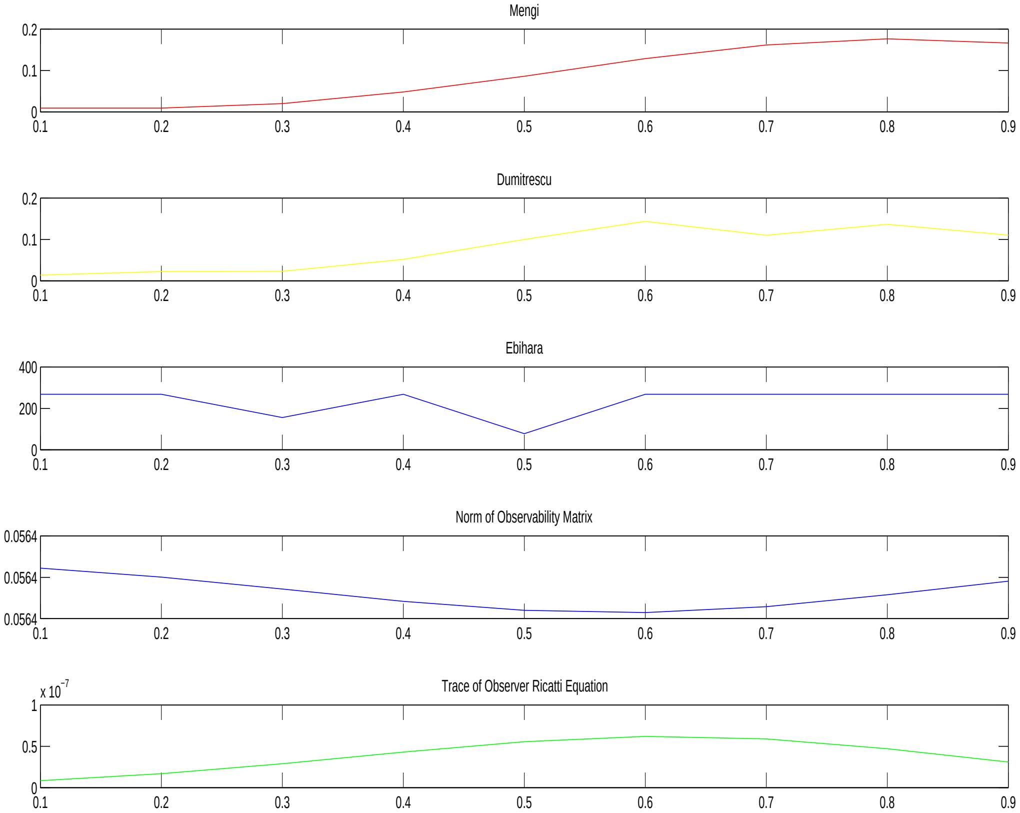
where μ represents the diffusion coefficient and v represents the advection velocity. The system was discretized using quadratic finite element methods, resulting in a linear first-order system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cz(t). \end{aligned} \tag{6.2}$$

Here the number of elements determine the size of A, B, and C. The distance to unobservability for the system was calculated using Ebihara, Dumetriscu, and Mengi to determine the location of the optimal sensor location. Since the distance to unobservability and the distance to uncontrollability are dual problems (see Equations (1.3) and (1.6)), the distance to unobservability for the system is equivalent to the distance to uncontrollability of the pair (A^T, C^T). We computed this distance to uncontrollability for each (single) sensor location for a given (uniformly spaced) mesh size. For example, with ten elements, we computed the distance to uncontrollability nine times using each of the three methods, since excluding the endpoints there are nine possibilities for sensor placement (nodes). The optimal sensor location should be where the distance to unobservability is largest, to reduce the possibility that a small perturbation in input would result in an unobservable system.

Figure 6.1 shows the results for the different methods of calculation with the following parameters:10 elements, ϵ^{-12} (tolerance parameter for SeDuMi), $\mu = 1, v = 10$. Here the independent axis is the location of the observer, and the dependent axis is the resulting distance to observability calculated from the matrices corresponding with that location.

Figure 6.1: Comparison of Methods using Initial Parameters



6.2.1 Ebihara's Method

As the results of running Ebihara's code on the smaller systems from the literature performed rather poorly, with only half of the systems resulting in a distance to uncontrollability agreeing with Gu [19], it was not surprising that Ebihara's method resulted in much larger distances than the other methods. As can be seen in Figure 6.1, the distance to uncontrollability for Ebihara's method (with 9 internal nodes in the mesh) is well over 200, while the other methods calculate distances between 0.1 and 0.2. Ebihara's method's call to SeDuMi most often terminated without convergence (timed-out) leading to longer computation times as well. This pattern held for larger number of nodes as well. For this reason, we decided to discontinue the inclusion of Ebihara's method in experiments with a larger number of nodes.

6.2.2 Agreement of τ as the number of elements increases

Table 6.2: Relationship between distances computed by Mengi's and Dumitrescu's methods

n	v	ℓ	$\frac{\max \text{mengi}}{\max \text{dumi}}$	$\max \frac{\text{mengi}}{\text{dumi}}$	$\min \frac{\text{mengi}}{\text{dumi}}$
9	10	12	0.81391	2.5135	0.66250
10	10	12	1.2315	3.6984	0.65483
11	10	12	0.84540	5.3454	0.63352
12	10	12	1.0989	4.7865	0.64064
13	10	12	1.0831	6.2441	1.0106
14	10	12	1.2028	7.5910	0.86347
15	10	12	1.1958	7.0564	1.0236
16	10	12	1.2010	6.3478	0.99751
17	10	12	1.2280	7.3902	1.0691
18	10	12	1.1987	7.6817	1.0871
19	10	12	1.2161	7.5282	1.0977
20	10	12	1.2645	6.7775	1.0482
21	10	12	1.2645	8.8652	1.1595
22	10	12	1.2724	9.5540	1.2123
23	10	12	1.2570	8.4099	1.1403
24	10	12	1.2812	7.2861	1.0963
25	10	12	1.4042	11.820	1.3394
26	10	12	1.2410	7.9100	1.1241
27	10	12	1.3898	11.485	1.3542
28	10	12	1.3755	10.204	1.2709
29	10	12	1.2436	7.2013	1.1027
30	10	12	1.5976	13.907	1.5209
31	10	12	1.3695	10.417	1.3124
32	10	12	2.0865	19.572	2.0425
33	10	12	1.9382	17.308	1.8613
34	10	12	1.5376	12.009	1.4556
35	10	12	1.8615	16.486	1.8271
36	10	12	2.0115	17.688	1.9307
37	10	12	2.6888	25.456	2.6412
38	10	12	2.6058	24.612	2.5759
39	10	12	1.8117	15.491	1.7802
40	10	12	2.7846	25.043	2.7344

From Table 6.2 we notice the general agreement in the maximum distance to observability. n is the number of internal nodes, v is the velocity parameter, while $\epsilon = 10^{-\ell}$ is the tolerance for SeDuMi. The fourth column, the ratio of the maximum τ calculated by Mengi's method to the maximum τ calculated by Dumitrescu's method, shows that the maximum distances are of the same order. However, the fifth column shows that for every n, there is a node for which Mengi's method results in a smaller value of τ, at times by a factor of 10 or more. The sixth column shows that for all but six values of n, Dumitrescu's method results in a larger value of τ at every node. Thus the conclusion is that Mengi's method tends to give results that are smaller than Dumitrescu's method, but at the nodes of interest, they give results of the same order.

6.2.3 Convergence of the optimal observer location as n increases

Of great interest is whether the optimal location of the observer remains constant as the number of nodes increases. As the mesh is refined, does the optimal sensor location change? Here x is the location of the node resulting in the maximum distance to observability, τ. Table 6.3 shows the results with fixed values of ϵ and velocity μ. It is clear that in both methods, the suggested location of the observer is in the latter half, as is also suggested by the location of the minimum of the trace of the observer Ricatti equation. However, both the Dumitrescu and Mengi methods result in a value of x greater than the 0.55 to 0.6 range suggested by the observer equation. The results of Dumitrescu's method suggest a range of 0.60 to 0.76, while the results of Mengi's method suggest a range of 0.76 to 0.84. The observer locations do not appear to be increasing or decreasing as the value of n increases, but instead stay clustered around a central value. These results are also summarized in Figure 6.2.

It is intuitive that the optimal observer location be somewhere downstream, considering the advection coefficient. From the results, it appears that the the optimal placement is not at the midpoint, or at the very end, but somewhere between the two. Combining the results of both methods it appears the optimal observer location is between 0.7 and 0.8.

Figure 6.2: Sensor location as n increases

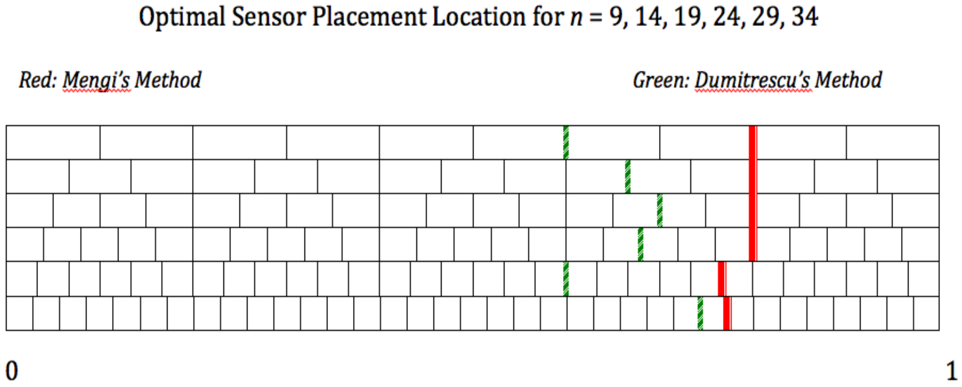


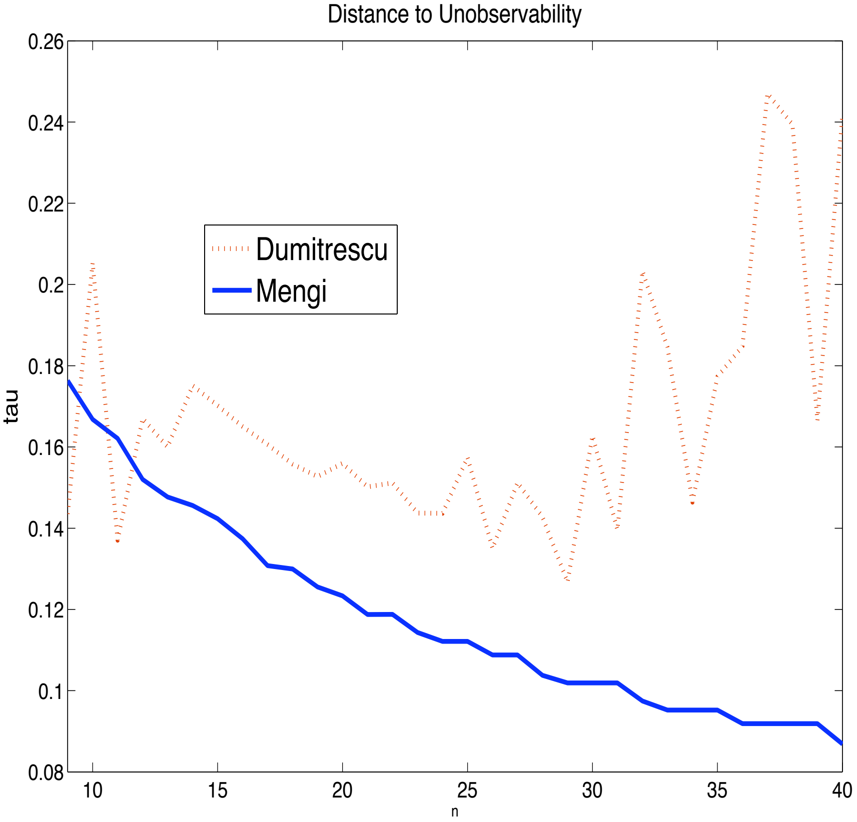
Table 6.3: Dumitrescu's and Mengi's Methods: Convergence with Fixed Velocity

			Dumitrescu's Method			Mengi's Method		
n	v	ϵ	max τ	x	location	max τ	x	location
9	10	12	0.14352	0.60000	0.60000	0.17634	0.80000	0.60000
10	10	12	0.20540	0.72727	0.54545	0.16679	0.81818	0.54545
11	10	12	0.13704	0.83333	0.58333	0.16209	0.83333	0.58333
12	10	12	0.16703	0.76923	0.53846	0.15199	0.76923	0.53846
13	10	12	0.15997	0.78571	0.57143	0.14770	0.78571	0.57143
14	10	12	0.17512	0.66667	0.60000	0.14559	0.80000	0.60000
15	10	12	0.17021	0.68750	0.56250	0.14234	0.81250	0.56250
16	10	12	0.16501	0.70588	0.58824	0.13740	0.82353	0.58824
17	10	12	0.16058	0.66667	0.55556	0.13076	0.83333	0.55556
18	10	12	0.15579	0.73684	0.57895	0.12997	0.78947	0.57895
19	10	12	0.15268	0.70000	0.55000	0.12555	0.80000	0.55000
20	10	12	0.15597	0.71429	0.57143	0.12334	0.80952	0.57143
21	10	12	0.15017	0.72727	0.59091	0.11876	0.77273	0.59091
22	10	12	0.15114	0.73913	0.56522	0.11879	0.78261	0.56522
23	10	12	0.14372	0.75000	0.58333	0.11434	0.79167	0.58333
24	10	12	0.14364	0.68000	0.56000	0.11212	0.80000	0.56000
25	10	12	0.15746	0.73077	0.57692	0.11213	0.80769	0.57692
26	10	12	0.13501	0.74074	0.55556	0.10879	0.77778	0.55556
27	10	12	0.15121	0.75000	0.57143	0.10880	0.82143	0.57143
28	10	12	0.14275	0.72414	0.55172	0.10378	0.79310	0.55172
29	10	12	0.12672	0.60000	0.56667	0.10190	0.76667	0.56667
30	10	12	0.16281	0.74194	0.58065	0.10191	0.77419	0.58065
31	10	12	0.13957	0.75000	0.56250	0.10192	0.81250	0.56250
32	10	12	0.20332	0.72727	0.57576	0.097448	0.78788	0.57576
33	10	12	0.18455	0.73529	0.55882	0.095216	0.76471	0.55882
34	10	12	0.14641	0.74286	0.57143	0.095221	0.77143	0.57143
35	10	12	0.17726	0.72222	0.55556	0.095225	0.80556	0.55556
36	10	12	0.18480	0.75676	0.56757	0.091873	0.75676	0.56757
37	10	12	0.24704	0.71053	0.57895	0.091876	0.76316	0.57895
38	10	12	0.23942	0.71795	0.56410	0.091880	0.79487	0.56410
39	10	12	0.16647	0.72500	0.57500	0.091882	0.82500	0.57500
40	10	12	0.24184	0.75610	0.56098	0.086851	0.75610	0.56098

6.2.4 Convergence of the maximum distance to unobservability as n increases

What should happen to τ as the number of nodes used in the mesh increases? Intuitively, the number of nodes in the mesh has a beneficial effect on the accuracy of the finite-element discretization method, but it has a negative effect on the observability of the system. This is because as the number of nodes increases, the location of observers becomes closer together so that unique input determination becomes more difficult. It is theorized that as the number of nodes in the mesh used for the finite element discretization increases to ∞, then the distance to unobservability of the system should decrease to zero. That is, if there are enough nodes, then the resulting system should be nearly unobservable. Accordingly, as the number of nodes increases, one would expect to see the value of τ decreasing monotonically. Figure 6.3 shows that using Mengi's method, this result (nearly) holds, while Dumitrescu's method shows values of τ that do not appear to increase or decrease monotonically as n increases. Note that the results are plotted using linear interpolation.

Figure 6.3: Comparison of τ as n increases

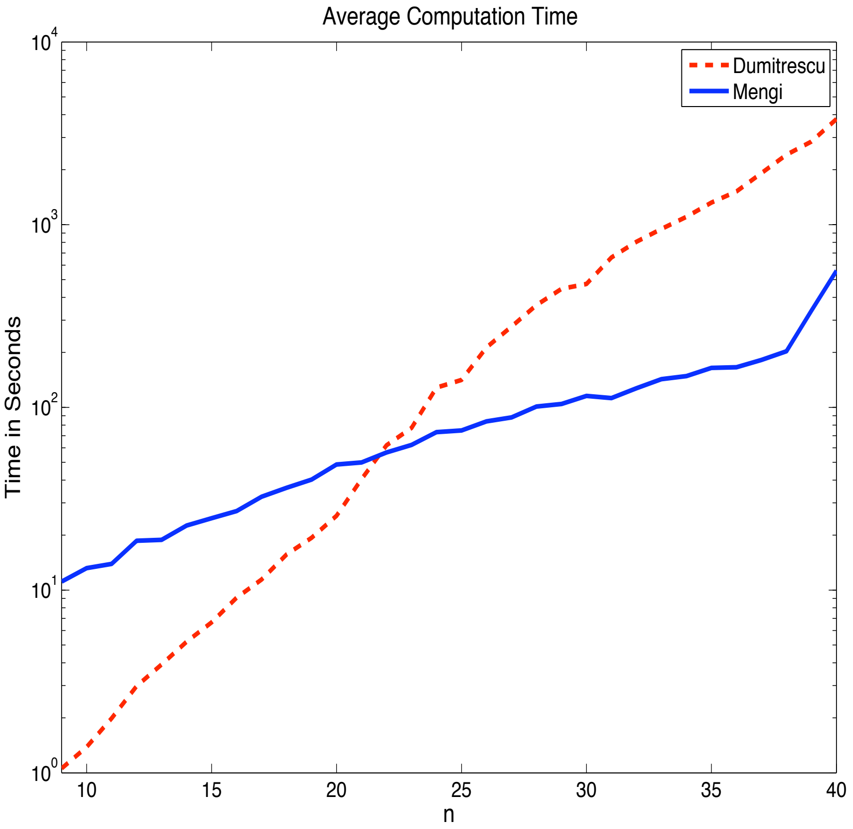


6.2.5 Computation Speed Comparison

As was noted, Ebihara's method was inferior in speed due to its lack of convergence when implementing SeDuMi. SeDuMi would often time-out for even a small number of nodes, and hence the computation time was significantly larger than the other two methods. Here the computation speed was determined by the `tic-toc` function in Matlab, placed before and after each call to the distance to uncontrollability functions. Thus there was a computation speed calculated for each node. Somewhat surprisingly, the method of Dumitrescu was faster, on average, than the method of Mengi for smaller systems, although with larger systems Mengi was faster. This perhaps means that while Mengi's method is order $\mathcal{O}(n^5)$ and Dumitrescu's method is order $\mathcal{O}(n^6)$, the leading coefficient in Mengi's method may be larger or that Dumitrescu's method requires more iterations to converge.

As can be seen from Figure 6.4, Dumitrescu's method is faster on average until there are over twenty nodes, upon which Mengi's method average speed is significantly faster.

Figure 6.4: Comparison of Computation Speed as n increases



The speed in Dumitrescu's method tended to have much less variance than in Mengi's method. In Mengi's method, due to the longer convergence time for nearly uncontrollable systems, the values for which the distance to uncontrollability was small were significantly slower than those where the distance was larger. In Dumitrescu's method, the minimum and maximum computation times were not significantly different. This suggests that after the call is made to SeDuMi, running additional iterations as the distance to uncontrollability decreases does not have as large an impact in the overall computation cost. This was also observed as the computation time did not decrease significantly when the tolerance, ϵ was increased to 10^{-8}. Figure 6.5 shows the comparison in the computation times at each node for each method with 15 nodes and $\epsilon = 10^{-12}$, while Figure 6.6 shows the effect the tolerance increase had on the computation time for Dumitrescu as well as the comparison of both times with Mengi's method. It should be noted that for values of ϵ greater than 10^{-12}, the resulting τ were significantly larger, making it impractical to use such values.

Figure 6.5: Comparison of Computation Variance with 15 nodes and $\epsilon = 10^{-12}$

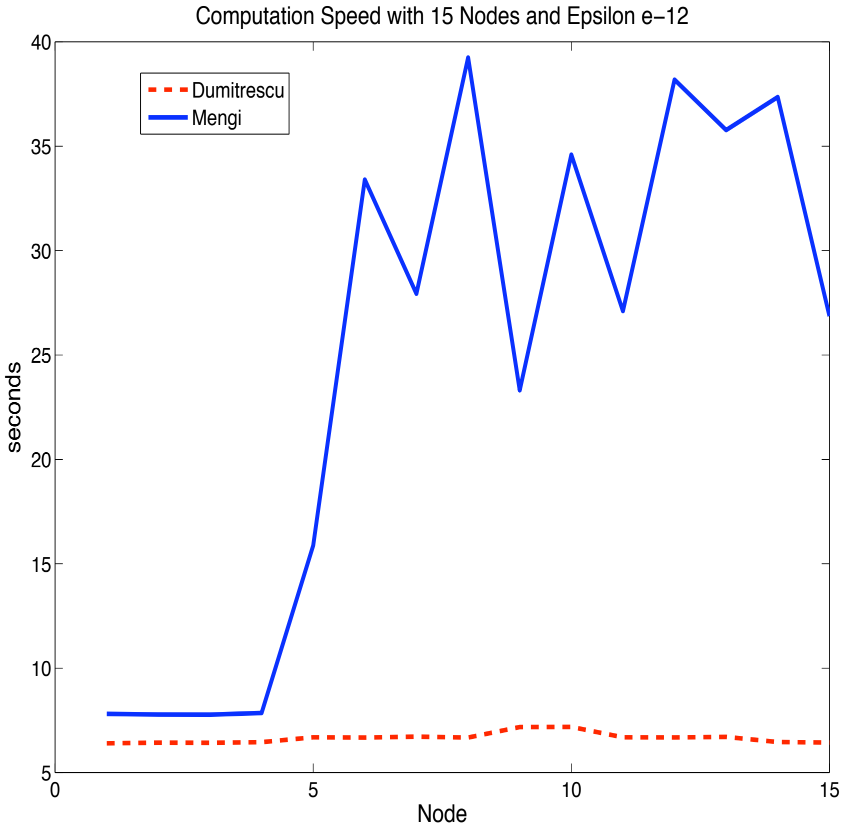
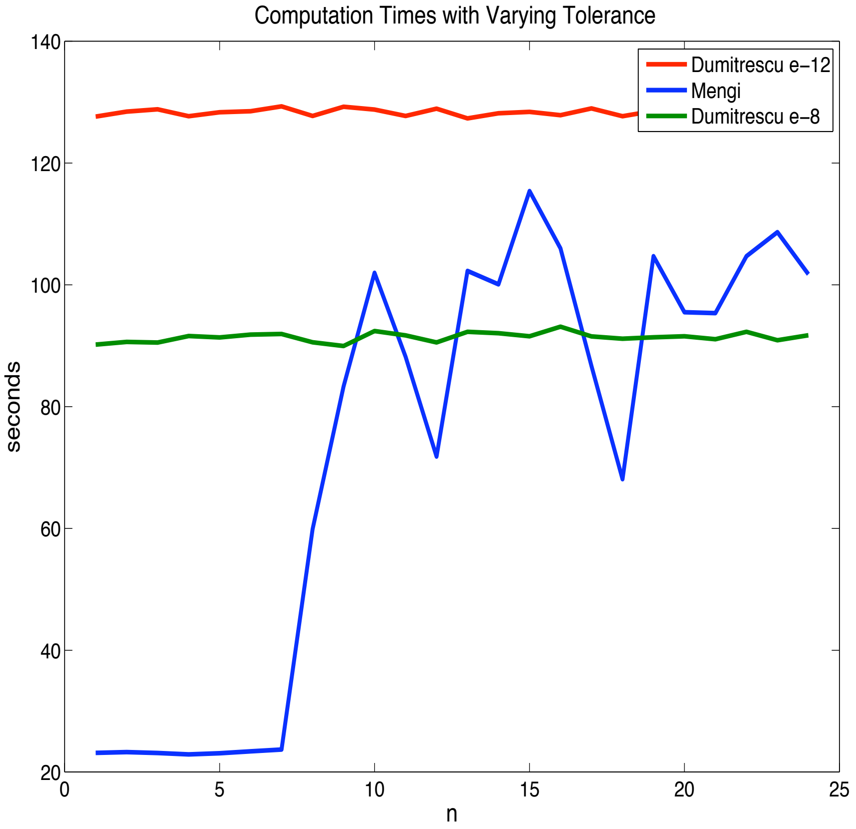


Figure 6.6: Effect of Epsilon on Dumitrescu's Computation Time and Comparison With Mengi (24 Nodes)



6.2.6 Effect of varying the velocity parameter v

Increasing the velocity parameter v, eventually makes the finite-element approximation unstable. One would intuitively expect that as the velocity increased, making it have a larger effect than the advection parameter μ, then the location of the optimal observer location might change further downstream. The system would become less stable and therefore the distance to unobservability would decrease. From Table 6.4, it appears that the change in velocity has a much greater effect on the observer location with Dumitrescu's method than with Mengi's method, as the range of observer locations for Dumitrescu's method was $.45 - .95$, while Mengi's method generally remains constant throughout the change in velocity with values clustered near 0.8. The values of τ actually consistently decreased with velocity parameter of $v = 15$ compared to $v = 10$ or $v = 20$ with Mengi's method, while in Dumitrescu's method there appeared to be little pattern for smaller nodes whle in the larger nodes the value of τ when $v = 20$ was significantly larger. If the results of Mengi's method are accurate, then this is useful because the location of the observer does not vary as the velocity parameter changes.

Table 6.4: Effect of v on τ

		Dumitrescu's Method		Mengi's Method	
n	v	τ	x	τ	x
9.0000	10.000	0.14352	0.60000	0.17634	0.80000
9.0000	15.000	0.12529	0.90000	0.16242	0.80000
9.0000	20.000	0.18196	0.80000	0.23788	0.80000
10.000	10.000	0.20540	0.72727	0.16679	0.81818
10.000	15.000	0.14146	0.81818	0.15730	0.81818
10.000	20.000	0.20189	0.81818	0.23817	0.81818
11.000	10.000	0.13704	0.83333	0.16209	0.83333
11.000	15.000	0.14364	0.83333	0.15267	0.83333
11.000	20.000	0.18887	0.75000	0.23391	0.83333
12.000	10.000	0.16703	0.76923	0.15199	0.76923
12.000	15.000	0.14284	0.84615	0.14841	0.84615
12.000	20.000	0.16932	0.84615	0.23199	0.76923
13.000	10.000	0.15997	0.78571	0.14770	0.78571
13.000	15.000	0.16318	0.85714	0.14178	0.85714
13.000	20.000	0.16375	0.78571	0.23251	0.78571
14.000	10.000	0.17512	0.66667	0.14559	0.80000
14.000	15.000	0.15527	0.86667	0.13938	0.80000
14.000	20.000	0.15700	0.80000	0.22754	0.80000
15.000	10.000	0.17021	0.68750	0.14234	0.81250
15.000	15.000	0.15204	0.87500	0.13340	0.81250
15.000	20.000	0.15807	0.81250	0.22785	0.81250
16.000	10.000	0.16501	0.70588	0.13740	0.82353
16.000	15.000	0.15905	0.76471	0.13045	0.82353
16.000	20.000	0.28528	0.82353	0.22463	0.82353
17.000	10.000	0.16058	0.66667	0.13076	0.83333
17.000	15.000	0.15564	0.66667	0.12594	0.77778
17.000	20.000	0.27849	0.88889	0.22310	0.77778
18.000	10.000	0.15579	0.73684	0.12997	0.78947
18.000	15.000	0.15499	0.78947	0.12601	0.84211
18.000	20.000	0.55888	0.94737	0.22065	0.73684

continued on next page

		Dumitrescu's Method		Mengi's Method	
n	v	τ	x	τ	x
19.000	10.000	0.15268	0.70000	0.12555	0.80000
19.000	15.000	0.15153	0.75000	0.11993	0.85000
19.000	20.000	0.65022	0.45000	0.22079	0.80000
20.000	10.000	0.15597	0.71429	0.12334	0.80952
20.000	15.000	0.17187	0.80952	0.11921	0.80952
20.000	20.000	0.18895	0.80952	0.21742	0.80952
21.000	10.000	0.15017	0.72727	0.11876	0.77273
21.000	15.000	0.15075	0.77273	0.11515	0.81818
21.000	20.000	0.23551	0.81818	0.21709	0.77273
22.000	10.000	0.15114	0.73913	0.11879	0.78261
22.000	15.000	0.15387	0.78261	0.11314	0.82609
22.000	20.000	0.28822	0.65217	0.21718	0.78261

Chapter 7

Conclusions

7.1 Conclusions

This focus of this paper has been upon the usefulness of SDP in solving DTUC problems. The numerical results indicated that the established method using standard eigenvalue solvers was superior, in practice, with a larger problem stemming from a PDE. However, the SDP methods are not without merit. While the SDP method due to Ebihara appears to be impractical for computation, it may be that for more generalized problems this method leads to new algorithms. The sum-of-squares SDP design was computationally feasible. It yielded similar, although inferior, results and performance on the larger matrices, but was faster and yielded same-order errors on the smaller systems. Future research may examine how these algorithms will perform on other large systems, the nature of the error using the sum-of-squares relaxation, and possible ways to combine the established method with the sum-of-squares method to improve performance and reliability.

Appendix

Consider the square matrix block

$$E = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

and assume that the matrices A and D are invertible. Then the Schur complement of block D of the matrix E is given by the square matrix

$$S = A - BD^{-1}C \tag{7.1}$$

and the Schur complement of block A of the matrix E is given by

$$T = D - CA^{-1}B. \tag{7.2}$$

The inverse of E exists so long as both D and S are invertible, and is given by

$$E^{-1} = \begin{bmatrix} S^{-1} & -S^{-1}BD^{-1} \\ -D^{-1}CS^{-1} & D^{-1} + D^{-1}CS^{-1}BD^{-1} \end{bmatrix}. \tag{7.3}$$

Similarly, if both A and T are invertible,

$$E^{-1} = \begin{bmatrix} A^{-1} + A^{-1}T^{-1}CA^{-1} & -A^{-1}BT^{-1} \\ -T^{-1}CA^{-1} & T^{-1} \end{bmatrix}. \tag{7.4}$$

Equating the first block in Equations (7.3) and (7.4) leads to the matrix inversion formula (see [1]) :

Lemma 7.1.1 *Matrix Inversion Formula*

If both A and D are invertible matrices, then

$$(A - BD^{-1}C)^{-1} = A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1}. \tag{7.5}$$

Now suppose additionally that the block matrix E is positive definite, i.e. $C = B^*$. Then the blocks A and D must each be positive definite, and since that the eigenvalues of E must be positive and hence their reciprocals, the eigenvalues of E^{-1}, must be positive as well. Thus the diagonal blocks in (7.3) and (7.4) must each be positive definite. In particular, T^{-1}, must be positive definite, so T must be as well.

This leads to the following characterization of positive definiteness, referred to in [5].

Lemma 7.1.2 *The matrix E defined above is positive definite if and only if both A and $T = D - B^*A^{-1}B$ are positive definite.*

Bibliography

[1] K.M. Abadir, J.R. Magnus, and P.C.B. Phillips. Matrix Algebra. Econometric Excercises Vol. 1, 2005.

[2] B. D. O. Anderson. A system theory criterion for positive real matrices. *SIAM Journal on Control*, 5:171, 1967.

[3] P.A. Bliman. A Convex Approach to Robust Stability for Linear Systems with Uncertain Scalar Parameters. *SIAM Journal on Control and Optimization*, 42(6):2016–2042, 2003.

[4] D.L. Boley and W.S. Lu. Measuring how far a controllable system is from an uncontrollable one. *IEEE Trans. Autom. Control*, 31(3):249–251, 1986.

[5] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory.* Society for Industrial Mathematics, 1994.

[6] S.P. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge Univ Pr, 2004.

[7] J.V. Burke, A.S. Lewis, and M.L. Overton. Pseudospectral components and the distance to uncontrollability. *SIAM Journal on Matrix Analysis and Applications*, 26(2):350–361, 2005.

[8] R. Byers. Detecting nearly uncontrollable pairs. In *Numerical methods proceedings of the international symposium MTNS-89*, volume 3, pages 447–457. Citeseer, 1990.

[9] M.D. Choi, T.Y. Lam, and B. Reznick. Sums of squares of real polynomials. In *K-Theory and Algebraic Geometry: Connections with Quadratic Forms and Division Algebras (B. Jacob, A. Rosenberg, eds.), Proc. Symp. Pure Math*, volume 58, pages 103–126, 1995.

[10] E. de Klerk, C. Roos, and T. Terlaky. A short survey on semidefinite programming. In W. K. Haneveld, O. J. Vrieze, and L. C. M. Kallenberg, editors, *Ten Years LNMB*, 1997.

[11] B. Dumitrescu. `http://schur.pub.ro/Idei2007/`, April 2008.

[12] B. Dumitrescu, B.C. Sicleru, and R. Stefan. Computing the controllability radius: a semidefinite programming approach. *IET Control Theory and Applications*, 3:654–660, 2009.

[13] Y. Ebihara. Computing the distance to uncontrollability via LMIs: Lower bound computation with exactness verification. *Systems & Control Letters*, 57(9):763–771, 2008.

[14] Y. Ebihara. An elementary proof for the exactness of (D,G) scaling. In *American Control Conference, 2009. ACC'09.*, pages 2433–2438. IEEE, 2009.

[15] Y. Ebihara and T. Hagiwara. Computing the Distance to Uncontrollability via LMIs: Lower and Upper Bounds Computation and Exactness Verification. In *2006 45th IEEE Conference on Decision and Control*, pages 5772–5777, 2006.

[16] R. Eising. Between controllable and uncontrollable. *Systems & control letters*, 4(5):263–264, 1984.

[17] M. Gao and M. Neumann. A global minimum search algorithm for estimating the distance to uncontrollability. *Linear Algebra and its Applications*, 188:305–350, 1993.

[18] M. Gu. New methods for estimating the distance to uncontrollability. *SIAM Journal on Matrix Analysis and Applications*, 21(3):989–1003, 2000.

[19] M. Gu, E. Mengi, M.L. Overton, J. Xia, and J. Zhu. Fast methods for estimating the distance to uncontrollability. *SIAM Journal on Matrix Analysis and Applications*, 28(2):477–502, 2007.

[20] M.L.J. Hautus. Controllability and observability conditions of linear autonomous systems. *Ned. Akad. Wetenschappen, Proc. Ser. A*, 72:443–448, 1969.

[21] F. Jarre. Interior-point methods for classes of convex programs. *Interior point methods of mathematical programming*, pages 255–296, 1996.

[22] V. Klema and A. Laub. The singular value decomposition: Its computation and some applications. *IEEE Transactions on Automatic Control*, 25(2):164–176, 1980.

[23] J. Lofberg. Yalmip wiki. `http://users.isy.liu.se/johanl/yalmip/`, May 2009.

[24] D.G. Luenberger. *Optimization by vector space methods.* Wiley-Interscience, 1997.

[25] G. Meinsma, Y. Shrivastava, and M. Fu. A dual formulation of mixed μ and on the losslessness of (D, G)-scaling. *IEEE Trans. Aut. Control*, 42(7):1032–1036, 1997.

[26] E. Mengi. Distance to uncontrollability of a matrix pair. `http://www.cs.nyu.edu/mengi/robust_stability/dist_uncont.html`.

[27] E. Mengi. *Measures for robust stability and controllability.* PhD thesis, New York University, September 2006.

[28] G.S. Miminis. Numerical algorithms for controllability and eigenvalue allocation. Master's thesis, School of Computer Science, McGill University, 1981.

[29] Y. Nesterov and A. Nemirovsky. A general approach to polynomial-time algorithms design for convex programming. *Report, Central Economical and Mathematical Institute, USSR Academy of Sciences, Moscow*, 1988.

[30] Y. Nesterov and A. Nemirovsky. Interior point polynomial methods in convex programming. *Studies in applied mathematics*, 13, 1994.

[31] J. Nocedal and S.J. Wright. *Numerical optimization.* Springer, 2000.

[32] A. Packard and J. Doyle. The complex structured singular value. *Automatica*, 29(1):71–109, 1993.

[33] C. Paige. Properties of numerical algorithms related to computing controllability. *IEEE Transactions on Automatic Control*, 26(1):130–138, 1981.

[34] P.A. Parrilo. Semidefinite programming relaxations for semi-algebraic problems. *Mathematical Programming*, 96(2):293–320, 2003.

[35] A. Rantzer. On the Kalman–Yakubovich–Popov lemma. *Systems & Control Letters*, 28(1):7–10, 1996.

[36] CW Scherer. LMI relaxations in robust control. *European Journal of Control*, 12(1):3–29, 2006.

[37] K. Sivaramakrishnan. *Linear Programming Approaches to Semidefinite Programming Problems.* PhD thesis, Rensselaer Polytechnic Institute, July 2002.

[38] J.F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1):625–653, 1999.

[39] J.F. Sturm. Sedumi. `http://sedumi.ie.lehigh.edu/`, June 2010.

[40] L. Vandenberghe and V. Balakrishnan. Algorithms and software for LMI problems in control. *IEEE Control Systems Magazine*, 17(5):89–95, 1997.

[41] L. Vandenberghe and S.P. Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.

[42] P. Wolfe. A duality theorem for nonlinear programming. *Quart. Appl. Math*, 19(3):239–244, 1961.