Methods for Optimization-based Fixed-Order Control Design

Eric Bengt Beran

Department of Mathematics and
Department of Automation

Technical University of Denmark

Version 0.40
September 18, 1997

Preface

This thesis is written as a partial fulfillment of the requirement for the Ph.D. degree from the Technical University of Denmark. The research was sponsored by the Danish Research Council under the research program "Robust and Reliable Control" lead by Mogens Blanke, University of Aalborg, Denmark, and the Technical University of Denmark. The research was carried out mostly at the Department of Mathematics and Department of Automation, Technical University of Denmark, but also at the Information Systems Lab., Stanford University, California.

Related presentations and Publications

Parts of the work presented in this thesis has been presented at the following conferences or workshops:

EURACO workshop 1995: The EURACO workshop on Recent Results in Robust and Optimal Control, 1995 in Florence, Italy.

IFAC 1996: The International Federation on Automatic Control World Congress, 1996 in San Francisco, California.

CDC 1996: The Conference on Decision and Control, 1996, in Kobe, Japan.

ACC 1996: The American Control Conference, 1997, in Albuquerque, New Mexico.

ECC 1997: The European Control Conference, 1997, in Brussels, Belgium.

Generally the thesis extends the results presented in

1. E. Beran and K. Grigoriadis. A combined alternating projection and semidefinite programming algorithm for low-order control design. In *IFAC 96, San Francisco*, volume C, pages 85–90, July 1996.

2. E. Beran. Induced Norm Control Toolbox (INCT). In *Proceedings of the Conference on Decision and Control*, Kobe, Japan, Dec. 1996.

3. E. Beran and K. Grigoriadis. Computational issues in alternating projection algorithms for fixed-order control design. In *Proceedings of the American Control Conference*, Albuquerque, New Mexico, June 1997.

4. E. Beran, L. Vandenberghe, and S. Boyd. A global BMI algorithm based on the generalized Benders decomposition. In *Proceedings of the European Control Conference*, Brussels, Belgium, July 1997.

5. K.M. Grigoriadis and E. Beran. Alternating projection algorithms for linear matrix inequalities problems with rank constraints. In L. El Ghaoui and S.-I. Niculescu, editors, *Recent advances on LMI methods in Control*. SIAM, 1997.

Target group

This thesis is aimed at researchers in both control theory and optimization. To understand chapter 2-3 and 6 the reader should have a background in control theory on master student level, and a good knowledge of linear algebra and a basic background in functional analysis, the same is sufficient to read chapter 4-5. However, chapter 4-5 and should also be understandable for researchers with a background in optimization.

Acknowledgements

I will especially like to thank my advisors, Dr. Hans Henrik Niemann, Prof. Martin Philip Bendsøe, and Prof. Jakob Stoustrup.

During my visit to the Information Systems Lab. at Stanford University in the spring of 1996 I had long discussions with Prof. Stephen Boyd and Dr. Lieven Vandenberghe. The working environment they have created has been very inspiring, and I enjoyed my stay.

When I attended my first conference, the Conference on Decision and Control in Orlando, Florida, I was introduced to Prof. Karolos M. Grigoriadis from the University of Houston. Since then it has been a pleasure to talk and work with Prof. Grigoriadis at series of workshops and our mutual visits.

During my stay at Stanford University I had a lot of helpful discussions with Anders Hansson, Shao Po Wu, David Banjerdpongchai, Michael C. Grant, and Constantinos Maglaras.

I would also like to thank the following people for interesting discussions and social activities: Prof. Robert Skelton, Prof Tetsuya Iwasaki, Laurent El Ghaoui, Prof. Tuan, Prof Safonov, Dr. Mesbahi, Dr. Goh, Prof. Tempo. Also a thank to the vast amount of young researchers that I met during the conferences for making the conference a social as well as a technical event, especially Robert van der Geest.

At the Technical University of Denmark a number of persons have been to great help. First of all I would like to thank Marc Cromme, whos help, and encouragement have been magnificent. I would also like to thank Uffe Høgsbro Thygesen, Mike Rank for interesting debates and talks.

Cooperation with various people at the Department of Automatic Control, Lund Institute of Technology. Especially the Lund-Lyngby days in Control have been very inspiring. A thanks to Ulf Jönsson, Lennart Andersson, Mikael Johansson, and Anders Rantzer for discussions and help.

Also a thanks to the people who have help me with the computer problems, especially Ole Ravn from the Department of Automation.

A thanks to all my friends for their help and encouragement.

Technical comments

This text is typeset in LaTeX, with a number of packages. The 'mathptm' package was used to provide the Times Roman font. MATLAB 4.2c and MATLAB 5.1.xx have be used to run a most of the examples. The SDP problems have been solved using the SP package by Boyd and Vandenberghe.

Dansk Resumé

I de seneste år..

Contents

List of Figures

List of Tables

Symbols

The notation in this thesis is fairly standard:

Number spaces

$\mathbb{R}$	The real numbers.
$\mathbb{R}^+$	The nonnegative real numbers.

Vector spaces

$\mathbb{R}^n$	The n-dimensional vector space of real numbers.
$\mathbb{R}^n_+$	The positive orthant.

Matrix spaces

$\mathbb{R}^{n\times m}$	The $n \times m$ dimensional matrix space of real numbers.
$\mathcal{S}^n$	The set of symmetric matrices in $\mathbb{R}^{n\times n}$: $\left\{X \mid X = X^T \in \mathbb{R}^{n\times n}\right\}$.

Linear Algebra

A^T	The transpose of A.
$\operatorname{im} A$	The image of A.
$\ker A$	The kernel of A.
$\lambda(A)$	The eigenvalues of A.
$A^\perp$	The left annihilator of A. i.e. $\operatorname{im} A = \ker A^\perp$, and $A^\perp A^{\perp T} > 0$
$\langle A, B\rangle$	Inner product.
$\operatorname{Diag}(A_1, A_2, \cdot, A_m)$	$\begin{bmatrix} A_1 & 0 & \dots & 0 \\ 0 & A_2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & A_m \end{bmatrix}$

Notation

Consider a matrix $X \in \mathbb{R}^{m\times n}$. The operator $\operatorname{Vec}: \mathbb{R}^{m\times n} \to \mathbb{R}^{mn}$ stacks the columns of X on top of each other, that is

$$\operatorname{Vec}(X) \triangleq \operatorname{Vec}\left(\left\{x_{ij}\right\}\right) \triangleq \begin{bmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{m1} \\ x_{12} \\ \vdots \\ x_{n2} \\ \vdots \\ x_{nn} \end{bmatrix}. \tag{0.1}$$

The Vec-operator is a bijective map, when the number of rows and columns are fixed, and therefore it has an inverse operator $\operatorname{Vec}: \mathbb{R}^{mn} \to \mathbb{R}^{m\times n}$, with the properties,

$$X = \operatorname{Vec}^{-1}(\operatorname{Vec}(X)), X \in \mathbb{R}^{m\times n}$$
$$x = \operatorname{Vec}(\operatorname{Vec}^{-1}(x)), x \in \mathbb{R}^{mn}$$

We define the set of symmetric matrices $\mathcal{S}^n$, as

$$\mathcal{S}^n \triangleq \left\{X : X \in \mathbb{R}^{n\times n}, X = X^T\right\} \quad .$$

We will often need an operator that takes a symmetric matrix in $\mathcal{S}^n$, and produces a vector consisting of the entries in the lower triangular part. For $X = X^T \in \mathbb{R}^{n\times n}$ define the operator

$$\mathrm{SVec}(X) \triangleq \mathrm{SVec}\left(\{x_{ij}\}\right) \triangleq \begin{bmatrix} x_{11} \\ \vdots \\ x_{n1} \\ x_{22} \\ \vdots \\ x_{n2} \\ \vdots \\ x_{nn} \end{bmatrix} \tag{0.2}$$

where $\mathrm{SVec}(X) \in \mathbb{R}^{\frac{n(n+1)}{2}}$. We note that $\mathrm{SVec}: \ \mathcal{S}^n \ \to \ \mathbb{R}^{\frac{n(n+1)}{2}}$ is a bijective map, and therefore it has an inverse. We denote this inverse operator by SVec^{-1}, and note that

$$\begin{aligned} X &= \mathrm{SVec}^{-1}(\mathrm{SVec}(X)), X \in \mathcal{S}^n \\ x &= \mathrm{SVec}(\mathrm{SVec}^{-1}(x)), x \in \mathbb{R}^{\frac{n(n+1)}{2}} \end{aligned}$$

The operator $y = \mathrm{SVec}(Y)$ takes a symmetric matrix and gives a vector $y \in \mathbb{R}^{(n+1)n/2}$, which gives a parameterization of $Y \in \mathcal{S}^n$ as $\mathrm{SVec}^{-1}(y), y \in \mathbb{R}^{(n+1)n/2}$. A basis $\{Y_i\}$ for the set of symmetric matrices can therefore be constructed as $Y_i = \mathrm{SVec}^{-1}(\{y_i = 1\}), i = 1, \cdots, (n+1)n/2$.

A *left annihilator* of a matrix $B \in \mathbb{R}^{m\times n}$ with *deficient row rank* $\underline{m} < m$ is a matrix $B^\perp \in \mathbb{R}^{\underline{m}\times m}$ with full row rank and im $B = \ker B^\perp$. The left annihilator is not unique, since if $B^\perp$ is a left annihilator to B, then so is $2B^\perp$.

Acronyms

AP	Altenating Projection
BB	Branch and Bound
BMI	Bilinear Matrix Inequality
GEVP	Generalized Eigenvalue Problem
LMI	Linear Matrix Inequality
LHBB	Lagrangean House Branch and Bound
LP	Linear Programming
LPBB	Lagrangean Pyramid Branch and Bound
NP	Non Polynomial
RBCBB	Relaxed Bilinear Connection Branch and Bound
SDP	Semidefinite Programming

Chapter 1

Introduction

1.1 Overview

In this thesis we deal with some problems arising in control and the associated optimization techniques customized to solve these problems. The systems in control we consider are all linear and can be written in state-space form. We search for a controller such that given specifications are fulfilled. As specifications on the system itself we consider stabilization and performance. We also consider constraints on the controller, especially we consider controllers with a specific number of states (control order). We seek to find these controllers by using different types of algorithms exploiting recent advanges in linear matrix inequalities.

1.1.1 Linear matrix inequalities

In recent years linear matrix inequalities (LMIs) have gained significant interest in the control society. There are two reasons for this; First of all, LMIs can due to new optimization techniques be solved efficiently. Secondly, LMIs offers a general framework for the formulation of problems arising in control.

The interest from the control community the optimization problem associated with LMIs has helped to push semidefinite programming (SDP) to become a major research topic in the optimization society. A synergy between control and optimization researchers is building up. The better semidefinite programming problems can be solved, the more problems in control can be solved using SDP, which induce more need for research in optimization of SDP's and so forth.

Today SDP software is only reliable for small[1], and medium size problems[2]. However, SDP has many similarities with linear programming (LP) that because of its application in operational research has received extensive interest during the last 30 years. For linear programming there is very efficient software on the market that can solve problems with ten thousands of variables and thousands of constraints. Connections between SDP and LP give hope for more efficient solvers for SDP problems in the near future.

The main reason for the efficiency of modern SDP algorithms is in the convexity of the problem. In optimization it is more important whether a problem is convex or non-convex than linear or not linear. Many problems in control can be formulated as SDP problems, thus being convex. However, just a small change in the control requirements and the problem becomes non-

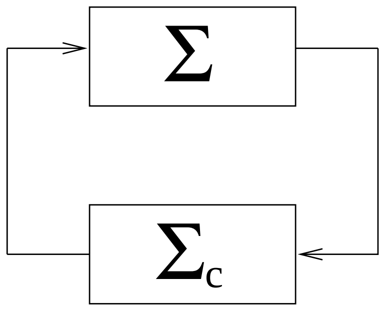
[1]Small problems: 5-10 variables with 1-5 simple (5 by 5 matrices) LMI constraints
[2]medium problems: 10-100 variables with 5-10 medium (10 by 10 matrices) LMI constraints

convex. In this context we shall consider some of these non-convex problems and derive solvers that use SDP among other techniques.

1.1.2 Control design and matrix inequalities

The number of problems arising in control that can be formulated in terms of LMIs are today very extensive. A large number of results have been published over the last 5 years. An overview is given in the book by Boyd et al on "Linear matrix inequalities in systems and control", [BGFB94], and the book by Skelton, Iwasaki and Grigoriadis on "A unified algebraic approach to Control Design", [SIG97].

The typical problem in control that can be formulated in terms of LMIs have the following form. Let a plant Σ and a controller Σ_c be given on state-space form where both the plant and controller is uniquely defined by a set of systems matrices.



The controller measure the output from the plant and uses the measurement plus additional information to compute an actuator input for the plant. The connected system consisting of the plant and controller is called the closed loop system, see figure 1.1.2, where under mild conditions, the closed loop system matrices can be formulated as an affine combination of the plant matrices and the controller matrices.

The goal is now to design a controller such that the closed loop system has a desired behavior. Certain specifications for the behavior of the closed loop system can be formulated as an LMI condition on the closed loop matrices.

Combining the LMI condition on the closed loop system with the affine combination of the plant and controller matrices, conditions for the existence of a controller can be given in terms of a bilinear matrix inequality (BMI). The BMI is not a convex constraint, but it offers a very general framework.

The BMI formulation of the problem can be convexified to an LMI by relaxing restrictions on the number of states in the controller, also called the control order. The control problem can now be written as a set of coupled linear matrix inequalities. The control order appears as the rank of a matrix, which, if constrainend, renders the problem non convex.

We are in this context generally interested in developing algorithms for solving fixed-order control problems. Both the rank formulation and the BMI formulation offer a framework for fixed order control design.

1.1.3 Fixed order design algorithms

It has been shown that certain problems related to the fixed-order control problems are NP hard, see [FL97, BT95]. That is the computational time grows faster than any polymonial in the number of variables. Even though it has not been proven that the fixed-order control problems are NP hard, the above results indicate that the problems might be NP hard. In spite of this compressing the control order might be possible in some cases.

Algorithms have been developed by several researchers for compressing the control order exploiting the rank formulation in connections with LMIs. The alternating projection algorithm [GS94, GS96] formulates the LMI conditions and the rank constraint as feasibility sets and seek by alternating projections on these sets a solution in the intersection of these sets. The balancing algorithm [GOA95] introduces an attracting function that can be used to minimize the rank and consequently the control order. Mesbahi and Papavassilopoulos have, under some condition, shown that rank-minimization problems can be solved to global optimality [MP97a], however in terms of the fixed-order control problems these condition are not fulfilled but can be approximated. Other methods have been presented to reduce the control order, but are not based on the rank formulation [IS95, Iwa97b], but rather on a various other formulations. In [IS95] an approach called the X-Y centering algorithm tries to solve the fixed order control problem by finding a matrix belonging to one set and its inverse to another. An algorithm can also be devised by using two dual formulations of the fixed order control problems [Iwa97b].

In this thesis we especially present a method based on the alternating projection scheme, which exploits the efficiency of semidefinite programming. However, the theoretical background for the algorithms [IS95, Iwa97b] will also be given.

1.1.4 Global optimization of bilinear matrix inequalities

The bilinear matrix inequality (BMI) was introduced in control design by Safonov and his research group, see [SGL94]. Since then several algorithms for solving BMIs have been proposed. Some methods have been based on branch and bound techniques [LW66] and these include a global optimization approach by Goh, Safonov and Papavassilopoulos [GSP95], and a difference convex (D.C.) method by Tuan, Hosoe and Tuy [THT96]. Cone programming techniques have been applied by Mesbahi and Papavassilopoulos [MP96] to the BMI feasibility problems.

It has been shown by Toker and Ozbay, that a known NP hard problem, can be reduced to a bilinear matrix inequality problem, [TO95] and thus indicating that the BMI problems are very hard to solve. However, the general framework that the bilinear matrix inequalities offer justify extensive research in solving these problems.

A family of algorithms is presented here to solve the BMI problem to global optimality. The family is based on Benders decomposition [Ben62, Geo72], branch and bound techniques, and Lagrangian duality [Ber95].

1.2 Outline

The setup in this thesis is the following. In the first chapter we formulate specifications for control design that can be reformulated in terms of linear matrix inequalities. In the next chapter we derive convex and non convex results for the existence of controllers of free or fixed order. In chapter 4 we present a method based on alternating projection techniques, which tries to sole the rank minimization problem. In the next chapter we present methods that solves the BMI optimization problem globally.

Part I

Problem formulation and background material

Chapter 2

Specifications for control design

The goal in control design is to construct a *controller* Σ_c that connected to a *system* Σ provides a desired behavior of the connected system (called *closed loop*), see the setup in figure 2.1. In this chapter we will discuss a set of specifications for the behavior of the closed loop system, and we will examine the effect of constraints on the controller. Combining the system, constraints on the controller, and specifications for the closed loop, we arrive at a problem which we will call a *control problem*.

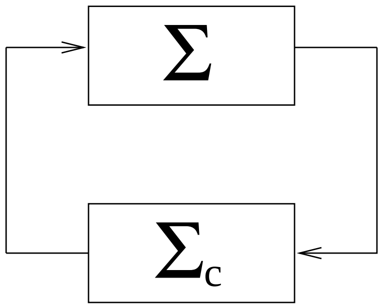


Figure 2.1: Closed loop system.

As possible specifications for the closed loop system we will examine stability and performance. This is a narrow selection from a wide variety of specifications. Our goal is not to provide a complete library of specifications, but merely to use the selected control problems to explain the main approach. In fact the solution to the problem with stability specification turns out to have the same form as the solution to a problem with a performance objective. Our main emphasis is to study the complexity of control problems when constraints on the controller are added. Especially we will consider the number of necessary states (order) of the controller, but we will also consider a bound on the gains of the controller.

The control problems presented here are formulated with general specifications, which are then reformulated in terms of linear matrix inequalities (LMIs). The term LMI was introduced by Willems in 1971 [Wil71]. A historical overview can be found in a book by Boyd et al. on linear matrix inequalities in systems and control [BGFB94]. In this chapter we exploit mainly the second method of Lyapunov [Lya47], that applied to our problems lead to an LMI formulation.

The approach is the following. First we develop analysis result in terms of LMIs. Next we take a look at the open loop system and the controller. It is shown that the closed loop system is an affine function of a certain controller parameter. A discussion of the possible constraints on the controller is done, and in the end of the chapter follows the selected control problems.

2.1 Control objectives

In this section we will consider two specifications for controller design, namely stability and performance. These two types of specification are only meant as examples. A broader class of specifications can be formulated in a similar fashion, see for instance [BGFB94] or [SIG97]. However these simple specifications do lead to control problems that feature the same kind of structure as the broader class. Later we want to study elements of the formulation that lead to efficient computation.

We emphasize that we only consider linear time-invariant systems that can be written in state-space form:

$$\begin{bmatrix} \dot{x} \\ z \end{bmatrix} = \begin{bmatrix} A & B_w \\ C_z & D_{zw} \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix}, \ x(0) = 0$$

where A, B_w, C_z, D_{zw} are fixed matrices of appropriate sizes. The vector x is a function of time and consists of the states of the system.

2.1.1 Stability

For a system to be stable we will require that without external input the states x decays to zero when time goes to infinity, or in other words the system returns to an equilibrium (state of balance) after any disturbance. To study this we consider the autonomous system

$$\dot{x} = Ax, \ x(0) = x_0 \ , \tag{2.1}$$

where $A \in \mathbb{R}^{n\times n}$ is given data, and $x \in \mathbb{R}^n$ are the states of the system.

We will say that the system 2.1 is **stable** if all possible trajectories of the system are bounded and decay to zero as time goes to infinity, that is

$$x(t) \to 0 \text{ as } t \to \infty.$$

Or more precisely $\forall \varepsilon > 0, \exists K : \ ||x(t)|| < \varepsilon, \ \forall t > K$. Often this kind of stability is called *asymptotic stability*.

In order to investigate the stability of the system, the following lemma is useful, see for instance definition 3.2 and section 3.8 in [ZDG95, Lya47, KB60],

Lemma 2.1 (Lyapunov stability) *The following is equivalent*

i) The system (2.1) is stable.

ii) All eigenvalues of A are in the open left half plane of the complex plane, i.e. Re $\lambda(A) < 0$.

iii) There exists a symmetric positive definite matrix Y such that $A^TY + YA$ *is negative definite.*

In lemma 2.1 the third condition is stated as a **linear matrix inequality** (abbreviated LMI). A linear matrix inequality has the general form

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i \geq 0, \tag{2.2}$$

where $x \in \mathbb{R}^m$ is the variable and $F_i \in \mathcal{S}^n,\ i = 0,\cdots,m_x$ are known data. $\mathcal{S}^n \subset \mathbb{R}^{n\times n}$ denotes the set of symmetric matrices of size n by n. Note that the map F from the variables x into the set of symmetric matrices $\mathcal{S}^n$, $F : \mathbb{R}^m \to \mathcal{S}^n$, is an affine function. The inequality sign in (2.2) is with respect to positive semidefinite matrices, or equivalently we require $F(x)$ to be *positive semidefinite*. A matrix $M \in \mathcal{S}^n$ is said to be *positive semidefinite*, if

$$v^T M v \geq 0, \forall v \in \mathbb{R}^n \ , \tag{2.3}$$

which is equivalent to saying that all eigenvalues of M are non-negative. Recall that a real symmetric matrix has all its eigenvalues on the real axis. If the inequality sign in equation (2.3) is strict we say the matrix M is positive definite and (2.2) is called a **strict linear matrix inequality**.

A very important property of linear matrix inequalities is that they define a convex constraint, that is, the set of x that fulfills the constraint is convex. A set $\mathcal{C}$ is convex if the line between any two points in the set is fully contained in the set, i.e. $x_1, x_2 \in \mathcal{C} \implies \alpha x_1 + (1-\alpha)x_2 \in \mathcal{C}, \forall \alpha \in [0;1]$. To see that the LMI is convex in x, consider x_1, x_2 with $v^T F(x_1) v \geq 0,\ v^T F(x_2) v \geq 0$ for all $v \in \mathbb{R}^n$. Then due to the affine structure of F,

$$v^T\left(F(\alpha x_1 + (1-\alpha)x_2)\right)v = \alpha v^T F(x_1) v + (1-\alpha) v^T F(x_2) v \geq 0 \ , \tag{2.4}$$

for $\alpha \in [0;1]$ and for all $v \in \mathbb{R}^n$.

Example 2.1 (Formulation of Lyapunov inequality as an LMI) *The third condition of lemma 2.1 can with the above introduction of positive definite be written as, $Y > 0, A^T Y + YA < 0$, which is indeed and LMI. To transform this LMI into the general form (2.2), we need a basis for $Y \in \mathcal{S}^n$, that is a basis for the set of symmetric matrices. Let $\{Y_i\}$, $i = 1,\cdots,(n+1)n/2$ be such a basis. The LMI $Y > 0, A^T Y + YA < 0$ can now be transformed into the form (2.2) by introducing*

$$F_i = \begin{bmatrix} Y_i & 0 \\ 0 & A^T Y_i + Y_i A \end{bmatrix} \text{ and } F_0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

The block diagonal structure in the above LMI is typical, when considering multiple constraint.

A solution to the system (2.1) is in general composed of a set of trajectories on the form $x(t) = t^q e^{(-\alpha+\omega j)t}$, where $-\alpha+\omega j$ is an eigenvalue of A. α is the *decay rate* of the trajectory and $\omega/2\pi$ is the frequency of the *oscillations* in the trajectory. In a simple stability specification we do not specify how well the states should be damped, or how fast the oscillations may be. The behavior of the system in terms of decay rate and oscillations are determined by the poles of the system, or in state space form the eigenvalues of A. To control the behavior of the trajectories the designer want to place the poles of the system in specific regions. As a simple example we consider α-stability, i.e. all states decays to zero faster than $e^{-\alpha t}$, or more precisely: there $\exists K > 0$ such that $\forall t > 0,\ \|x(t)\|_2 < K e^{-\alpha t}$. The notation $\|\bullet\|_2$ denotes the standard Euclidean vector norm. The α-stability is equivalent to the real parts of the eigenvalues of A being left of $-\alpha$. From lemma 2.1 it follows that α stability holds if and only if $\exists Y > 0,\ (A+\alpha I)^T Y + Y(A+\alpha I) < 0$. For α fixed the constraint $(A+\alpha I)^T Y + Y(A+\alpha I) < 0$ is an LMI, but we are usually interested in the optimal α. That is we want to solve the problem:

$$\begin{array}{ll} \underset{Y,\alpha}{\text{Maximize}} & \alpha \\ \text{subject to} & Y > 0 \\ & (A+\alpha I)^T Y + Y(A+\alpha I) < 0. \end{array} \tag{2.5}$$

This is a so called *generalized eigenvalue problem* (GEVP) , that is a problem on the form:

$$\begin{array}{ll} \underset{y,\lambda}{\text{Minimize}} & \lambda \\ \text{subject to} & \lambda B(y) - A(y) > 0 \\ & B(y) > 0, C(y) > 0 \end{array} \tag{2.6}$$

where A, B and C are symmetric matrices that are affine functions in y. The problem (2.5) is of the form (2.6), which can be seen by writing it as:

$$\begin{array}{ll} \underset{Y,\alpha}{\text{Minimize}} & (-\alpha) \\ \text{subject to} & (-\alpha)(2Y) - \left(A^T Y + YA\right) < 0 \\ & (2Y) > 0 \ . \end{array} \tag{2.7}$$

As stated earlier α stability means that all poles are left of a line going through $-\alpha$ and parallel to the imaginary axis. A general result for *pole placement* in specific convex regions using LMIs can be found in [CG96].

2.1.2 Performance

In the previous subsection we discussed stability. Stability basically relates to the internal behavior of a system. However, the reaction on disturbances coming from the outside is also very interesting and important. The part of the system where the reaction is of interest is measured by an error output. Consider the setup in figure 2.2 in state-space form

$$\Sigma : \begin{bmatrix} \dot{x} \\ e \end{bmatrix} = \begin{bmatrix} A & B_d \\ C_e & D_{ed} \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix}. \tag{2.8}$$

That is the system has one vector input and one vector output, see figure 2.2. The input $d : \mathbb{R} \to \mathbb{R}^{n_d}$ is used to model disturbances, whereas the error signal $e : \mathbb{R} \to \mathbb{R}^{n_e}$ is used to pick out the essential parts of the system. Both the disturbance signal d and the error signal e are mappings from time into a space of real vectors of appropriate dimension. Internally the system is characterized by the states x, which again is a mapping from time into the space of real vectors $\mathbb{R}^n$.

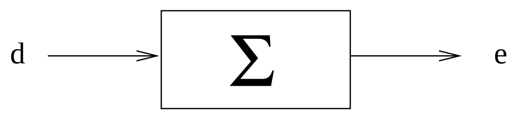


Figure 2.2: The setup for performance analysis.

Both the disturbance and the error are considered to be signals. The mapping from disturbances to errors can be characterized by the convolution operator with kernel $K_{ed}(t) = C_e e^{At} B_d + D_{ed}$, and we will denote this mapping by T_{ed}.

For the system (2.8) we want to specify how well disturbances are damped when seen at the error signal. To formally describe this specification we first take a look at how signals are measured, and then use these measures to formulate different performance measures.

Therefore consider a general signal of the form

$$h : \mathbb{R} \to \mathbb{R}^n,$$

where we throughout this text will assume that any signal is a *piece-wise continuous function*, implying that the function is measurable. To construct different norms on signals we will first take the vector norm at each time instant, and then take another norm on this mapped scalar signal. We consider the following norms for vectors in $\mathbb{R}^n$

$$||v||_1 = \sum_{i=1}^{n} |v_i|\,, \quad ||v||_2 = \sqrt{v^T v}, \text{ and } ||v||_\infty = \max_i |v_i|\,. \tag{2.9}$$

The norms are refered to as 1-norm, 2-norm or Euclidean norm, and ∞-norm or max-norm, respectively.

Applying a vector norm to each time instants of h we get a signal from time into $\mathbb{R}^+$. On this kind of signals we can define an $\mathcal{L}_p$-norm. There are different possibilities, and they are defined in the following way:

$$\begin{aligned} ||h||_{\mathcal{L}_p} &= \left(\int_{-\infty}^{\infty} ||h(t)||_p \, dt \right)^{1/p} \qquad \text{for } p > 0, p \in \mathbb{R} \\ ||h||_{\mathcal{L}_\infty} &= \sup_{t \in \mathbb{R}} |h(t)| \end{aligned} \tag{2.10}$$

We will especially consider the $\mathcal{L}_p$-norms with $p = 1, 2, \infty$. It is fairly standard to refer to $||\bullet||_{\mathcal{L}_2}$ as the *energy* norm. We will also introduce the names *resource* for $||\bullet||_{\mathcal{L}_1}$ and *peak* for $||\bullet||_{\mathcal{L}_\infty}$.

Combining the different vector norms and $\mathcal{L}_p$-norms we can compose various signal norms. Suppose we first take the vector r-norm and then the $\mathcal{L}_p$-norm we get the signal norm,

$$||h(t)||_{\mathcal{L}_{p,r}} = || \, ||h(\bullet)||_r \, ||_{\mathcal{L}_p} \,,$$

where the notation $||h(\bullet)||_r$ means that we take the vector norm at each time instant. Note the notation $\mathcal{L}_{p,r}$ refering to the chosen vector r-norm and the outer $\mathcal{L}_p$-norm. For both the vector norm and the $\mathcal{L}_p$-norm only the numbers one, two and infinity are used here. This leads to 9 possible signal norms. The 5 most commonly used are stated in table 2.1. All the norms have been given names refering to their physical interpretations.

Resource(R):	$\vert\vert h\vert\vert_{\mathcal{L}_{1,1}}$	$= \vert\vert h\vert\vert_R$	$= \int_{-\infty}^{\infty} \sum_{i=1}^{n} \vert h_i(t)\vert \, dt$
Euclidean Resource(ER):	$\vert\vert h\vert\vert_{\mathcal{L}_{1,2}}$	$= \vert\vert h\vert\vert_{ER}$	$= \int_{-\infty}^{\infty} \sqrt{h^T(t)h(t)}\, dt$
Energy(E):	$\vert\vert h\vert\vert_{\mathcal{L}_{2,2}}$	$= \vert\vert h\vert\vert_E$	$= \left(\int_{-\infty}^{\infty} h^T(t)h(t)\, dt \right)^{1/2}$
Euclidean Peak(EP):	$\vert\vert h\vert\vert_{\mathcal{L}_{\infty,2}}$	$= \vert\vert h\vert\vert_{EP}$	$= \sup_{t \in \mathbb{R}} \sqrt{h^T(t)h(t)}$
Peak(P):	$\vert\vert h\vert\vert_{\mathcal{L}_{\infty,\infty}}$	$= \vert\vert h\vert\vert_P$	$= \sup_{t \in \mathbb{R}} \max_i \vert h_i(t)\vert$

Table 2.1: Important signal norms and their names.

We have now defined how we measure signals. The next step towards defining performance is how we will measure the attenuation from disturbances to error. Consider a mapping T_{ed} from disturbances $d: \mathbb{R} \to \mathbb{R}^{n_d}$ to error output $e: \mathbb{R} \to \mathbb{R}^{n_e}$. We will measure the disturbances in one signal norm $||d||_{\mathcal{L}_{p,r}}$ and the error in $||e||_{\mathcal{L}_{q,s}}$. A natural question is: how do a set of disturbances bounded in $\mathcal{L}_{p,r}$ norm influence the error measured in $\mathcal{L}_{q,s}$. The measure is formally defined in

Definition 2.1 (Induced norm) *The $\mathcal{L}_{p,r}$ **to** $\mathcal{L}_{q,s}$ **induced norm** is defined as*

$$||T_{ed}||_{\mathcal{L}_{p,r} \to \mathcal{L}_{q,s}} = \sup_{d \in \mathcal{L}_{p,r}} \frac{||e||_{\mathcal{L}_{q,s}}}{||d||_{\mathcal{L}_{p,r}}}, \;\; d \neq 0$$

If the disturbances are well damped the induced norm is low, while it is high if the damping is poor. In some cases the induced norm is even infinity. In many cases the induced norm can be reformulated as a norm on the convolution operator. This is usually derived by showing that the induced norm is bounded above by some measure on the convolution operator or transfer function, and then proving that there exists a disturbance that achieves this upper bound. In this context we are only treat the induced norms, which lead to computable result in terms of the state-space matrices. We are especially interested in cases where these results are given as linear matrix inequalities.

As a simple example we will consider the induced norm from Energy to Euclidean Peak.

Example 2.2 (Energy to Euclidean Peak Induced Norm) *The Energy to Euclidean Peak induced norm denoted by* $||\bullet||_{E\to EP}$. *A standard result due to Wilson [Wil89] leads to the following result: The* $||T_{ed}||_{E\to EP}$*-norm is less than* γ *if and only if* $D_{ed} = 0$ *and there exists* Y *such that*

$$\begin{gathered} Y > 0 \\ YA^T + AY + B_d B_d^T < 0 \\ C_e Y C_e^T < \gamma^2 I. \end{gathered}$$

Note that if the LMI $YA^T + AY + B_d B_d^T < 0$ *holds, stability is implied since* $B_d B_d \geq 0$ *and therefore* $YA^T + AY < 0$.

Similar results can be derived for the Energy to Peak, Resource to Energy, and Euclidean Resource to Energy induced norms, see [SGC97, Ber94]. For the Euclidean Resource to Euclidean Resource and Euclidean Peak to Euclidean Peak induced norms, see [ANP96].

The same analysis could be done by solving a Lyapunov equation, but the strength of the above formulation is that it defines a convex region of all possible Lyapunov matrices Y, that guarantees the Energy to Euclidean Peak induced norm to be less than γ.

In table 2.2 is a table over possible induced norms. Some of them can be formulated in terms of LMIs.

	R	ER	E	EP	P
R	o		$\bullet$		o
ER		$\bullet$	$\bullet$		
E	∞	∞	$\bullet$	$\bullet$	$\bullet$
EP	∞	∞	∞	$\bullet$	
P	∞	∞	∞		o

Table 2.2: Some known induced norms

The ones marked with $\bullet$ are known and can be computed using the state space representation and LMIs directly. The three marked with *o* are known, but the induced norm can not be formulated in terms of LMIs. The mark ∞ indicates that the induced norm is in general infinity - except for the zero system. The blanks are currently unknown.

A well examined induced norm is the Energy to Energy induced norm, also known as the $\mathcal{H}_\infty$ norm. It has connections to robustness via the small gain theorem, see for instance [ZDG95].

For the $\mathcal{H}_\infty$ norm we have the following lemma (see [Sch90]; the proof is basically an extension of Lyapunov theory)

Lemma 2.2 (Bounded real lemma) *The following statements are equivalent*

i) $||T_{ed}||_{\mathcal{L}_{2,2}\to\mathcal{L}_{2,2}} < \gamma$ *and A is stable.*

ii) There exists a symmetric X such that

$$X > 0 \text{ and } \begin{bmatrix} A^T X + XA & XB_d & C_e^T \\ B_d^T X & -\gamma I & D_{ed}^T \\ C_e & D_{ed} & -\gamma I \end{bmatrix} < 0 \tag{2.11}$$

In (2.11) we have a convex region of Lyapunov matrices which satisfies the performance specification. This formulation is done for synthesis purposes. However, the result can be used to compute the $\mathcal{H}_\infty$ -norm by solving the following problem

$$\begin{array}{ll} \underset{X,\gamma}{\text{Minimize}} & \gamma \\ \text{subject to} & X > 0 \\ & \begin{bmatrix} A^T X + XA & XB_d & C_e^T \\ B_d^T X & -\gamma I & D_{ed}^T \\ C_e & D_{ed} & -\gamma I \end{bmatrix} < 0 \end{array} \tag{2.12}$$

which provides an optimal γ equal to the $\mathcal{H}_\infty$ -norm, and a Lyapunov matrix that proves stability and the optimality of the $\mathcal{H}_\infty$ -norm.

The above problem (2.12) is a *semidefinite programming problem* (SDP). A general formulation of a semidefinite programming problem is the following: Let $c \in \mathbb{R}^n$ and an LMI constraint $F(x) \geq 0$ be given. Then a semidefinite programming problem is an optimization problem of the form:

$$\begin{array}{ll} \underset{x}{\text{Minimize}} & c^T x \\ \text{subject to} & F(x) = F_0 + \sum_{i=1}^m x_i F_i \geq 0, F_i = F_i^T. \end{array} \tag{2.13}$$

A semidefinite programming problem is a convex problem since the objective is convex (linear), and the constraint is also convex.

Solution methods fora semidefinite programming problems is subject to intensive research at the moment, and various methods exist. Interior point methods were matured during the research by Nemirovskii and Nesterov [NN94], and resulted among others in the projective method by [NG94], and the primal-dual method by Boyd and Vandenberghe [VB96]. More recent so-called homogeneous algorithms have been proposed to solve the SDP problems.

Software for formulating and solving SDP's are numerous. The Projective Method by Nemirovskii and Gahinet have been implemented in the LMI control toolbox for use with MATLAB , [GNLC95]. The SP package by Boyd and Vandenberghe, [VB94], solves SDP's, but the problem has to be formulated in a special way, thus making it difficult to use for common users. This is alleviated by the user friendly front ends, SDPSOL [BW95] and LMITOOL [GDN95]. A large range of new methods are coming up, pointing out references to them all would be extensive, and out of date in a short time. Instead a search on the internet would be more fruitful and up to date. There are several pages dedicated to semidefinite programming, see `http://WWW-ISL.Stanford.EDU/ ~boyd/group_index.html`.

2.1.3 Other control objectives

Other performance objectives can be formulated in terms of LMI. First of all robustness with respect to unstructured uncertainties is equivalent to the $\mathcal{H}_\infty$ performance control problem by the small gain theorem, see for instance [ZDG95] or [Iwa93]. For robust $\mathcal{H}_2$ performance analysis the work by Paganini [Pag96] provides an LMI formulation, see also [Iwa93]. Also objectives in terms of systems that are linear parameter varying or linear time varying etc. have LMI formulations, but we will not consider these types of systems in the sequal.

2.2 Controllers and closed loop

In the previous section we considered specifications for the behavior of the closed loop system. The closed loop system consists of the connection between the original system and the controller. The connection between these two subsystems is established by use of actuators and measurement. The controller uses the measurement, denoted y, to construct a feedback signal u, that governs the actuators to provide the system with the desired behavior.

In this section we present a general model of the system, which includes the dynamical model of the system, the disturbance acting on it, the signals of specific interest (the error), and the available measurement and actuator signals. The set of possible controllers is then given. We then show a simple way of computing the closed loop matrices from the general model and the controller.

2.2.1 Open loop system

The general model we consider is the following.

$$\begin{bmatrix} \dot{x} \\ e \\ y \end{bmatrix} = \begin{bmatrix} \mathsf{A} & \mathsf{B}_d & \mathsf{B}_u \\ \mathsf{C}_e & \mathsf{D}_{ed} & \mathsf{D}_{eu} \\ \mathsf{C}_y & \mathsf{D}_{yd} & \mathsf{D}_{yu} \end{bmatrix} \begin{bmatrix} x \\ d \\ u \end{bmatrix} \tag{2.14}$$

The system has two inputs and two output, see figure 2.3. These input/outputs are paired, such that e, d are used to model the influence of disturbances d on a set of interesting signals e, refered to as the error, and y, u are the signals available for control. Since the system in (2.14) has not

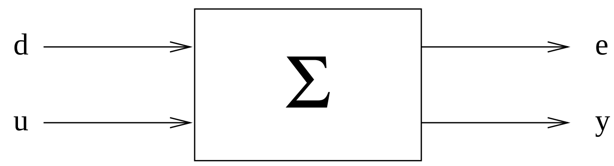


Figure 2.3: The setup for synthesis.

been connected with the controller yet, it is refered to as the *open loop system*. We denote the matrices of the open loop system using a sans serif font $\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}$.

We will in the following assume that $\mathsf{D}_{yu} = 0$. We can do this without loss of generality, see page 16.

2.2.2 Controller

To the above system we apply a controller between y and u of the form

$$\begin{bmatrix} \dot{x}_c \\ u \end{bmatrix} = \underbrace{\begin{bmatrix} \mathsf{A}_c & \mathsf{B}_c \\ \mathsf{C}_c & \mathsf{D}_c \end{bmatrix}}_{G} \begin{bmatrix} x_c \\ y \end{bmatrix}, \tag{2.15}$$

where $x_c \in \mathbb{R}^{n_c}$ are the states of the controller. **The number n_c of controller states is the order of the controller.** A static controller ($n_c = 0$) reduces to the simple form $u = \mathsf{D}_c y$. We introduce the notation G as indicated in (2.15). We will refer to G as the *controller parameter*.

2.2.3 Closed loop system

By applying the controller equation (2.15) to the system equation (2.14) we get the *closed loop system*. We will now derive formulation for the closed loop matrices as affine functions of the openloop matrices. This formulation will come out handy in the derivation of LMI formulation in the next chapter.

We will denote the closed loop matrices as in equation (2.8). Due to the assumption $\mathsf{D}_{yu} = 0$ it is possible to compute the closed loop matrices as an affine combination of the open loop matrices and the controller matrices. First augment the system equation (2.14) with the controller states. Next add an extra output namely the controller states x_c, and the derivative of the controller states $\dot{x}_c$ as an extra input. We call this augmented open loop system for the *augmented system*.

The open loop can now be written as

$$\begin{aligned} \begin{bmatrix} \dot{\tilde{x}} \\ \hline e \\ \hline \tilde{y} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{x}_c \\ \hline e \\ \hline x_c \\ y \end{bmatrix} &= \left[\begin{array}{cc|c|cc} \mathsf{A} & 0 & \mathsf{B}_d & 0 & \mathsf{B}_u \\ 0 & 0 & 0 & I & 0 \\ \hline \mathsf{C}_e & 0 & \mathsf{D}_{ed} & 0 & \mathsf{D}_{eu} \\ \hline 0 & I & 0 & 0 & 0 \\ \mathsf{C}_y & 0 & \mathsf{D}_{yd} & 0 & 0 \end{array}\right] \begin{bmatrix} x \\ x_c \\ \hline d \\ \hline \dot{x}_c \\ u \end{bmatrix} \\ &= \begin{bmatrix} \tilde{A} & \tilde{B}_d & \tilde{B}_u \\ \tilde{C}_e & \tilde{D}_{ed} & \tilde{D}_{eu} \\ \tilde{C}_y & \tilde{D}_{yd} & 0 \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \hline d \\ \hline \tilde{u} \end{bmatrix} \end{aligned} \tag{2.16}$$

where $\tilde{x} = [x^T\, x_c^T]^T$, $\tilde{u} = [\dot{x}_c^T\, u^T]^T$, and $\tilde{y} = [x_c^T\, y^T]^T$. The matrices in the augmented system are denoted with a tilde on the top of letters in a normal slanted font, $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$. Since we have included the controller states in the augmented system the controller law reduces to the simple algebraic relation $\tilde{u} = G\tilde{y}$.

Combining $\tilde{u} = G\tilde{y}$ and (2.16) and eliminating $\tilde{u}, \tilde{y}$ the following expression for the closed loop system can be derived

$$\begin{aligned} \begin{bmatrix} \dot{\tilde{x}} \\ \hline e \end{bmatrix} &= \begin{bmatrix} \tilde{A} & \tilde{B}_d \\ \tilde{C}_e & \tilde{D}_{ed} \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \hline d \end{bmatrix} + \begin{bmatrix} \tilde{B}_u \\ \tilde{D}_{eu} \end{bmatrix} \tilde{u} \\ &= \left(\begin{bmatrix} \tilde{A} & \tilde{B}_d \\ \tilde{C}_e & \tilde{D}_{ed} \end{bmatrix} + \begin{bmatrix} \tilde{B}_u \\ \tilde{D}_{eu} \end{bmatrix} G \begin{bmatrix} \tilde{C}_y & \tilde{D}_{yd} \end{bmatrix} \right) \begin{bmatrix} \tilde{x} \\ \hline d \end{bmatrix} \quad . \end{aligned} \tag{2.17}$$

The closed loop matrices can be extracted as an affine combination of the augmented system matrices and the controller parameter G:

$$\begin{bmatrix} A & B_d \\ C_e & D_{ed} \end{bmatrix} = \begin{bmatrix} \tilde{A} & \tilde{B}_d \\ \tilde{C}_e & \tilde{D}_{ed} \end{bmatrix} + \begin{bmatrix} \tilde{B}_u \\ \tilde{D}_{eu} \end{bmatrix} G \begin{bmatrix} \tilde{C}_y & \tilde{D}_{yd} \end{bmatrix} \tag{2.18}$$

The closed loop matrices are written in a normal font, A, B, C, D. The affine formulation of the closed loop matrices has two advantages. First it is easy to compute the matrices, and second, it is extremely useful in synthesis, as we will see in chapter 3.

We now return to treat the case where $\mathsf{D}_{yu} \neq 0$. Design the controller as if D_{yu} was zero. That is introduce a fictitious output $\hat{y} = y - \mathsf{D}_{yu}u$, and design the controller with this output. The controller is then given by the following relation

$$\begin{bmatrix} \dot{x}_c \\ u \end{bmatrix} = \begin{bmatrix} \mathsf{A}_c & \mathsf{B}_c \\ \mathsf{C}_c & \mathsf{D}_c \end{bmatrix} \begin{bmatrix} x_c \\ \hat{y} \end{bmatrix}.$$

We can eliminate $\hat{y}$ and get the following controller

$$\begin{bmatrix} \dot{x}_c \\ u \end{bmatrix} = \begin{bmatrix} \mathsf{A}_c - \mathsf{B}_c\mathsf{D}_{yu}(I + \mathsf{D}_c\mathsf{D}_{yu})^{-1}\mathsf{C}_c & \mathsf{B}_c(I - \mathsf{D}_{yu}(I + \mathsf{D}_c\mathsf{D}_{yu})^{-1}\mathsf{D}_c) \\ (I + \mathsf{D}_c\mathsf{D}_{yu})^{-1}\mathsf{C}_c & (I + \mathsf{D}_c\mathsf{D}_{yu})^{-1}\mathsf{D}_c \end{bmatrix} \begin{bmatrix} x_c \\ y \end{bmatrix} \tag{2.19}$$

assuming that $(I + D_cD_{yu})^{-1}$ exists.

2.2.4 Controller constraints

When we restricted the controller in equation (2.15) to be in state-space form, finite-dimensional and time-invariant, we did so mostly of convenience since we are designing a controller for systems that have the same form. Normally a controller is implemented using Programmable Logic Controllers (PLC's). The PLC unit is used to implement the transfer function of the controller as a filter on state space form. The output of the PLC unit is then fed into an amplifier that can supply the desired energy. However, usually such a PLC unit have constraints on the possible order of the controller, and moreover the unit has finite precision arithmetic. In other words, we have to constraint the controller in our design.

The constraint on the control order is easily stated by the requirement that $n_c \leq \overline{n}_c$, but it cannot be formulated in terms of LMIs. In fact the constraint on the orderimposes a non-convex constraint on the LMI formulations. We will treat this in great detail later.

The limited precision of the arithmetic makes the input/output mapping of the controller very dependent on the gains in the controller. We will impose constraints on the gain of the controller by restricting the controller parameter G. This can be done in two ways. First of all each entry G_{kl} can be bounded by $l_{kl} \leq G_{kl} \leq u_{kl}$, where l_{kl}, u_{kl} are finite. and $l_{kl} < u_{kl}$. As an alternatively we can add an upper bound β on the Frobenius norm of the controller parameter:

$$||G||_{\text{Frob}} = \sqrt{\operatorname{Tr} G^T G} < \beta. \tag{2.20}$$

Given G it is easy to calculate the Frobenius norm of it using (2.20). If we want to impose the constraint in a search for controllers fulfilling other specifications, a convex formulation in terms of LMIs are more adequate. We have the following, see [BGFB94, chap. 2]:

Proposition 2.1 (Frobenius norm as LMI constraint) *Given $G \in \mathbb{R}^{p \times q}$, and $0 < \beta < \infty$, then the following is equivalent*

i) The Frobenius norm of G is less than β, that is $\sqrt{\operatorname{Tr} G^T G} < \beta$.

ii) There exist symmetric $S \in \mathcal{S}^p$ such that $\operatorname{Tr} S < \beta^2$, $\begin{bmatrix} S & G \\ G^T & I \end{bmatrix} \geq 0$.

The convex constraint on G is constructed by introducing a so called slack variable, S, extending the number of variables with the number of independent variables in $\mathcal{S}^p$. We will also use the result in the above proposition in chapter 4.

2.3 Control problems

In this section we will combine the specifications and the closed loop formulation given above to formulate a couple of interesting problems. Each problem is basically on the form: Given a set of specifications and an open loop system (2.14) find a controller (2.15), if it exists, such that the closed loop (2.8) fulfills the given specifications. As specifications we consider objectives in relation to stability, α-stability, and performance, as defined earlier. Again we stress that these objectives are only instances of the number of specifications that can be solved in the same way. We will also include the order of the controller in the problem formulation. When we say *fixed-order ...* control problem we are looking for a controller of a order not exceeding a specified number.

2.3.1 Stability

The most simple control problem is that of finding a controller such that the closed loop system is stable.

Stabilizing control problem: Given the open loop system equation (2.14) find a controller equation (2.15) (if it exists) such that the closed loop system $\dot{\tilde{x}} = A\tilde{x}$ is stable.

The stabilizing control problem seems very simple, but in fact there are many unsolved questions in relations to the problem. We will later consider an extra constraint in terms of the control order on the above control problem.

With a constraint on the control order the stabilizing control problem get the following form

Fixed-order Stabilizing control problem: Given the open loop system equation (2.14), and a desired control order $\overline{n}_c$ find a controller equation (2.15) of order at most $\overline{n}_c$ (if it exists) such that the closed loop system $\dot{\tilde{x}} = A\tilde{x}$ is stable.

This control problem contains essentially the same computational difficulties as the problem we pose in the sequel. In fact when studying the efficiency of a computational method dealing with fixed order control design it is sufficient to study the *fixed-order stabilizing control problem*.

Here we return to formalize another simple control problem. In the stabilizing control problem we just ask for a convergence of the states to zero. We have no constraints on the speed of convergence, but by considering α-stability we get exponential convergence of the states.

α-stabilizing control problem: Given $\alpha \in \mathbb{R}, \alpha > 0$ and the open loop system equation (2.14) find a controller equation (2.15) (if it exists) such that the closed loop system has α-stability.

This problem can be solved for any $0 < \alpha < \infty$ under simple assumptions. In other words the maximal α is infinity. However, if constraints are added on the controller order the optimal α can be shown to be finite even for simple problems.

Fixed order Optimal α-stabilizing control problem: Given the open loop system (2.14) and a desired control order $\overline{n}_c$ find a controller equation (2.15) of order at most $\overline{n}_c$ (if it exists) such that the closed loop system has maximal possible α-stability.

Similarly we could have defined control problems which deal with general pole-placement, see [CG96].

2.3.2 Performance

The issues of stability and α-stability are very important, but another interesting question is how well disturbances are damped. We refered to this as the performance of the system, and showed that it had relations to the induced norm. A low induced norm is a sign of good performance. The induced norm of a system is a positive real number, and one specification could be to constraint the induced norm of the closed loop system to be less than a specified positive number γ.

For the induced norm we consider to find the controller with a certain guaranteed performance:

Performance control problem: Given an induced norm $||\bullet||_{\bullet}$, a $\gamma > 0$, and the open loop system equation (2.14) find a controller equation (2.15) if it exists such that the closed loop system is stable and has induced norm $||T_{ed}||_{\bullet}$ less than γ.

The possible induced norms in this context are the one indicated with $\bullet$'s in table 2.2. Since the induced norm from energy to energy is of great importance and widely examined in the literature, we will specifically state the related control problem:

$\mathcal{H}_\infty$ **control problem:** Given a $\gamma > 0$, and the open loop system equation (2.14) find a controller equation (2.15) if it exists such that the closed loop system is stable and has induced norm $||T_{ed}||_{\mathcal{L}_{2,2} \to \mathcal{L}_{2,2}}$ less than γ.

2.3.3 Other control problems

The above given control problems are the ones we will consider explicitly in this presentation, and we will derive formulations for the existence of controllers for these problems. A lot of other control problems could be defined using different objectives, but the above given problems, are sufficient for our purpose. That is, studying optimization algorithms for solving control problems.

Chapter 3

Control problems as optimization problems

The problems stated in the previous chapter have been studied in the field of control theory during the last 40 years. We will not try to give a complete overview of the published results, but present the most important.

For the stabilizing control problem, results derived in the 1970's provide a complete parameterization of all stabilizing controllers. This characterization of all stabilizing controllers is usually refered to as *Youla parameterization*, see for instance [ZDG95, chap 12]. Although the result is nice for the stabilizing control problem it is not easily extended to other control problems.

The $\mathcal{H}_\infty$ problem was solved in beginning of the 1980's, see [Sto92] for an historical overview. The approach based on the solution to two Riccati equation was first presented in [DGKF89], and gave an elegant result solving the $\mathcal{H}_\infty$ control problem by use of the Riccati equation. At the same time the so called $\mathcal{H}_2$ control problem was solved using the same technique. The drawback of this technique is that it always provide controllers of order equal to the plant. The order of the controller could be reduced using model reduction techniques, but usually at the cost of a worse performance for the closed loop. Recall that we are interested in minimizing the control order, since it is usually limited by hardware.

By formulating the problems in terms of matrix inequalities, we can derive results where it is possible to bound the control order. In a paper by Packard, Zhou, Pandey, and Becker, [PZPB91], a collection of robust problems leading to LMIs was given, initiating the search for LMI formulations of a number of control problems. The first milestone was reached in 1992 in [IS93, GA93], when the $\mathcal{H}_\infty$ -control problem was solved simultaneously by two research groups. Skelton and Iwasaki [IS94] provided a complete parameterization of all controllers achieving a specific bound γ on the $\mathcal{H}_\infty$ norm. Gahinet and Apkarian [GA94] presented a similar LMI solution to the $\mathcal{H}_\infty$ control problem, but omitted the parameterization. However, they presented some additional properties, that we will point out later in this chapter. In his Ph.D.'s thesis [Iwa93], Iwasaki provided the solution to a series of other control problems within continuous and discrete systems,

Since the derivation of the LMI result for the $\mathcal{H}_\infty$ control problem a series of other control problem have been solved in a similar fashion. It should be emphasized that a large class of many other control design problems such as Linear Quadratic control, covariance control, positive-real control, μ-synthesis with constant scaling, and linear parameter-varying control can be formulated in a similar mathematical framework via linear matrix inequalities and a coupling matrix rank constraint; see [SIG97].

In this chapter we will present a brief introduction to the results given in [IS94, GA94]. The approach is the following. We derive results in terms of bilinear matrix inequalities (BMIs) by combining the analysis results on the closed loop with the affine parameterization of all possible closed loop matrices in terms of the open loop matrices and the controller parameter. However, we will see that the resulting BMI is very difficult to solve in general, although an attempt will be made in chapter 5. However, the search for a solution to the BMI can be reformulated as a search for a solution to a set of coupled LMIs, by relaxing the controller order. It turns out that the control order is the rank of a given matrix.

We will show the derivation of both the BMI result and the LMI result, by studying the stabilizing and $\mathcal{H}_\infty$ control problems. It turns out that the major steps are the same, and the difference can be traced back to the difference between Lyapunov stability result and the bounded real lemma.

3.1 Bilinear Matrix Inequality formulation

In this section we formulate the control problems defined in last chapter in terms of bilinear matrix inequalities.

The fixed order stabilizing control problem is to find a controller G of order at most n_c, such that the closed loop $\dot{\tilde{x}} = A\tilde{x}$, is stable. We will use the Lyapunov result, $\exists Y > 0,\ A^T Y + YA < 0$, to study the stability of the closed loop. Recall from (2.18) the relation

$$A = \tilde{A} + \tilde{B}_u G \tilde{C}_y$$

between the closed loop matrix A, the controller parameter G and the augmented matrices $\tilde{A}, \tilde{B}_u$, and $\tilde{C}_y$. Inserting $A = \tilde{A} + \tilde{B}_u G \tilde{C}_y$ in $A^T Y + YA$ gives the following formulation of the fixed order stabilizing control problem:

$$\text{Find } G, Y \text{ such that } Y > 0,\ \left(\tilde{A} + \tilde{B}_u G \tilde{C}_y\right)^T Y + Y\left(\tilde{A} + \tilde{B}_u G \tilde{C}_y\right) < 0. \tag{3.1}$$

where the augmented matrices are extented to include the controller states of size n_c. The last expression in (3.1) is a *bilinear matrix inequality*, i.e. an inequality of the form:

$$F(x,y) \triangleq F_0 + \sum_{i=1}^{m_x} x_i F_i^x + \sum_{j=1}^{m_y} y_j F_j^y + \sum_{j=1}^{m_y} \sum_{i=1}^{m_x} y_j x_i F_{ij}^{xy} \geq 0. \tag{3.2}$$

where the variables are $x \in \mathbb{R}^{m_x}$ and $y \in \mathbb{R}^{m_y}$, and the matrices $F_i^x \in \mathcal{S}^n, i = 1, \ldots, m_x$, $F_j^y \in \mathcal{S}^n, j = 1, \ldots, m_y$, and $F_{ij}^{xy} \in \mathcal{S}^n, i = 1, \ldots, m_x, j = 1, \ldots, m_y$, are given data. The inequality sign '$\geq$' denotes, as usual, positive semidefiniteness. We will refer to the problem of finding feasible (x,y), i.e. (x,y) such that $F(x,y) \geq 0$, as the *BMI feasibility problem*. One of the goals in this thesis is to derive algorithms that can solve the BMI feasibility problem. In chapter 5 we will formalize what we exactly mean by “solving”.

The term BMI was introduced by Safonov, Goh and Ly in 1994 in the conference paper [SGL94]. The BMI is a general framework and can be used to formulate a series of system synthesis problems. In this chapter we will concentrate on control problems.

The feasibility set of a BMI $F(x,y) \geq 0$, i.e. $\{(x,y)\ :\ F(x,y) \geq 0\}$ is in general non-convex. Take for example $F(x,y) = 1 - xy \geq 0$. Then the feasibility set $\{(x,y) : x,y \in \mathbb{R},\ 1 - xy \geq 0\}$ is non-convex.

However, a BMI do have one nice property: For fixed y the BMI (3.2) reduces to a linear matrix inequality (LMI) in the variable x; for fixed x it reduces to an LMI in the variable y. This property is called *biconvexity*. The biconvexity can be used to solve the BMI feasibility problem as described above, see chapter 5.

For the fixed order α stabilizing control problem we can in a similar way derive the following lemma:

Lemma 3.1 (BMI) *Let a scalar $\alpha > 0$, a desired control order $\overline{n}_c$ and augmented matrices A, B_u and C_y with control order $\overline{n}_c$ be given. The matrices $\tilde{A}$, $\tilde{B}_u$ and $\tilde{C}_y$ are augmented with the control order as in (2.16). The following statements are equivalent*

(i) There exists a matrix G such that the closed loop $A = \tilde{A} + \tilde{B}_u G \tilde{C}_y$ is α stable.

(ii) There exists $Y \in \mathcal{S}^{n+n_c}$ and $G \in \mathbb{R}^{(n_c+n_y)\times(n_c+n_u)}$ such that

$$Y > 0, \ \left(\tilde{A} + \tilde{B}_u G \tilde{C}_y + \alpha I\right)^T Y + Y\left(\tilde{A} + \tilde{B}_u G \tilde{C}_y + \alpha I\right) < 0 \ . \tag{3.3}$$

Suppose we want to solve the fixed order optimal α-stabilizing control problem. We then get the following problem:

$$\begin{array}{ll} \underset{Y,G,\alpha}{\text{Maximize}} & \alpha \\ \text{subject to} & Y > 0, \ G \in \mathbb{R}^{(n_c+n_y)\times(n_c+n_u)} \\ & \left(\tilde{A} + \tilde{B}_u G \tilde{C}_y + \alpha I\right)^T Y + Y\left(\tilde{A} + \tilde{B}_u G \tilde{C}_y + \alpha I\right) < 0. \end{array} \tag{3.4}$$

The above is what we call a *Bilinear matrix inequality optimization problem*:

$$\begin{array}{ll} \underset{x,y}{\text{Minimize}} & c^T x + d^T y \\ \text{subject to} & F(x,y) = F_0 + \sum_{i=1}^{m_x} x_i F_i^x + \sum_{j=1}^{m_y} y_j F_j^y + \sum_{j=1}^{m_y}\sum_{i=1}^{m_x} y_j x_i F_{ij}^{xy} \geq 0. \end{array} \tag{3.5}$$

where x and y are the variables, and $c \in \mathbb{R}^{m_x}$,$d \in \mathbb{R}^{m_y}$, and $F_\bullet^\bullet \in \mathcal{S}^n$ are given data. Since the BMI is non-convex, the BMI optimization problem is a non-convex optimization problem. We will in chapter 5 partition the BMI into convex and non-convex constraints, and try to exploit this partition together with the biconvexity to find an efficient solver.

For the fixed order $\mathcal{H}_\infty$ control problem we get a formulation in terms of BMIs by inserting the parameterization of the closed loop matrices equation (2.18) in the bounded real lemma. We obtain the following BMI feasibility problem:

Find G and $X > 0$ such that

$$\begin{bmatrix} \left(\tilde{A} + \tilde{B}_u G \tilde{C}_y\right)^T X + X\left(\tilde{A} + \tilde{B}_u G \tilde{C}_y\right) & X\left(\tilde{B}_d + \tilde{B}_u G \tilde{D}_{yd}\right) & \left(\tilde{C}_e + \tilde{D}_{eu} G \tilde{C}_y\right)^T \\ \left(\tilde{B}_d + \tilde{B}_u G \tilde{D}_{yd}\right)^T X & -\gamma I & \left(\tilde{D}_{ed} + \tilde{D}_{eu} G \tilde{D}_{yd}\right)^T \\ \left(\tilde{C}_e + \tilde{D}_{eu} G \tilde{C}_y\right) & \left(\tilde{D}_{ed} + \tilde{D}_{eu} G \tilde{D}_{yd}\right) & -\gamma I \end{bmatrix} < 0 \tag{3.6}$$

The optimal $\mathcal{H}_\infty$ controller can be found by finding the optimum of the BMI optimization problem:

$$\begin{array}{ll} \underset{\gamma,X,G}{\text{Minimize}} & \gamma \\ \text{subject to} & X > 0, \ G \in \mathbb{R}^{(n_c+n_y)\times(n_c+n_u)} \\ & \text{the BMI (3.6)} \end{array} \tag{3.7}$$

In the BMI optimization problems (3.7) and (3.4) the set of variables are in general unbounded. This causes problems for any attempt to solve the problem. For the controller parameter G we can remove this by adding a bound on the Frobenius norm of G in the form of an LMI, cf. proposition 2.1. A bound on the Lyapunov matrix can be made by adding $\operatorname{Tr} X \leq \beta$. This does introduce some conservatism, but as we will discuss in both chapter 4 and chapter 5 it has some advantages in terms of numerical computations.

3.2 Convex formulation

In this section we will show how to derive convex results for the solvability of a specific control problem. The framework is the simple matrix inequality

$$FGH + (FGH)^T + Q < 0. \tag{3.8}$$

Any of the BMIs formulated in the last section can be written in the form (3.8). The convex results are then obtained by applying the so called elimination lemma to matrix inequalities on the form (3.8), and showing that a relaxation of the control order is sufficient to derive convex results.

First we show that all the BMI problems can be reformulated in the form (3.8). The BMI in (3.3) for the α stabilizing control problem

$$\left(\tilde{A} + \tilde{B}_u G \tilde{C} + \alpha I\right)^T Y + Y \left(\tilde{A} + \tilde{B}_u G \tilde{C} + \alpha I\right) < 0$$

can be rewritten as

$$\underbrace{Y\tilde{B}_u}_{F_{\alpha,n_c}} G \underbrace{\tilde{C}_y}_{H_{\alpha,n_c}} + \left(Y\tilde{B}_u G \tilde{C}_y\right)^T + \underbrace{\tilde{A}^T Y + Y\tilde{A} + 2\alpha Y}_{Q_{\alpha,n_c}} < 0. \tag{3.9}$$

In the above we have F_{α,n_c}, H_{α,n_c} and Q_{α,n_c}, defined as

$$\begin{aligned} F_{\alpha,n_c} &= Y\tilde{B}_u \\ H_{\alpha,n_c} &= \tilde{C}_y \\ Q_{\alpha,n_c} &= \tilde{A}^T Y + Y\tilde{A} + 2\alpha Y \end{aligned}$$

where α denotes the relation to the α stabilizing control problem, and n_c refers to the augmentation of the matrices $\tilde{B}_u$, $\tilde{C}_y$ and $\tilde{A}$. Since we consider the open loop matrices to be fixed, we observe that only F_α and Q_α are functions of Y, where as H_α is constant. By congrurence transformation with $Y^{-1} = X$ we get a dual formulation (denoted with $'$) with

$$\begin{aligned} F'_{\alpha,n_c} &= \tilde{B}_u \\ H'_{\alpha,n_c} &= \tilde{C}_y X \\ Q'_{\alpha,n_c} &= \tilde{A}X + X\tilde{A}^T + 2\alpha X \end{aligned}$$

Similarly we can rewrite the BMI (3.6) as,

$$\underbrace{\begin{bmatrix} \tilde{A}^T X + X\tilde{A} & X\tilde{B}_d & \tilde{C}_e^T \\ \tilde{B}_d^T X & -\gamma I & \tilde{D}_{ed}^T \\ C_e & \tilde{D}_{ed} & -\gamma I \end{bmatrix}}_{Q_{\mathcal{H}_\infty,n_c}} + \underbrace{\begin{bmatrix} X\tilde{B}_u \\ 0_{n_d\times n_u} \\ \tilde{D}_{eu} \end{bmatrix}}_{F_{\mathcal{H}_\infty,n_c}} G \underbrace{\begin{bmatrix} \tilde{C}_y & \tilde{D}_{yd} & 0_{n_y\times n_e} \end{bmatrix}}_{H_{\mathcal{H}_\infty,n_c}}$$
$$+ \left(\begin{bmatrix} X\tilde{B}_u \\ 0_{n_d\times n_u} \\ \tilde{D}_{eu} \end{bmatrix} G \begin{bmatrix} \tilde{C}_y & \tilde{D}_{yd} & 0_{n_y\times n_e} \end{bmatrix} \right)^T < 0 \quad (3.10)$$

and we get

$$F_{\mathcal{H}_\infty,n_c} G H_{\mathcal{H}_\infty,n_c} + \left(F_{\mathcal{H}_\infty,n_c} G H_{\mathcal{H}_\infty,n_c}\right)^T + Q_{\mathcal{H}_\infty,n_c} < 0.$$

where

$$\begin{aligned} F_{\mathcal{H}_\infty,n_c} &= \begin{bmatrix} X\tilde{B}_u \\ 0_{n_d\times n_u} \\ \tilde{D}_{eu} \end{bmatrix} \\ H_{\mathcal{H}_\infty,n_c} &= \begin{bmatrix} \tilde{C}_y & \tilde{D}_{yd} & 0_{n_y\times n_e} \end{bmatrix} \\ Q_{\mathcal{H}_\infty,n_c} &= \begin{bmatrix} \tilde{A}^T X + X\tilde{A} & X\tilde{B}_d & \tilde{C}_e^T \\ \tilde{B}_d^T X & -\gamma I & \tilde{D}_{ed}^T \\ C_e & \tilde{D}_{ed} & -\gamma I \end{bmatrix} . \end{aligned} \quad (3.11)$$

In general F, H, and Q are functions of the Lyapunov matrix, so there is a bilinear connection between the Lyapunov matrix and the controller parameter G. The first step towards removing the bilinear connection is to eliminate the controller parameter G from the BMI constraint. To this end we use the following lemma.

Lemma 3.2 (Elimination lemma) *Let matrices $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{k\times n}$ and $Q = Q^T \in \mathbb{R}^{n\times n}$ be given. Consider a set of matrices*

$$\mathcal{G}(B,C,Q) \triangleq \left\{ G \in \mathbb{R}^{m\times k} : BGC + (BGC)^T + Q < 0 \right\} \quad (3.12)$$

The following statements are equivalent

(i) $\mathcal{G}(B,C,Q) \neq \emptyset$.

(ii) The following statements hold

$$\begin{aligned} B^{\perp} Q B^{\perp T} < 0 \quad &or \quad BB^T > 0 \\ C^{T\perp} Q C^{T\perp T} < 0 \quad &or \quad C^T C > 0 \ . \end{aligned}$$

Proof:

A rigorous proof of the above lemma can be found in [Iwa93]. Here we will justify the lemma by some simple observations. To do this we exploit, that if $Q > 0$ and T has full row rank, then $TQT^T > 0$. Multiplying $BGC + (BGC)^T + Q < 0$ from the left with the left annihilator of B, denoted $B^{\perp}$, ($B^{\perp}$ has full column rank and $B^{\perp}B = 0$), and from the right with its transpose, we get

$$0 > B^{\perp}BGCB^{\perp T} + B^{\perp}C^TG^TB^TB^{\perp T} + B^{\perp}QB^{\perp T} = B^{\perp}QB^{\perp T}.$$

In a similar way we can get the condition $C^{T\perp}QC^{T\perp T} < 0$, by multiplying from the left with $C^{T\perp}$ and from the right with its transpose. Noting that the left annihilator of B only exists if B does not have full column rank, we get the constraint $BB^T > 0$. •

We now use the elimination lemma to eliminate the controller parameter G from the BMI. The BMI (3.9) has a solution if and only if

$$\left(Y\tilde{B}_u\right)^{\perp}\left(\tilde{A}^TY + Y\tilde{A} + 2\alpha Y\right)\left(Y\tilde{B}_u\right)^{\perp T} < 0 \ \text{ or } \ \left(Y\tilde{B}_u\right)\left(Y\tilde{B}_u\right)^T > 0 \tag{3.13}$$

and

$$\tilde{C}_y^{T\perp}\left(\tilde{A}^TY + Y\tilde{A} + 2\alpha Y\right)\tilde{C}_y^{T\perp T} < 0 \ \text{ or } \ \tilde{C}_y^T\tilde{C}_y > 0. \tag{3.14}$$

Since $Y > 0$ the condition $Y\tilde{B}_u\tilde{B}_u^TY > 0$ is equivalent to

$$\tilde{B}_u\tilde{B}_u^T = \begin{bmatrix} 0 & \mathsf{B}_u \\ I & 0 \end{bmatrix}\begin{bmatrix} 0 & \mathsf{B}_u \\ I & 0 \end{bmatrix}^T > 0$$

which again is equivalent to B_u having full row rank. That is the rank of B_u is equal to the number of states, considering the relation $\dot{x} = \mathsf{A}x + \mathsf{B}_u u$, we see that we have full control over the derivative of the states, if we know the states x. Similarly the condition $\tilde{C}_y^T\tilde{C}_y > 0$ implies that C_y has full column rank, and we can solve the equation $\mathsf{C}x = y$ for x. Having explained this implication of $Y\tilde{B}_u\tilde{B}_u^TY > 0$ and $\tilde{C}_y^T\tilde{C}_y > 0$, we restrict our attention to

$$\begin{aligned} \left(Y\tilde{B}_u\right)^{\perp}\left(\tilde{A}^TY + Y\tilde{A} + 2\alpha Y\right)\left(Y\tilde{B}_u\right)^{\perp T} &< 0 \\ \tilde{C}_y^{T\perp}\left(\tilde{A}^TY + Y\tilde{A} + 2\alpha Y\right)\tilde{C}_y^{T\perp T} &< 0 \ . \end{aligned} \tag{3.15}$$

By noting that $\left(Y\tilde{B}_u\right)^{\perp}$ can be chosen as $\tilde{B}_u^{\perp}Y^{-1}$, we first get the following formulation of the left condition in (3.13):

$$\tilde{B}_u^{\perp}\left(Y^{-1}\tilde{A}^T + \tilde{A}Y^{-1} + 2\alpha Y^{-1}\right)\tilde{B}_u^{\perp T} < 0 \tag{3.16}$$

We introduce the following sets:

$$\begin{aligned} \mathcal{X}_{\alpha,n_c} &\triangleq \left\{X : X \in \mathcal{S}^{n+n_c},\ \tilde{B}_u^{\perp}\left(X\tilde{A}^T + \tilde{A}X + 2\alpha X\right)\tilde{B}_u^{\perp T} < 0\right\}, \\ \mathcal{Y}_{\alpha,n_c} &\triangleq \left\{Y : Y \in \mathcal{S}^{n+n_c},\ \tilde{C}_y^{T\perp}\left(\tilde{A}^TY + Y\tilde{A} + 2\alpha Y\right)\tilde{C}_y^{T\perp T} < 0\right\}. \end{aligned}$$

With this notation we have that there exist a controller of order n_c, that solves the α stabilizing control problem if and only if there exists Y such that

$$Y > 0, Y^{-1} \in \mathcal{X}_{\alpha,n_c} \text{ and } Y \in \mathcal{Y}_{\alpha,n_c} \tag{3.17}$$

This is a non convex formulation and we would like to find a convex one. We now need to introduce the following partition of Y and its inverse:

$$Y = \begin{bmatrix} \mathsf{Y} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} \text{ and } Y^{-1} = \begin{bmatrix} \mathsf{X} & * \\ * & * \end{bmatrix} \tag{3.18}$$

with $\mathsf{X}, \mathsf{Y} \in \mathbb{R}^{n\times n}$. This is done to exploit the structure of the augmented matrices, equation (2.16). The structure of the augmented matrices $\tilde{B}_u$ and $\tilde{C}_y$ impose a nice structure on these left annihilators:

$$\tilde{B}_u^{\perp} = \begin{bmatrix} \mathsf{B}_u^{\perp} & 0 \end{bmatrix} \text{ and } \tilde{C}_y^{T\perp} = \begin{bmatrix} \mathsf{C}_y^{T\perp} & 0 \end{bmatrix}$$

A closer look at the left condition in (3.14) yields the following:

$$\begin{aligned} 0 &> \tilde{C}_y^{T\perp} \left(\tilde{A}^T Y + Y\tilde{A} + 2\alpha Y \right) \tilde{C}_y^{T\perp T} \\ &= \begin{bmatrix} \mathsf{C}_y^{T\perp} & 0 \end{bmatrix} \left(\begin{bmatrix} \mathsf{A}^T & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathsf{Y} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} + \begin{bmatrix} \mathsf{Y} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} \begin{bmatrix} \mathsf{A} & 0 \\ 0 & 0 \end{bmatrix} + \right. \\ &\qquad \left. 2\alpha \begin{bmatrix} \mathsf{Y} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} \right) \begin{bmatrix} \mathsf{C}_y^{T\perp T} \\ 0 \end{bmatrix} \\ &= \mathsf{C}_y^{T\perp} \left(\mathsf{A}^T \mathsf{Y} + \mathsf{Y}\mathsf{A} + 2\alpha \mathsf{Y} \right) \mathsf{C}_y^{T\perp T}. \end{aligned} \tag{3.19}$$

Similarly we can simplify (3.16), by inserting the choice of $\tilde{B}_u^{\perp} = \begin{bmatrix} \mathsf{B}_u^{\perp} & 0 \end{bmatrix}$ given above:

$$\mathsf{B}_u^{\perp} \left(\mathsf{A}\mathsf{X} + \mathsf{X}\mathsf{A} + 2\alpha \mathsf{X} \right) \mathsf{B}_u^{\perp T} < 0 \ . \tag{3.20}$$

The condition (3.17) can now be written as

$$\begin{gathered} \mathsf{B}_u^{\perp} \left(\mathsf{A}\mathsf{X} + \mathsf{X}\mathsf{A} + 2\alpha \mathsf{X} \right) \mathsf{B}_u^{\perp T} < 0 \\ \mathsf{C}_y^{T\perp} \left(\mathsf{A}^T \mathsf{Y} + \mathsf{Y}\mathsf{A} + 2\alpha \mathsf{Y} \right) \mathsf{C}_y^{T\perp T} < 0 \\ Y > 0 \\ Y = \begin{bmatrix} \mathsf{Y} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} \\ Y^{-1} = \begin{bmatrix} \mathsf{X} & * \\ * & * \end{bmatrix} \end{gathered} \tag{3.21}$$

The first two conditions above are independent of the control order, where as the size of Y and therefore the last three conditions are dependent. By relaxing the control order the last three conditions in (3.21) can be convexified. We summarize this in the following theorem

Theorem 3.1 (Coupling condition) *Given* X, Y *satisfying*

$$\begin{bmatrix} \mathsf{X} & I \\ I & \mathsf{Y} \end{bmatrix} \geq 0 \ , \tag{3.22}$$

there exists $Y \in \mathcal{S}^{n+n_c}$ *where* $n_c = \operatorname{Rank} X - Y^{-1}$ *such that*

$$Y = \begin{bmatrix} \mathsf{Y} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} > 0 \tag{3.23}$$

$$Y^{-1} = \begin{bmatrix} \mathsf{X} & * \\ * & * \end{bmatrix} \tag{3.24}$$

On the contrary, let Y *fulfill (3.23) and (3.24), then* X, Y *fulfill (3.22).*

Proof:

The proof for the above was originally given in [PZPB91], see also [Iwa93]. Since the theorem plays an important rule in the LMI formulation, we give a complete proof here.

First assume that (3.23) and (3.24) holds. We consider two cases, $n_c = 0$ and $n_c > 0$. For $n_c > 0$ the positive definiteness of Y implies that $Y_{22} > 0$ and $\mathsf{Y} > 0$ and $\mathsf{X} > 0$ and via the inversion formulae for matrices, see [ZDG95, chap. 2] we have $\mathsf{X} = \left(\mathsf{Y} - Y_{12}Y_{22}^{-1}Y_{12}^T\right)^{-1}$ which is equivalent to

$$\mathsf{Y} - \mathsf{X}^{-1} = Y_{12}Y_{22}^{-1}Y_{12}^T \quad . \tag{3.25}$$

The right hand side in equation (3.25) is positive semidefinite, because $Y_{22} > 0$, and we get $\mathsf{Y} - \mathsf{X}^{-1} \geq 0$. Using Schur's complement and the fact that $\mathsf{X} > 0$ implies equation (3.22).

For $n_c = 0$ we have $0 < Y = \mathsf{Y}$ and $0 < Y^{-1} = \mathsf{X}$ implying that $\mathsf{Y} = \mathsf{X}^{-1}$, which relaxed becomes $\mathsf{Y} \geq \mathsf{X}^{-1}$. Schur's complement with $\mathsf{X} > 0$ and $Y - X^{-1} \geq 0$ implies equation (3.22).

Suppose now that (3.22) holds then we need to show that we can find Y such that (3.23) and (3.24) hold. The coupling constraint is equivalent to $\mathsf{X} > 0, \mathsf{Y} \geq \mathsf{X}^{-1}$ or $\mathsf{Y} > 0, \mathsf{X} \geq \mathsf{Y}^{-1}$. The left hand side of (3.25) is positive semidefinite because $\mathsf{Y} \geq \mathsf{X}^{-1}$, and we will define the rank of it as $n_c \triangleq \mathrm{Rank}(\mathsf{Y} - \mathsf{X}^{-1})$. The upper left part of Y^{-1} is equal to $\left(\mathsf{Y} - Y_{12}Y_{22}^{-1}Y_{12}^T\right)^{-1}$ which we require to be equal to X, that is

$$\left(\mathsf{Y} - Y_{12}Y_{22}^{-1}Y_{12}^T\right)^{-1} = \mathsf{X} \quad ,$$

which is equivalent to

$$Y_{12}Y_{22}^{-1}Y_{12}^T = \mathsf{Y} - \mathsf{X}^{-1} \quad . \tag{3.26}$$

Choose $Y_{22} = I_{n_c} > 0$ then since the right hand side of (3.26) is positive semidefinite we can find Y_{12} as the matrix square root of $\mathsf{Y} - \mathsf{X}^{-1} = Y_{12}Y_{12}^T$. From this it follows that

$$\mathsf{Y} - Y_{12}Y_{12}^T = \mathsf{X}^{-1} > 0$$

and together with $\mathsf{Y} > 0$ that $Y \in \mathcal{S}^{n+n_c}$ and

$$Y = \begin{bmatrix} Y & Y_{12} \\ Y_{12}^T & I_{n_c} \end{bmatrix} > 0 \quad .$$

By construction Y^{-1} has X in the upper left corner. •

The condition (3.22) is refered to as the *coupling condition*. As it can be seen from theorem 3.1 the control order can be found as

$$n_c = \mathrm{Rank}\left(\mathsf{X} - \mathsf{Y}^{-1}\right) \quad . \tag{3.27}$$

Theorem 3.1 states that if X and Y fulfills the coupling constraint then we can always construct a Lyapunov matrix of appropriate size satisfying (3.21).

We now have the following result for the existence of a α stabilizing controller: Find X and Y such that

$$\mathsf{B}_u^{\perp}\left(\mathsf{AX} + \mathsf{XA} + 2\alpha\mathsf{X}\right)\mathsf{B}_u^{\perp T} < 0 \tag{3.28}$$

$$\mathsf{C}_y^{T\perp}\left(\mathsf{A}^T\mathsf{Y} + \mathsf{YA} + 2\alpha\mathsf{Y}\right)\mathsf{C}_y^{T\perp T} < 0 \tag{3.29}$$

$$\begin{bmatrix} \mathsf{X} & I \\ I & \mathsf{Y} \end{bmatrix} \geq 0 \quad . \tag{3.30}$$

The above three LMI constraints each define a convex set. We will define the feasibility set of the first two LMIs (3.28) and (3.29) as:

$$\Gamma_\alpha \triangleq \left\{ (\mathsf{X},\mathsf{Y}) : \mathsf{X},\mathsf{Y} \in \mathcal{S}^n, \begin{array}{l} \mathsf{B}_u^\perp (\mathsf{A}\mathsf{X} + \mathsf{X}\mathsf{A} + 2\alpha\mathsf{X})\, \mathsf{B}_u^{\perp T} < 0, \\ \mathsf{C}_y^{T\perp} \left(\mathsf{A}^T\mathsf{Y} + \mathsf{Y}\mathsf{A} + 2\alpha\mathsf{Y}\right) \mathsf{C}_y^{T\perp T} < 0 \end{array} \right\} \tag{3.31}$$

Note that $\Gamma_\alpha = \mathcal{X}_{\alpha,0} \times \mathcal{Y}_{\alpha,0}$.

We will also define the set of matrices that fulfills the coupling constraint:

$$\mathcal{Z} \triangleq \left\{ (\mathsf{X},\mathsf{Y}) \;\; : \mathsf{X},\mathsf{Y} \in \mathcal{S}^n, \begin{bmatrix} \mathsf{X} & I \\ I & \mathsf{Y} \end{bmatrix} \geq 0 \right\} \; .$$

The derivation given above completely eliminates the controller parameter from the conditions of existence. However, more recent research shows that it possible to skip the elimination, and in stead use a change of variables. We refer to [MOS95, Sch95, SGC97] for further details. This formulation is of great interest in multi objective design, but the formulation does not allow for a control order constraints.

Following the same lines for the $\mathcal{H}_\infty$ case we can obtain the following conditions for the existence of a suboptimal $\mathcal{H}_\infty$ controller: Find X and Y such that

$$\left[\begin{array}{c|c} \begin{bmatrix} \mathsf{B}_u \\ \mathsf{D}_{eu} \end{bmatrix}^\perp & 0 \\ \hline 0 & I \end{array}\right] \left[\begin{array}{cc|c} \mathsf{A}\mathsf{X} + \mathsf{X}\mathsf{A}^T & \mathsf{X}\mathsf{C}_e^T & \mathsf{B}_d \\ \mathsf{C}_e\mathsf{X} & -\gamma I & \mathsf{D}_{ed} \\ \hline \mathsf{B}_e^T & \mathsf{D}_{ed}^T & -\gamma I \end{array}\right] \left[\begin{array}{c|c} \begin{bmatrix} \mathsf{B}_u \\ \mathsf{D}_{eu} \end{bmatrix}^\perp & 0 \\ \hline 0 & I \end{array}\right]^T < 0 \tag{3.32}$$

$$\left[\begin{array}{c|c} \begin{bmatrix} \mathsf{C}_y^T \\ \mathsf{D}_{yw}^T \end{bmatrix}^\perp & 0 \\ \hline 0 & I \end{array}\right] \left[\begin{array}{cc|c} \mathsf{A}^T\mathsf{Y} + \mathsf{Y}\mathsf{A} & \mathsf{Y}\mathsf{B}_w & \mathsf{C}_z^T \\ \mathsf{B}_w^T\mathsf{Y} & -\gamma I & \mathsf{D}_{zw}^T \\ \hline \mathsf{C}_z & \mathsf{D}_{zw} & -\gamma I \end{array}\right] \left[\begin{array}{c|c} \begin{bmatrix} \mathsf{C}_y^T \\ \mathsf{D}_{yw}^T \end{bmatrix}^\perp & 0 \\ \hline 0 & I \end{array}\right]^T < 0 \tag{3.33}$$

$$\begin{bmatrix} \mathsf{X} & I \\ I & \mathsf{Y} \end{bmatrix} \geq 0 \; . \tag{3.34}$$

Note the above formulation is convex in both γ, X, and Y implying that we can find the optimal γ by solving the semidefinite programming problem: Minimize γ subject to equation (3.32), equation (3.33) and equation (3.34). As with the α stabilizing control problem we will define

$$\Gamma_{\mathcal{H}_\infty} \triangleq \{(\mathsf{X},\mathsf{Y}) : \mathsf{X},\mathsf{Y} \in \mathcal{S}^n \text{ and fulfills (3.32) and (3.33)}\}$$

The derivation done above for the α stabilizing and $\mathcal{H}_\infty$ control problems, can be extended to several other control problems. For instance the induced norm performance control problems for Energy to Peak, Energy to Euclidean Peak, Resource to Energy, and Euclidean Resource to Energy, see [Iwa93, SIG97], where as Euclidean Peak to Euclidean Peak can be found in [ANP96]. For an overview of control problems that can be treated in a seminar fashion, see [Iwa97a]. The work by Shafai, Uddin, Niemann and Stoustrup [SUNS96] gives an LMI approach to fixed order LTR controller. Instances of the $\mathcal{H}_\infty$ control problem and the E2EP and ER2ET control problem have more simple LMI formulations. These includes the model reduction and filtering problem, see [Gri95, GW96, GLS96]. We will take a look at the $\mathcal{H}_\infty$ model reduction problem:

Example 3.1 (Model reduction [Gri95]) *The model reduction problem is to reduce the order of the model, but such that the error between the reduced and the original model is minimized in some norm. Given a system Σ with input d and output e:*

$$\begin{bmatrix} \dot{x} \\ e \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ d \end{bmatrix}$$

we want to find another system Σ_m with input d and output $\hat{e}$, such that the induced norm from d to $e-\hat{e}$ is minimize for a given order n_m of Σ_m. If we for a minute disregard the model order we can formulate the model reduction problem as a standard $\mathcal{H}_\infty$ -control problem: Let Σ have output $y = d$ and input $u = -I\hat{e}$, that is we get a control setup on the form

$$\begin{bmatrix} \dot{x} \\ e \\ y \end{bmatrix} = \begin{bmatrix} A & B & 0 \\ C & D & -I \\ 0 & I & 0 \end{bmatrix} \begin{bmatrix} x \\ d \\ \hat{e} \end{bmatrix} \tag{3.35}$$

The zeros and identities in (3.35) makes the left annihilators in (3.32) and (3.33) fairly simple. Consider first the left annihilator in (3.32) then

$$\begin{aligned} \left[\begin{array}{c|c} \begin{bmatrix} \mathsf{B}_u \\ \mathsf{D}_{eu} \end{bmatrix}^{\perp} & 0 \\ \hline 0 & I \end{array}\right] &= \left[\begin{array}{c|c} \begin{bmatrix} 0 \\ I \end{bmatrix}^{\perp} & 0 \\ \hline 0 & I \end{array}\right] \\ &= \left[\begin{array}{c|c} \begin{bmatrix} I & 0 \end{bmatrix} & 0 \\ \hline 0 & I \end{array}\right] \\ &= \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & I \end{bmatrix} \end{aligned}$$

which simplifies (3.32) to

$$\begin{bmatrix} AX + XA^T & B \\ B^T & -\gamma I \end{bmatrix} < 0 \ .$$

Similarly we can simplify (3.33) to

$$\begin{bmatrix} A^TY + YA & C^T \\ C & -\gamma I \end{bmatrix} < 0 \ . \tag{3.36}$$

If we require the model order to be static $n_m = 0$ we want $Y = X^{-1}$. Imposing this on (3.36) we can apply a congruence transformation

$$M = \begin{bmatrix} X & 0 \\ 0 & I \end{bmatrix} > 0$$

and we get

$$M \begin{bmatrix} A^TX^{-1} + X^{-1}A & C^T \\ C & -\gamma I \end{bmatrix} M^T = \begin{bmatrix} XA^T + AX & XC^T \\ CX & -\gamma I \end{bmatrix} < 0 \ . \tag{3.37}$$

We can now write conditions for the existence of a static model with error less than γ *as*

$$X > 0 \tag{3.38}$$

$$\begin{bmatrix} XA^T + AX & XC^T \\ CX & -\gamma I \end{bmatrix} < 0 \tag{3.39}$$

$$\begin{bmatrix} AX + XA^T & B \\ B^T & -\gamma I \end{bmatrix} < 0 \ . \tag{3.40}$$

The optimal static model can be found by solving the following SDP

$$\begin{array}{ll} \underset{\gamma,X}{\text{minimize}} & \gamma \\ \text{subject to} & X > 0 \\ & \begin{bmatrix} XA^T + AX & XC^T \\ CX & -\gamma I \end{bmatrix} < 0 \\ & \begin{bmatrix} AX + XA^T & B \\ B^T & -\gamma I \end{bmatrix} < 0 \ . \end{array} \tag{3.41}$$

Conditions for the existence of a model of n_m*'th order can be derived using the results obtained in next section.*

The above formulations for the existence of a controller solving a given control problem rendered convex on the relaxation of the control order. In the next section we will return to formulations for fixed order control design.

3.3 Formulations for fixed order control design

In the last section we saw how we can reformulate the free order control problem as a convex optimization problem. The existence of a controller could be verified by finding a solution X, Y to an LMI in X and an LMI in Y, plus a coupling constraint.

In this section we will summarize a number of formulations of the fixed-order control problems, that lead to heuristic algorithms for the design. Some have the origin in the above derivations, but we will also present one extra approach, that recently has proven successful.

We will assume for simplification that $\tilde{B}_u\tilde{B}_u^T \not> 0$ and $\tilde{C}_y^T\tilde{C}_y \not> 0$. We will look at the α stabilizing control problem only.

The first formulation we present is derived directly from the elimination lemma. The second exploits that the control order is related to the rank of the coupling constraint. The third uses mutual existence of a state feedback gain, and a output injection gain.

Y, inverse Y problem

The formulation of the fixed order α stabilizing control problem given in (3.17) states that we need to find a matrix Y, such that it is positive definite, and belongs to a convex set, and its inverse belongs to a convex set. We state this in the following:

Lemma 3.3 (Y, inverse Y problem) *Let a scalar* α*, a desired control order* $\overline{n}_c$ *and matrices* A*,* B_u *and* C_y*. The matrices* $\tilde{A}$*,* $\tilde{B}_u$ *and* $\tilde{C}_y$ *are augmented with the control order as in (2.16). The following statements are equivalent* *(i) There exists a matrix* $G \in \mathbb{R}^{n_y+\overline{n}_c \times n_y+\overline{n}_c}$ *such that the closed loop* $A = \tilde{A} + \tilde{B}_u G \tilde{C}_y$ *is* α *stable.*

(ii) There exist $Y \in \mathcal{S}^{n+\overline{n}_c}$ *such that*

$$Y > 0,\ Y^{-1} \in \mathcal{X}_{\alpha,\overline{n}_c} \text{ and } Y \in \mathcal{Y}_{\alpha,\overline{n}_c} \tag{3.42}$$

(iii) There exist $X, Y \in \mathcal{S}^{n+\overline{n}_c}$ *such that*

$$X > 0,\ X \in \mathcal{X}_{\alpha,\overline{n}_c},\ \ Y > 0,\ Y \in \mathcal{Y}_{\alpha,\overline{n}_c} \text{ and } XY = I \tag{3.43}$$

Rank constraint

We will now formulate the fixed order control problem as an LMI problem plus an additional rank constraint.

Consider the rank of the coupling constraint

$$\begin{aligned}
\text{Rank}\begin{bmatrix} \mathsf{X} & I \\ I & \mathsf{Y} \end{bmatrix} &= \text{Rank}\begin{bmatrix} I & 0 \\ -\mathsf{X}^{-1} & I \end{bmatrix}\begin{bmatrix} \mathsf{X} & I \\ I & \mathsf{Y} \end{bmatrix}\begin{bmatrix} I & \mathsf{X}^{-1} \\ 0 & I \end{bmatrix} \\
&= \text{Rank}\begin{bmatrix} \mathsf{X} & 0 \\ 0 & \mathsf{Y}-\mathsf{X}^{-1} \end{bmatrix} \\
&\leq \text{Rank}(\mathsf{Y}-\mathsf{X}^{-1}) + \text{Rank}(\mathsf{X})
\end{aligned}$$

Suppose we are looking for a controller of order less than or equal to $\overline{n}_c$, then we want to restrict the possible X and Y with

$$\text{Rank}\left(\mathsf{X} - \mathsf{Y}^{-1}\right) \leq \overline{n}_c \tag{3.44}$$

which is equivalent to

$$\text{Rank}\begin{bmatrix} \mathsf{X} & I \\ I & \mathsf{Y} \end{bmatrix} \leq n + \overline{n}_c$$

Combining this with the convex results from last section we get the following lemma:

Lemma 3.4 (Rank constraint) *Let a scalar* α*, a desired control order* $\overline{n}_c$ *and matrices* A, B_u *and* C_y*. The matrices* $\tilde{A}$, $\tilde{B}_u$ *and* $\tilde{C}_y$ *are augmented with the control order as in (2.16). The following statements are equivalent*

(i) There exists a matrix $G \in \mathbb{R}^{n_y+n_c \times n_y+n_c}$ *such that the closed loop* $A = \tilde{A} + \tilde{B}_u G \tilde{C}_y$ *is* α *stable.*

(ii) There exists $\mathsf{X}, \mathsf{Y} \in \mathcal{S}^n$ *such that*

$$\begin{gathered}
\mathsf{B}_u^{\perp}\left(\mathsf{A}\mathsf{X} + \mathsf{X}\mathsf{A} + 2\alpha\mathsf{X}\right)\mathsf{B}_u^{\perp T} < 0 \\
\mathsf{C}_y^{T\perp}\left(\mathsf{A}^T\mathsf{Y} + \mathsf{Y}\mathsf{A} + 2\alpha\mathsf{Y}\right)\mathsf{C}_y^{T\perp T} < 0 \\
\begin{bmatrix} \mathsf{X} & I \\ I & \mathsf{Y} \end{bmatrix} \geq 0 \\
\text{Rank}\begin{bmatrix} \mathsf{X} & I \\ I & \mathsf{Y} \end{bmatrix} \leq n + \overline{n}_c.
\end{gathered} \tag{3.45}$$

The rank constraint in lemma 3.4 is however non-convex, which the following example shows:

Example 3.2 *Consider*

$$\mathsf{X}_1 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}, \mathsf{Y}_1 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

then the corresponding control order via equation (3.27) is 1, the same with

$$\mathsf{X}_2 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \mathsf{Y}_2 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} .$$

If the set was convex the midpoint between $\mathsf{X}_1, \mathsf{Y}_1$ *and* $\mathsf{X}_2, \mathsf{Y}_2$ *should also correspond to a controller of order 1, but instead it yields a controller of order 2.*

Define the in general non convex set

$$\mathcal{Z}_{\overline{n}_c} \triangleq \left\{ (\mathsf{X}, \mathsf{Y}) \;: \mathsf{X}, \mathsf{Y} \in \mathcal{S}^n,\ \begin{bmatrix} \mathsf{X} & I \\ I & \mathsf{Y} \end{bmatrix} \geq 0,\ \mathrm{Rank} \begin{bmatrix} \mathsf{X} & I \\ I & \mathsf{Y} \end{bmatrix} \leq n + \overline{n}_c \right\} .$$

Since $\mathcal{Z}_{\overline{n}_c}$ is more restrictive than $\mathcal{Z}$ then $\mathcal{Z}_{\overline{n}_c} \subset \mathcal{Z}$. The set $\mathcal{Z}_{\overline{n}_c}$ is convex if and only if $\overline{n}_c = n$, and in this case $\mathcal{Z}_n = \mathcal{Z}$. If $\overline{n}_c < n$ then $\mathcal{Z}_{\overline{n}_c}$ lies in the boundary of $\mathcal{Z}$, and in fact $\mathcal{Z}_{n-1} = \partial \mathcal{Z}_n$. If we can find X, Y on the boundary of $\mathcal{Z}_n$, we are guaranteed to have a controller of order one less than the plant. This can in many cases be done by minimizing $\mathrm{Tr}\,(\mathsf{X} + \mathsf{Y})$ subject to the normal constraints, since the coupling constraint is usually the strongest. That is solving the problem:

$$\begin{array}{ll} \underset{\mathsf{X},\mathsf{Y}}{\text{Minimize}} & \mathrm{Tr}(\mathsf{X} + \mathsf{Y}) \\ \text{subject to} & \mathsf{B}_u^{\perp} \left(\mathsf{AX} + \mathsf{XA} + 2\alpha \mathsf{X} \right) \mathsf{B}_u^{\perp T} < 0 \\ & \mathsf{C}_y^{T\perp} \left(\mathsf{A}^T \mathsf{Y} + \mathsf{YA} + 2\alpha \mathsf{Y} \right) \mathsf{C}_y^{T\perp T} < 0 \\ & \begin{bmatrix} \mathsf{X} & I \\ I & \mathsf{Y} \end{bmatrix} \geq 0. \end{array} \tag{3.46}$$

We will return to a deeper study of this formulation in chapter 4.

Dual formulation

The formulations presented above relates the existence of a controller obtaining the α stability with the existence of a Lyapunov matrix. However a preliminary step could be to relate the existence of a output feedback controller with the existence of a state feedback gain and output injection gain achieving the desired goal. First we will see that $X \in \mathcal{X}_{\alpha,n_c}, X > 0$ is equivalent of the existence of a state feedback gain that achieves α stability for the closed loop. The constraint that $X \in \mathcal{X}_{\alpha,n_c}$ is by definition that X fulfills

$$\tilde{B}_u^{\perp} \left(X\tilde{A}^T + \tilde{A}X + 2\alpha X \right) \tilde{B}_u^{\perp T} < 0,\ X > 0$$

which via the elimination lemma with $F = \tilde{B}_u$ and $H = X$ is equivalent to the existence of a K such that

$$(\tilde{A} + \tilde{B}_u K)X + X(\tilde{A} + \tilde{B}_u K)^T < -2\alpha X,\ X > 0$$

Similarly we can get an equivalence between $Y \in \mathcal{Y}_{\alpha,n_c}, Y > 0$ and the existence of an output injection gain F such that

$$Y(\tilde{A} + F\tilde{C}_y) + (\tilde{A} + F\tilde{C}_y)^T Y < -2\alpha Y,\ Y > 0$$

We now have the following (see [Iwa93]):

Lemma 3.5 (Dual formulation) *Let a scalar α, a desired control order $\overline{n}_c$ and matrices A, B_u and C_y be given. The matrices $\tilde{A}$, $\tilde{B}_u$ and $\tilde{C}_y$ are augmented with the control order as in (2.16). The following statements are equivalent*

(i) There exists a matrix G such that the closed loop $A = \tilde{A} + \tilde{B}_u G \tilde{C}_y$ is α stable.

(ii) There exist $K \in \mathbb{R}^{n_u \times (n+n_c)}$, $F \in \mathbb{R}^{(n+n_c) \times n_y}$ and $0 < X \in \mathcal{S}^{n+n_c}$

$$
\begin{aligned}
(\tilde{A} + \tilde{B}_u K)X + X(\tilde{A} + \tilde{B}_u K)^T &< -2\alpha X \\
(\tilde{A} + F\tilde{C}_y)X + X(\tilde{A} + F\tilde{C}_y)^T &< -2\alpha X
\end{aligned}
$$

(iii) There exist $K \in \mathbb{R}^{n_u \times (n+n_c)}$ and $0 < X \in \mathcal{S}^{n+n_c}$

$$
\begin{aligned}
\tilde{B}_u^{\perp}(\tilde{A}X + X\tilde{A} + 2\alpha X)\tilde{B}_u^{\perp T} &< 0 \\
(\tilde{A} + F\tilde{C}_y)X + X(\tilde{A} + F\tilde{C}_y)^T &< -2\alpha X
\end{aligned}
$$

(iv) There exist $K \in \mathbb{R}^{n_u \times (n+n_c)}$, $F \in \mathbb{R}^{(n+n_c) \times n_y}$ and $0 < Y \in \mathcal{S}^{n+n_c}$

$$
\begin{aligned}
Y(\tilde{A} + \tilde{B}_u K) + (\tilde{A} + \tilde{B}_u K)^T Y &< -2\alpha Y \\
Y(\tilde{A} + F\tilde{C}_y) + (\tilde{A} + F\tilde{C}_y)^T Y &< -2\alpha Y
\end{aligned}
$$

(v) There exist $K \in \mathbb{R}^{n_u \times (n+n_c)}$ and $0 < Y \in \mathcal{S}^{n+n_c}$

$$
\begin{aligned}
Y(\tilde{A} + \tilde{B}_u K) + (\tilde{A} + \tilde{B}_u K)^T Y &< -2\alpha Y \\
\tilde{C}_y^{T\perp}\left(\tilde{A}^T Y + Y\tilde{A} + 2\alpha Y\right)\tilde{C}_y^{T\perp T} &< 0
\end{aligned}
$$

Matrix X fulfilling (ii) or (iii) in the above lemma is the Lyapunov matrix for the closed loop, and by using the dual formulation of (3.4) we can find the controller parameter G. Similarly we can use Y fulfilling (iv) or (v) inserted in (3.4) to find the controller parameter.

3.4 Implementation issues

In this section we will discuss some issues related to implementation of the above LMI formulations of the above control problem. Several steps in the process of computing a controller from the state space matrices and design specifications to final controller requires careful numerical considerations. For control problems in general a collection of articles on numerical linear algebra for systems and control is given in [PLD94]. We will especially consider numerical issues related to the determination of the right control order and computation of a "good" Lyapunov function.

Computing the Lyapunov matrix

Finding X and Y in the intersection of Γ_α and $\mathcal{Z}$ is equivalent to the existence of a controller solving the α stabilizing control problem. The control order was according to theorem 3.1 equal to the rank of $\mathsf{X} - \mathsf{Y}^{-1}$. However, the construction of $Y \in \mathcal{S}^{n+n_c}$ as it was used in the proof, is not very adequate for numerical implementation. The Lyapunov function is related to the state space realization of the system and the controller. Choosing $Y_{22} = I_{n_c}$ as suggested in the proof fixes the state space representation in the controller, with out relations to the representation in the system. If for instance the norm of Y is big, then the condition number Y will be worse than $||\mathsf{Y}||$, and thus making the computation of the controller parameter difficult.

Instead the following procedure should be used.

Compute the singular value decomposition of $I - XY$ as

$$I - \mathsf{X}\mathsf{Y} = U\Sigma V^T \quad .$$

Since $I - \mathsf{X}\mathsf{Y}$ by definition has rank n_c, and since $\text{Rank}(\mathsf{X} - \mathsf{Y}^{-1}) = \text{Rank}(\mathsf{X}\mathsf{Y} - I)$ choosing the first n_c columns of U and V, and the first n_c rows and columns of Σ, denoted $U_{n_c}, V_{n_c}, \Sigma_{n_c}$, is sufficient for

$$I - \mathsf{X}\mathsf{Y} = U_{n_c}\Sigma_{n_c}V_{n_c}^T$$

The following choice of $N, M \in \mathbb{R}^{n \times n_c}$, as

$$M = U_{n_c}\Sigma_{n_c}^{1/2}$$
$$N = V_{n_c}\Sigma_{n_c}^{1/2}$$

fulfills $MN^T = I - XY$. One candidate for the Lyapunov matrix Y can now be computed from the relation

$$\begin{bmatrix} \mathsf{Y} & I \\ N^T & 0 \end{bmatrix} = Y \begin{bmatrix} I & \mathsf{X} \\ 0 & M^T \end{bmatrix} \quad , \tag{3.47}$$

which provides

$$Y = \begin{bmatrix} \mathsf{Y} & N \\ N^T & Y_{22} \end{bmatrix} \text{ and } Y^{-1} = \begin{bmatrix} \mathsf{X} & M \\ M^T & * \end{bmatrix} \quad . \tag{3.48}$$

To see this do the following partition of Y as

$$Y = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix}$$

then the right hand side of (3.47) becomes

$$\begin{bmatrix} Y_{11} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} \begin{bmatrix} I & \mathsf{X} \\ 0 & M^T \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{11}\mathsf{X} + Y_{12}M^T \\ Y_{12}^T & Y_{12}^T\mathsf{X} + Y_{22}M^T \end{bmatrix}$$

which gives the following four equations

$$\begin{aligned} Y_{11} &= \mathsf{Y} \\ Y_{12}^T &= N^T \\ I &= Y_{11}\mathsf{X} + Y_{12}M^T \\ 0 &= Y_{12}^T\mathsf{X} + Y_{22}M^T \quad . \end{aligned}$$

Inserting the result of the first two equations in the third we get $I = \mathsf{Y}\mathsf{X} + NM^T$, which holds by $(I - XY)^T = NM^T$. We can now find Y_{22} by solving the fourth equation:

$$\begin{aligned} 0 &= N^T\mathsf{X} + Y_{22}M^T \\ &= N^T\mathsf{X} + Y_{22}\Sigma_{n_c}^{1/2}U_{n_c} \end{aligned}$$

which gives

$$Y_{22} = \Sigma^{1/2} V_{n_c}^T \mathsf{X} U_{n_c}^T \Sigma^{-1/2}$$

It can be shown that Y_{22} is positive definite, implying that $Y > 0$. Similarly it can be shown that Y^{-1} by (3.47) has X in the upper left corner.

Removing small parts of X

The upper procedure gave directions for a sound computation of Y. However, there is still one hatch, numerically the rank of a matrix is computed as the number of singular values above a certain threshold. Due to this it is difficult to determine the rank. For our purposes the rank of the coupling constraint is not so important, what is more important is if the Lyapunov matrix we construct do in fact make the associated BMI feasible, that is lemma 3.3 holds. In fact it is possible that we can remove some parts of X and still make the new $\hat{\mathsf{X}}$ be in $\mathcal{X}_{\alpha,0}$, and such that $(\hat{\mathsf{X}}, \hat{Y}) \in \mathcal{Z}_{n_c}$.

We formalize this in the following theorem

Theorem 3.2 *Let* X, Y *fulfill*

$$\begin{bmatrix} \mathsf{X} & I \\ I & \mathsf{Y} \end{bmatrix} \geq 0$$

then for any n_c*,* $0 \leq n_c \leq \operatorname{Rank}\left(X - Y^{-1}\right)$ *there exists* $L \in \mathcal{S}^n$*,* $Y \in \mathcal{S}^{n+n_c}$ *such that*

$$\begin{aligned} L &\geq 0 \\ \begin{bmatrix} \mathsf{X} - L & I \\ I & \mathsf{Y} \end{bmatrix} &\geq 0, \\ n_c &= \operatorname{Rank}\left(\mathsf{X} - L - \mathsf{Y}^{-1}\right), \\ Y &= \begin{bmatrix} \mathsf{Y} & * \\ * & * \end{bmatrix} > 0, \\ Y^{-1} &= \begin{bmatrix} \mathsf{X} - L & * \\ * & * \end{bmatrix} \end{aligned}$$

Proof:

Suppose the coupling constraint holds, then $X - Y^{-1}$ is symmetric and positive semidefinite. Since $X - Y^{-1}$ is symmetric, we can consider its eigenvalue decomposition:

$$X - Y^{-1} = V \Lambda V^T$$

where $\Lambda = \operatorname{Diag} \lambda_1, \lambda_2, \cdots, \lambda_n$ with $\lambda_1 \geq \lambda_2 \geq, \cdots, \lambda_n$. Note that $\Lambda \geq 0$. Partition Λ in the n_c biggest eigenvalues and the rest:

$$\begin{aligned} \mathsf{X} - \mathsf{Y}^{-1} = V \Lambda V^T &= \begin{bmatrix} V_{n_c} \overline{V}_{n_c} \end{bmatrix} \begin{bmatrix} \Lambda_{n_c} & 0 \\ 0 & \overline{\Lambda}_{n_c} \end{bmatrix} \begin{bmatrix} V_{n_c} \overline{V}_{n_c} \end{bmatrix}^T \\ &= V_{n_c} \Lambda_{n_c} V_{n_c}^T + \underbrace{\overline{V}_{n_c} \overline{\Lambda}_{n_c} \overline{V}_{n_c}^T}_{L} \end{aligned}$$

where V_{n_c} are the first n_c columns of V. The diagonal matrix Σ_{n_c} contains the n_c largest eigenvalues. We have defined

$$L \triangleq \overline{V}_{n_c} \overline{\Lambda}_{n_c} \overline{V}_{n_c}^T \quad .$$

It follows that L is positive definite and that

$$(\mathsf{X} - L) - \mathsf{Y}^{-1} = V_{n_c} \Lambda_{n_c} V_{n_c}^T \geq 0 \quad ,$$

with

$$\mathrm{Rank}\left(\mathsf{X} - L - \mathsf{Y}^{-1}\right) = \mathrm{Rank}\, \Sigma_{n_c} = n_c \quad .$$

Since $\mathsf{Y} > 0$ we have

$$\begin{bmatrix} \mathsf{X} - L & I \\ I & \mathsf{Y} \end{bmatrix} \geq 0 \quad . \tag{3.49}$$

Using theorem 3.1 we know that

$$\begin{aligned} Y &= \begin{bmatrix} \mathsf{Y} & * \\ * & * \end{bmatrix} > 0 \quad , \\ Y^{-1} &= \begin{bmatrix} \mathsf{X} - L & * \\ * & * \end{bmatrix} \end{aligned}$$

holds. This proves the theorem. •

The above theorem states that we can remove parts L from X with $\hat{\mathsf{X}} = \mathsf{X} - L$, and still we can construct a positive definite Lyapunov matrix Y, that has Y in the upper left part, and $\hat{\mathsf{X}}$ in the upper left part of its inverse.

Computation of a controller

Suppose we think a controller of order n_c is sufficient and $(\mathsf{X}, \mathsf{Y}) \in \Gamma_\alpha \cap \mathcal{Z}$, then we use theorem 3.2 to remove an appropriate part of X, and the relation 3.47 provides an equation for computing the Lyapunov matrix Y. We have now solved the n_c fixed order α stabilizing control problem if $(\hat{\mathsf{X}}, \mathsf{Y}) \in \Gamma_\alpha \cap \mathcal{Z}_{n_c}$. The construction of $\hat{\mathsf{X}}$ guarantees that $(\hat{\mathsf{X}}, \mathsf{Y}) \mathcal{Z}_{n_c}$, and $(\mathsf{X}, \mathsf{Y}) \in \Gamma_\alpha$ reduces to examine if

$$\hat{\mathsf{X}} \in \mathcal{X}_{\alpha,0} \quad . \tag{3.50}$$

If (3.50) holds then a controller can now be computed by inserting the Lyapunov matrix Y in the BMI equation (3.9), and solving for G. This is a LMI feasibility problem, and can be solved as such. However algebraic solutions can be derived [GA94, Iwa93], but are in numerical implementation usually ill conditioned, as pointed out in [GA94]. Also due to numerical problems the LMI in G, equation (3.9), might even be infeasible. However, this problem can be circumvented by solving the following SDP problem:

$$\begin{array}{ll} \underset{G,l}{\text{minimize}} & l \\ \text{subject to} & F_{\alpha,n_c} G F_{\alpha,n_c} + (F_{\alpha,n_c} G F_{\alpha,n_c})^T + Q_{\alpha,n_c} - lI < 0 \quad . \end{array}$$

If $l \leq 0$ then G solves the problem, but even though $l > 0$ then the problem might be solved anyway. The G might be a solution to the problem, but with a different Lyapunov matrix. We will not go into deeper numerical technicalities.

The above described procedure for designing controllers have been implemented in [BG97] as a top on the software by Boyd and Vandenberghe [BV94]. The MATLAB LMI control toolbox [GNLC95] offers similarly design procedure. New LMI formulations can be tested in [GDN95, GNLC95, WB96].

Part II

Solvers and optimization

Chapter 4

Control order reduction

In this chapter we will consider a heuristic algorithm for design of controllers of low order. The algorithm uses what is called alternating projections, and we will explain this in great detail. This is not the only way of solving the low-order control problem. Several researchers have consider the same problem over the last three decades. We stress that it is still an open problem of finding necessary and sufficient conditions for the existence of a static output feedback stabilizer with out using heavy mathematics like algebraic. In fact this problem was mentioned as one of the major open problems in the survey [BGL95].

In chapter 3 we presented a series of matrix inequality formulations for the existence of a fixed order α stabilizing controller. In lemma 3.1 we presented an approach based on a BMI in the controller parameter and the Lyapunov matrix. Applying the elimination lemma to the BMI result we obtained the result presented in lemma 3.3. Exploiting that the control order is the rank of the coupling constraint we get lemma 3.4. It was possible to reformulate the constraints from the existence of a free order controller to be constraints on the existence of a state feedback gain/output injection gain, and we got lemma 3.5.

It has been shown by Blondel and Tsitsiklis, that the problem of finding a controller of low-order is NP hard, if one constrains the controller in a prescribed area, see [BT95].

Our main goal in this chapter is to develop heuristic algorithms, that will compress the control order and obtain optimal α stability. Several methods have achieved attention over the last years. The formulation lemma 3.3 has been used by Iwasaki and Skelton, see [IS95], to derive an algorithm that could search for controllers of fixed order. The approach used the analytical center approach to make $X = Y^{-1}$. Note, that recently Fu and Luo have showed that the formulation lemma 3.3(iii) leads to an NP hard problem, see [FL97].

Another method based on lemma 3.5 was also presented by Iwasaki, see [Iwa97b]. This method was based on solving generalized eigenvalue problems.

The rank formulation lemma 3.4 has been studied by a series of authors. First approach was made by Grigoriadis in his ph.d. thesis, and presented in [GS94, GS96]. His approach was based on alternating projections, and we will study this approach in greater details in this chapter. El Ghaoui, Outstry and Aitrami developed a linearization algorithm [GOA95], where the rank of the coupling was reduced using an attracting function. The approach seem to work very well in practice, see [Kri97]. The rank problem can in some cases be reduced exactly, as it was shown by Mesbahi and Papavassilopoulos in [MP97a, MP97b].

4.1 Problem statement

In this chapter we will consider an algorithm that tries to solve the following problem

$$\begin{array}{ll} \underset{X,Y}{\text{Minimize}} & \text{rank} \begin{bmatrix} X & I \\ I & Y \end{bmatrix} \\ \text{subject to} & X, Y \in \Gamma_{\text{convex}} \cap \mathcal{Z} \end{array} \tag{4.1}$$

where Γ_{convex} is dependent on the control problem considered. See the definitions in last chapter.

As described in last chapter the rank of a matrix is not a convex constraint, and the problem (4.1) is supposely difficult to solve. We will however try to solve the problem by using a technique called alternating projection, presented in the next section. The alternating projection can be implemented effectively using semidefinite programming as we shall see in the subsequent section. We then proceed with a description of adequate divergence and convergence criteria. In the end we give a few essential examples.

4.2 Alternating Projecting Schemes

Based on the formulation provided in Section 2, the fixed-order control design problem reduces to a feasibility problem of obtaining a matrix pair that satisfies a family of LMIs and a coupling matrix rank constraint. In this section, alternating projection methods are presented for the solution of this type of feasibility problems. These method exploit the geometry of the design space to find feasible solutions and they have been used successfully to address image restoration and statistical estimation problems in signal processing [CT90, YW82, WA86]. Both the standard alternating projection method and the directional alternating projection method are presented [BD86, Han88, GPR67].

4.2.1 The Standard Alternating Projection Method

Consider a family $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m$ of closed, convex sets in the space of symmetric matrices. We suppose that the sets have a nonempty intersection and we seek to solve the feasibility problem of finding a matrix in the intersection $\mathcal{C}_1 \bigcap \mathcal{C}_2 \bigcap \ldots \bigcap \mathcal{C}_m$. Let $P_{\mathcal{C}_i}$ denote the orthogonal projection operator onto the set $\mathcal{C}_i$, where $i = 1, \ldots, m$. Hence, for any $n \times n$ symmetric matrix X, the matrix $P_{\mathcal{C}_i}(X)$ denotes the orthogonal projection of X onto $\mathcal{C}_i$, that is the matrix in $\mathcal{C}_i$ which has minimum distance from the matrix X. The orthogonal projection theorem [Lue68] guarantees that this projection is unique. The question we would like to answer is the following: Is it possible to provide a solution to the feasibility problem by making use of the orthogonal projections onto each constraint set? The answer is yes, and is provided by the following result which we call the *Standard Alternating Projection Theorem* [CG59, GPR67].

Theorem 4.1 *Let X_0 be any $n \times n$ symmetric matrix and $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m$ be a family of closed, convex sets in the space of symmetric matrices. Then, if there exists an intersection, the sequence* *of alternating projections*

$$\begin{aligned} X_1 &= P_{C_1} X_0 \\ X_2 &= P_{C_2} X_1 \\ &\vdots \\ X_m &= P_{C_m} X_{m-1} \\ &\vdots \\ X_{2m} &= P_{C_1} X_{2m-1} \\ X_{2m+1} &= P_{C_2} X_{2m} \\ &\vdots \\ X_{3m} &= P_{C_m} X_{3m-1} \\ &\vdots \end{aligned} \tag{4.2}$$

converges to a point in the intersection of the sets, i.e. $X_i \to X$ *where* $X \in C_1 \bigcap C_2 \bigcap \ldots \bigcap C_m$. *If no intersection exists, the sequence does not converge.*

Hence, starting from any symmetric matrix, the sequence of alternating projections onto the constraint sets converges to a solution of the feasibility problem, if one exists. The case of no intersection can be detected by examining the convergence of the even and odd subsequences of the above sequence of alternating projections. A schematic representation of the Standard Alternating Projection Method is shown in Fig. 4.1. It can be easily verified that the limit X of the alternating projection sequence depends on the starting point X_0, as well as the order of the projections. Hence, by rearranging the sequence of projections we can obtain a different feasible point.

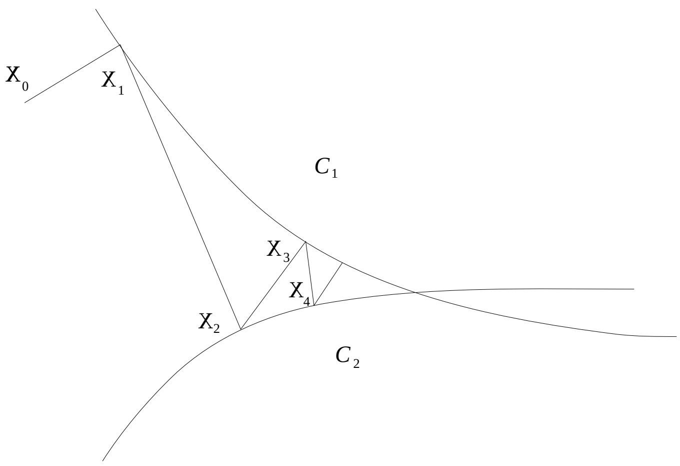


Figure 4.1: Standard Alternating Convex Projection Algorithm

An important feature of the standard alternating projection algorithms (4.2) is that the algorithms can be implemented very easily and usually the amount of calculations in one iteration is very small. However, in some cases the algorithms may suffer from slow convergence. For example, consider the case of two planes intersecting with a small angle. In this case the standard alternating projection algorithm (4.2) might oscillate for many iterations between the two sets before it converges to a point in the intersection. An effective remedy is often obtained by *Directional Alternating Convex Projection* Algorithm, described below [GPR67].

4.2.2 The Directional Alternating Projection Method

The directional alternating projection method uses information about the geometry of the constraint sets to provide an algorithm with accelerated convergence to solve the matrix feasibility problem. The basic idea behind this approach is to utilize in each iteration the tangent plane of one of the constraint sets, so that the sequence of points we obtain approaches the intersection of the sets more rapidly (see Fig. 4.2). For simplicity, we will consider the case of two closed and convex constraint sets $\mathcal{C}_1$ and $\mathcal{C}_2$. The *Directional Alternating Convex Projection* algorithm is described next, where $\langle X, Y\rangle$ denotes the inner product of two matrices X and Y.

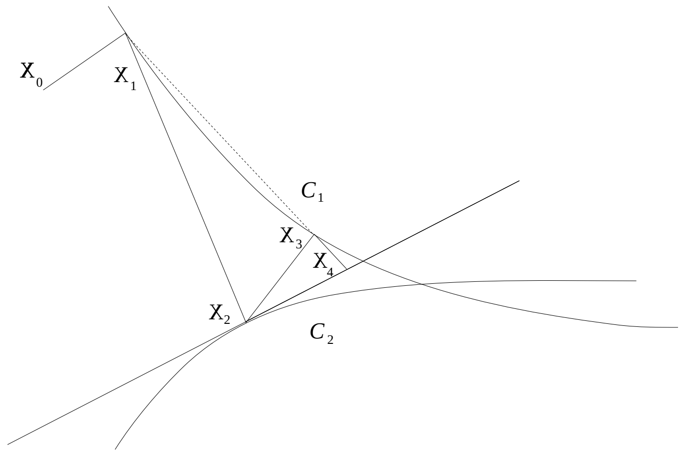


Figure 4.2: Directional Alternating Projection Algorithm

Theorem 4.2 *Let X_0 be any $n \times n$ symmetric matrix. Then the sequence of matrices $\{X_i\}$, $i = 1, 2, \ldots, \infty$ given by*

$$\begin{aligned} X_1 = P_{\mathcal{C}_1} X_0,\ \ X_2 = P_{\mathcal{C}_2} X_1,\ \ X_3 = P_{\mathcal{C}_1} X_2 \\ X_4 = X_1 + \lambda_1 (X_3 - X_1),\ \ \lambda_1 = \frac{\|X_1 - X_2\|_F{}^2}{\langle X_1 - X_3, X_1 - X_2\rangle} \\ X_5 = P_{\mathcal{C}_1} X_4,\ \ X_6 = P_{\mathcal{C}_2} X_5,\ \ X_7 = P_{\mathcal{C}_1} X_6 \\ X_8 = X_5 + \lambda_2 (X_7 - X_5),\ \ \lambda_2 = \frac{\|X_5 - X_6\|_F{}^2}{\langle X_5 - X_7, X_5 - X_6\rangle} \\ \vdots \end{aligned} \tag{4.3}$$

converges to a point in the intersection of the sets $\mathcal{C}_1$ and $\mathcal{C}_2$.

Hence, starting from any symmetric matrix, the sequence of directional alternating projections (4.3) provides an accelerated numerical algorithm to obtain a feasible matrix in the intersection of the constraint sets $\mathcal{C}_1 \bigcap \mathcal{C}_2$. In fact, it can be easily verified that when the two sets $\mathcal{C}_1$ and $\mathcal{C}_2$ are hyper planes in the space of symmetric matrices then the alternating projection algorithm converges to a feasible point in one cycle, independently of the angle between the two hyper planes.

4.3 Mixed AP/SDP Design for Low-order Control.

In this section we describe how to solve low-order control problems by exploiting the alternating projection technique described in the previous section and efficient semidefinite programming (SDP) algorithms.

Recall that the fixed-order control design problem has a solution if and only if there exist a matrix pair (X,Y) in the intersection of a family of LMI constraint sets and a rank constraint set.

Lets denote by Γ_{convex} the LMI constraint that the matrices X and Y need to satisfy and define the following constraint set $\mathcal{Z}_{n_c}$ presented in last chapter. Hence, a necessary and sufficient condition for the existence of a controller for fixed order n_c is that there exists $(X,Y) \in \Gamma_{\text{convex}} \cap \mathcal{Z}_{n_c}$. The set $\mathcal{Z}_{n_c}$ that restricts the controller order is a non-convex constraint set. Also, notice that, depending on the specific control design problem, the set Γ_{convex} corresponds to the set Γ_α or the set $\Gamma_{\mathcal{H}_\infty}$ defined in chapter 3.

The alternating projection scheme presented in the last section can now be used to find a matrix pair (X,Y) in the intersection of the convex constraint set Γ_{convex} and the non-convex rank constraint set $\mathcal{Z}_{n_c}$. Our first task is to compute the projections onto the convex constraint set Γ_α or $\Gamma_{\mathcal{H}_\infty}$. One approach is to decompose the LMI constraints as intersections of sets of simpler geometry and to find analytical expressions for the projection operators onto these simpler sets. This approach was followed in [GS96]. However, the iterative projections onto these multiple sets require a large number of iterations for convergence resulting in computationally expensive algorithms. Alternatively, the projection onto the set Γ_α or $\Gamma_{\mathcal{H}_\infty}$ can be computed using SDP for convex optimization. Hence, the multiple projections required in [GS96] onto the set Γ_{convex} are eliminated and faster convergence can be achieved.

The following result, see [BGFB94], provides the projection onto a general LMI constraint set Γ as the solution to an SDP problem

Proposition 4.1 *Let Γ be a convex set described by an LMI. Then the projection $X^* = \mathcal{P}_\Gamma X$ can be computed as the unique solution Y to the following SDP problem*

$$
\begin{array}{ll}
\underset{S,X,Y}{\textit{minimize}} & \textit{trace}\ S \\
\textit{subject to} & \begin{bmatrix} S & Y-X \\ Y-X & I \end{bmatrix} \geq 0, \\
& Y \in \Gamma \quad \textit{and} \\
& S \in \mathcal{S}^n
\end{array}
$$

In our problem our objective is to compute the projection onto the joint set $\Gamma_{\text{convex}} \subset \mathcal{S}^n \times \mathcal{S}^n$ of matrix pairs (X,Y). This projection can be found by solving the following SDP problem, where (X_0,Y_0) are the given matrices that we seek the projection and $X,Y,S,T \in \mathcal{S}^n$ are the free variables

$$
\begin{array}{ll}
\underset{S,T,X,Y}{\text{minimize}} & \text{Trace } (T+S) \\
\text{subject to} & \begin{bmatrix} T & (X-X_0) \\ (X-X_0) & I \end{bmatrix} \geq 0 \\
& \begin{bmatrix} S & (Y-Y_0) \\ (Y-Y_0) & I \end{bmatrix} \geq 0 \\
& (X,Y) \in \Gamma_{\text{convex}},\ T,S \in \mathcal{S}^n
\end{array}
\tag{4.4}
$$

We denote the minimizing solutions by (X^*,Y^*), that is the projection onto Γ_{convex} is written as

$$(X^*,Y^*) = P_{\Gamma_{\text{convex}}}(X_0,Y_0).$$

In addition to the above LMI constraints sets, we seek to compute the orthogonal projection onto the nonconvex constraint set $\mathcal{Z}_{n_c}$. To this end, define the following sets in the space of symmetric matrices

$$\mathcal{D} \stackrel{\Delta}{=} \left\{ Z \in \mathcal{S}^{2n} : Z = \begin{bmatrix} X & 0 \\ 0 & Y \end{bmatrix},\ \ X,Y \in \mathcal{S}^n \right\} \tag{4.5}$$

$$\mathcal{P} \stackrel{\Delta}{=} \{ Z \in \mathcal{S}^{2n} : Z \geq -J \} \tag{4.6}$$

$$\mathcal{R}_k \stackrel{\Delta}{=} \{ Z \in \mathcal{S}^{2n} : \text{rank } (Z+J) \leq k \} \tag{4.7}$$

where $k = n + n_c$ and

$$J = \begin{bmatrix} 0 & I_n \\ I_n & 0 \end{bmatrix} \in \mathcal{S}^{2n}. \tag{4.8}$$

Then, the connection between $\mathcal{Z}_{n_c}$ and $\mathcal{D}$, $\mathcal{P}$, and $\mathcal{R}_k$ is the following

$$(X,Y) \in \mathcal{Z}_{n_c} \Leftrightarrow \begin{bmatrix} X & 0 \\ 0 & Y \end{bmatrix} \in \mathcal{D} \cap \mathcal{P} \cap \mathcal{R}_{n+n_c}$$

Notice that the sets $\mathcal{D}$ and $\mathcal{P}$ are closed convex sets, where as $\mathcal{R}_k$ is the only non-convex set.

The expressions for the orthogonal projections onto these sets are provided next.

Theorem 4.3 *Let*

$$Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{12}^T & Z_{22} \end{bmatrix} \in \mathcal{S}^{2n}. \tag{4.9}$$

The orthogonal projection, $Z^* = P_{\mathcal{D}}Z$, of Z onto the set $\mathcal{D}$ is given by

$$Z^* = \begin{bmatrix} Z_{11} & 0 \\ 0 & Z_{22} \end{bmatrix} \in \mathcal{S}^{2n}. \tag{4.10}$$

The orthogonal projection onto the set $\mathcal{P}$ is provided by the following result which follows from [Hig88].

Theorem 4.4 *Let $Z \in \mathcal{S}^n$ and let $Z + J = L\Lambda L^T$ be the eigenvalue-eigenvector decomposition of $Z + J$ where Λ is the diagonal matrix of the eigenvalues and L is the orthogonal matrix of the normalized eigenvectors. The orthogonal projection, $Z^* = P_{\mathcal{P}}Z$, of Z onto the set $\mathcal{P}$ is given by*

$$Z^* = L\Lambda_- L^T - J \tag{4.11}$$

where Λ_- is the diagonal matrix obtained by replacing the negative eigenvalues in Λ by zero.

Hence, this projection requires an eigenvalue-eigenvector decomposition of the $2n \times 2n$ symmetric matrix $Z+J$.

We note that the rank constraint set $\mathcal{R}_k$, defined by (4.7), is a closed set, but it is not convex. Therefore, given a matrix Z in $\mathcal{S}^{2n}$, there might be several matrices in $\mathcal{R}_k$ which minimize the distance from Z. We will call any such matrix a projection of Z on $\mathcal{R}_k$. The following result provides a projection onto the set $\mathcal{R}_k$ [HJ91].

Theorem 4.5 *Let $Z \in \mathcal{S}^n$ and let $Z+J = U\Sigma V^T$ be a singular value decomposition of $Z+J$. The orthogonal projection, $Z^* = P_{\mathcal{R}_k} Z$, of Z onto the set $\mathcal{R}_k$ is given by*

$$Z^* = U\Sigma_k V^T - J \tag{4.12}$$

where Σ_k is the diagonal matrix obtained by replacing the smallest $2n-k = n-n_c$ singular values in $Z+J$ by zero.

Notice that the sequence of the projections onto the two sets $\mathcal{P}$ and $\mathcal{R}_k$ can be computed in one step via the eigenvalue-eigenvector decomposition of $Z+J$ followed by zeroing the appropriate number of the smallest eigenvalues. If we denote this sequence of projections by $P_{\mathcal{P}\mathcal{R}_k} Z$ then the directional alternating projection can be used to find the projection onto the set $\mathcal{Z}_{n_c}$ via the following sequence of iterations

$$\begin{aligned}
Z_i^a &= P_{\mathcal{P}\mathcal{R}_k} Z_{i-1} \\
Z_i^b &= P_{\mathcal{D}} Z_a \\
Z_i^c &= P_{\mathcal{P}\mathcal{R}_k} Z_b \\
Z_i &= Z_i^a + \lambda_i (Z_i^c - Z_i^a), \;\; \lambda_i = \frac{||Z_i^a - Z_i^b||_F{}^2}{\langle Z_i^a - Z_i^c, Z_i^a - Z_i^b \rangle}
\end{aligned} \tag{4.13}$$

We will call each step of the above procedure for an *inner iteration*, as oppose to the outer iterations given below. Hence, the above running a series of inner iteration provides the projection $P_{\mathcal{Z}_{n_c}}(X,Y)$ of (X,Y) onto $\mathcal{Z}_{n_c}$.

The alternating projection algorithm for the fixed order control problem can now be programmed utilizing SDP for the projection onto Γ_{convex} and the above inner iteration scheme for the projection onto $\mathcal{Z}_{n_c}$. The proposed procedure is the following: First find a solution that corresponds to a full-order controller. This is simply done by solving an LMI feasibility problem for the constraint set Γ_{convex}. Next, obtain a solution that corresponds to a controller of order at most $n_c - 1$. This can be done via the SDP problem

$$\begin{array}{ll}
\underset{X,Y}{\text{minimize}} & \text{Tr}(X+Y) \\
\text{subject to} & (X,Y) \in \Gamma_{\text{convex}} \cap \mathcal{Z}
\end{array} \tag{4.14}$$

The obtained solution will be the starting point for our alternating projection algorithm. The steps of this alternating projection algorithm are as follows. Notice that in order to have fast convergence the directional alternating projection algorithm is used as mentioned in Section 4.2.2.

Step 1 Solve the SDP problem (4.14) that corresponds to a controller of order at most $n_c = n-1$. The solution (X_0, Y_0) will be our starting point.

Step 2 Consider the problem where the controller order is reduced by one, i.e. set $n_c = n_c - 1$. Compute the following iterative sequence of projections

$$
\begin{aligned}
(X_i^a, Y_i^a) &= P_{\mathcal{Z}_{n_c}}(X_{i-1}, Y_{i-1}) \\
(X_i^b, Y_i^b) &= P_{\Gamma_{\text{convex}}}\left(X_i^a, Y_i^b\right) \\
(X_i^c, Y_i^c) &= P_{\mathcal{Z}_{n_c}}\left(X_i^b, Y_i^b\right) \\
X_i &= X_i^a + \lambda_i^X (X_i^c - X_i^a), \ \lambda_i^X = \frac{\|X_i^a - X_i^b\|_F{}^2}{\langle X_i^a - X_i^c, X_i^a - X_i^b \rangle} \\
Y_i &= Y_i^a + \lambda_i^Y (Y_i^c - Y_i^a), \ \lambda_i^Y = \frac{\|Y_i^a - Y_i^b\|_F{}^2}{\langle Y_i^a - Y_i^c, Y_i^a - Y_i^b \rangle}
\end{aligned}
\tag{4.15}
$$

Step 3 Return to step 2, until the controller order n_c is the desired one, or infeasibility has been detected.

We will call each sequence of iterations in Step 2 an *outer iteration*, as oppose to the inner iterations used to project onto $\mathcal{Z}_{n_c}$.

The APSP algorithm is heuristic, and it is not possible to show that it always converges to the global optimum. Since the algorithm is heuristic conditions for convergence and divergence are a little difficult to devise. We will however give some suggestions in the following section.

4.4 Convergence and divergence

In last section we devised an algorithm that heuristically tries to compress the control order. This was done by restricting the rank of the coupling matrix. However, we only mentioned briefly criteria for convergence, and a criteria for divergence was not mentioned at all. In this section we will consider this more deeply.

4.4.1 Convergence

Suppose we want to test if X, Y can be used for constructing a controller of order $\overline{n_c}$ we go ahead and use the following procedure:

(i) Use theorem 3.2 to compute $\hat{\mathsf{X}} = \mathsf{X} - L$ with

$$
\begin{bmatrix} \hat{\mathsf{X}} & I \\ I & \mathsf{Y} \end{bmatrix} \geq 0
$$

(ii) Compute the eigenvalues $\lambda_i(M_X)$ of

$$
M_X = \tilde{B}_u^{\perp}\left(X\tilde{A}^T + \tilde{A}X + 2\alpha X\right)\tilde{B}_u^{\perp T}
$$

Denote the smallest eigenvalue by $\underline{\lambda}$, and the largest by $\overline{\lambda}$.

(iii) If the smallest of the eigenvalues $\underline{\lambda} > -\varepsilon$, $\varepsilon > 0$ then we assume that we have proven the existence of a controller of desired order. If $\underline{\lambda} < -\varepsilon$ increase $\overline{n}_c$ and goto (i).

(iv) Compute the Lyapunov matrix Y from $\hat{\mathsf{X}}$ and Y, and solve the following SDP problem

$$\begin{array}{ll} \underset{G,l}{\text{minimize}} & l \\ \text{subject to} & F_{\alpha,n_c}GF_{\alpha,n_c} + (F_{\alpha,n_c}GF_{\alpha,n_c})^T + Q_{\alpha,n_c} - lI < 0 \end{array}$$

If $l < \delta$ then the controller is probably fine, otherwise increase $\overline{n}_c$ and goto (i).

The only thing that remains is to choose ε, δ. An appropriate choice for ε is to be determined by numerical experiments. $\varepsilon = 10^{-3}n^2$ has been used in the examples. Note that typically M_X is close to zero, because all constraints in $\mathcal{X}_{\alpha,0}$, $\mathcal{Y}_{\alpha,0}$ and $\mathcal{Z}$ are active. δ is also difficult to choose, has been used in the examples below.

4.4.2 Divergence

As a criteria for divergence we will consider the length of the projection onto Γ_{convex}, that is we consider

$$\phi(i) = \left\|X_i^b - X_i^a\right\| + \left\|Y_i^b - Y_i^a\right\| > 0 \ .$$

It turns out that ϕ_i in average go exponentially to zero, if the algorithms converges. Therefore consider $f(i) = \log\phi(i)$. At iteration N we find the least squares line $ai + b$ fitting the last K values of $f(i)$. If $a > 0$ then the algorithm is diverging, and if $a < 0$ we cannot say anything.

One might infer that a criteria like $\phi(i)$ less than a small number is adequate for determining convergence, but this number will be dependent of the problem data.

4.5 Examples

The above mentioned algorithm, which we will call the *APSP algorithm*[1], has been implemented in MATLAB under the so called Induced Norm Control Toolbox. In appendix a description of the calls for this toolbox is described, see chapter A.

We now consider a very simple example, later we shall solve this problem to global optimum with BMI techniques.

Example 4.1 (A simple α stabilizing control problem) *We consider the α stability of the following system*

$$\begin{bmatrix} A & B_u \\ C_y & 0 \end{bmatrix} = \left[\begin{array}{cc|c} 0 & 1 & 1 \\ 1 & -1 & 0 \\ \hline 1 & 1 & 0 \end{array}\right]. \tag{4.16}$$

with a zero'th order controller. A root locus plot of the above system is given in figure 4.3. The optimal controller has gain -5 and places bot the closed loop poles at -3. A total of 21 problems have been solved using INCT for different decay rates between 1 and 3. The APSP algorithm could find a zeroth order controller up to 2.95. In figure 4.4 a plot of the condition number of the Lyapunov function and the number of necessary iterations are plotted. As it can be seen the condition number grows with the α stability. In fact we cannot find a Lyapunov function that proves that $A_{cl}(-5) = A + B_u(-5)C_y$ has an α stability of -3, since $A_{cl}(-5) + 3I$ is singular. The number of iterations gets also higher, when α get closed to the optimal values.

[1] For historical reasons the algorithm is called the APSP algorithm and not APSDP

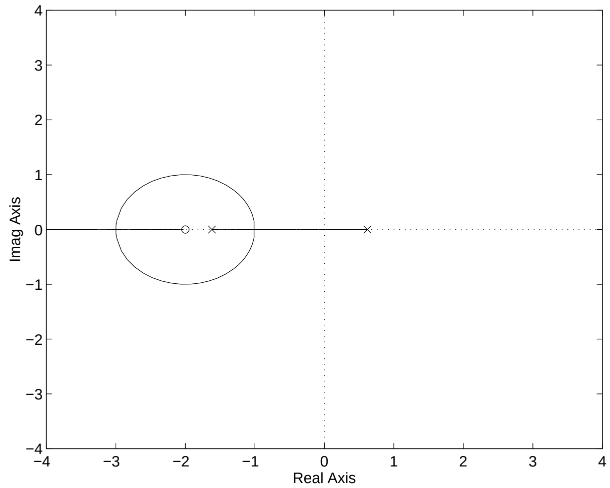


Figure 4.3: Root locus plot of simple control problem

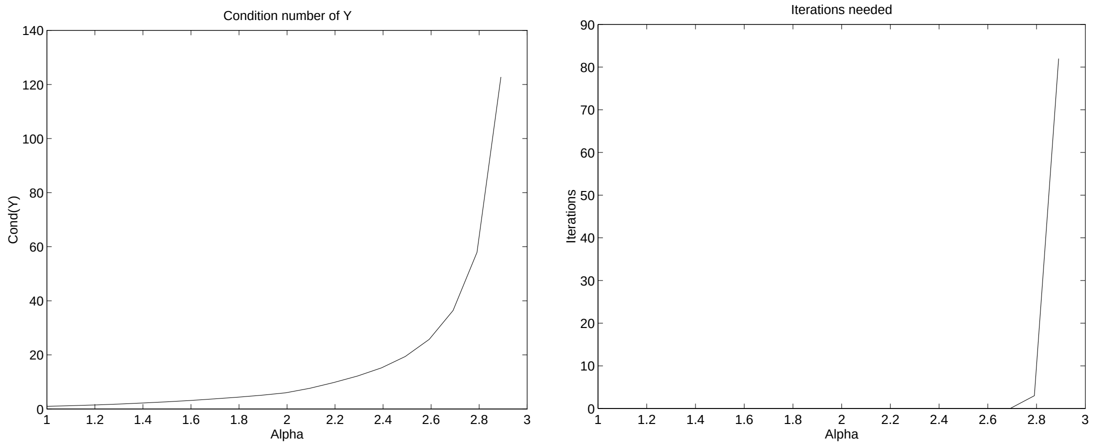


Figure 4.4: Condition number of Lyapunov function and iterations as a function .α

We will consider one example that can show the divergence and convergence criteria.

Example 4.2 (Two wagon example I) *Consider the following state-space representation of a two mass-spring system*

$$\begin{bmatrix} \dot{x} \\ y \end{bmatrix} = \begin{bmatrix} \mathsf{A} & \mathsf{B}_u \\ \mathsf{C}_y & 0 \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$$
$$= \left[\begin{array}{cccc|c} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \end{array}\right] \begin{bmatrix} x \\ u \end{bmatrix} \tag{4.17}$$

We seek to find the lowest order stabilizing controller with the best decay rate α*. First note that the system (4.17) is controllable and observable, which means that we can place the poles freely* *with a full order controller. For a reduced-order controller $n_c < n-1$ we can not expect to be able to place the poles freely.*

We have used the APSP algorithm on the above wagon example for different control orders and decay rates. In table 4.1 the results are presented. We have given number of iterations, and norm of the controller parameter G. First note that 0 iterations means that the we have solve the problem (4.14). For a lot of the cases this is sufficient to obtain the required order for the specific decay rate. For controllers of order $n_c = n-1 = 3$ we can find a controller that obtain the decay rate. Note that the norm of the controller parameter increases with the decay rate.

For controllers of order 2 the maximum achieved with APSP seems to be .42. *However Iwasaki have in [Iwa97b] reported .72 as an achievable decay rate with a controller of order 2. Iwasaki obtained this result by transforming the system to discrete time and then computing the controller in that framework.*

	Effort	α					
n_c	req.	0.01	0.15	0.2	0.3	0.43	1.0
3	Iter.	as	as	0	0	0	0
	$\|\|G\|\|$	$n_c = 2$	$n_c = 2$	4.67	13.6	34.1	490
2	Iter.	0	0	2	4	18	NC
	$\|\|G\|\|$	1.96	2.71	3.01	4.01	6.6	

Table 4.1: Results of the combined Alternating Projection and Semidefinite Programming Algorithm.

Decay rate $\alpha = .43$

A controller of order 3 was found using (4.14). The Lyapunov matrix obtained was

$$Y = \begin{bmatrix} 4.1837 & -0.9429 & -2.1687 & -3.9770 & 5.5845 & -2.0547 & 1.4909 \\ -0.9429 & 15.3083 & 1.6362 & -3.0636 & 4.8844 & 9.0320 & -1.3588 \\ -2.1687 & 1.6362 & 5.0176 & 2.7083 & -5.0084 & 6.0431 & 2.6004 \\ -3.9770 & -3.0636 & 2.7083 & 7.0885 & -8.4411 & 0.2814 & -1.3429 \\ 5.5845 & 4.8844 & -5.0084 & -8.4411 & 12.1681 & -1.9891 & -0.3157 \\ -2.0547 & 9.0320 & 6.0431 & 0.2814 & -1.9891 & 11.0280 & 2.9417 \\ 1.4909 & -1.3588 & 2.6004 & -1.3429 & -0.3157 & 2.9417 & 4.5415 \end{bmatrix}$$

with Frobenius norm 36.02 *and condition number* 636. *The obtained controller was*

$$\begin{bmatrix} \dot{x}_c \\ u \end{bmatrix} = \left[\begin{array}{ccc|c} -3.2543 & -2.4695 & 2.9917 & -8.0230 \\ 2.4697 & 2.6591 & -5.0253 & 8.9036 \\ 2.9914 & 5.0249 & -2.4472 & 9.5606 \\ \hline -8.0231 & -8.9032 & 9.5615 & -24.2401 \end{array}\right] \begin{bmatrix} x_c \\ y \end{bmatrix}$$

with closed loop poles at

$$\lambda_i(A) = \begin{bmatrix} -0.4389 \pm 3.2659j \\ -0.4331 \pm 1.4204j \\ -0.4324 \pm 0.7440j \\ -0.4336 \end{bmatrix}$$

Searching for a controller of order 2, we show the development of $\phi(i)$ *and* $\underline{\lambda}$ *in figure 4.5.*

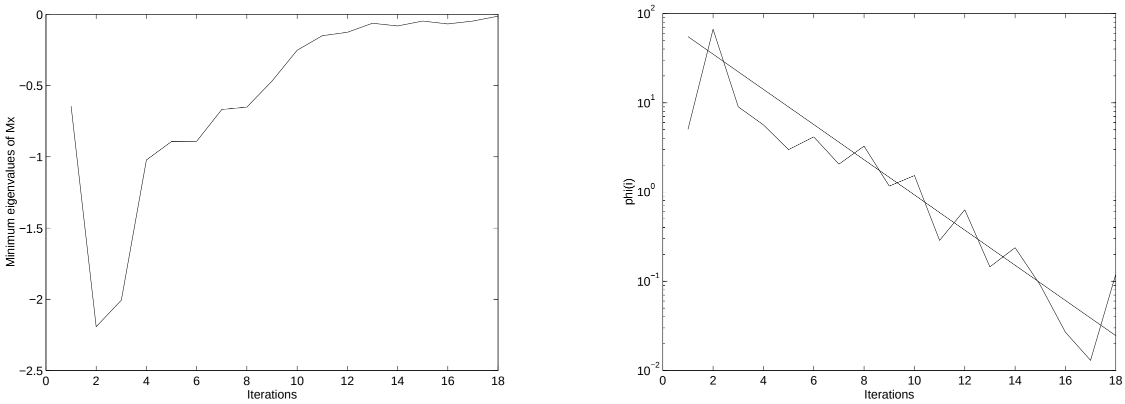


Figure 4.5: Convergence for wagon example.
To the left a plot of the minimum eigenvalue of M_x as a function of the iterations. To the right a logarithmic plot of $\phi(i)$ as a function of the iterations. On top of this a linear approximation of the $log(phi(i))$.

After 18 iterations a controller of order 2 was obtained, that solves the problem. The matrices X, Y *were*

$$\mathsf{X} = \begin{bmatrix} 37.1341 & 22.9670 & -17.8133 & -24.4235 \\ 22.9670 & 32.8450 & 5.9549 & -14.4488 \\ -17.8133 & 5.9549 & 26.4639 & 9.1680 \\ -24.4235 & -14.4488 & 9.1680 & 21.5018 \end{bmatrix}$$

$$\mathsf{Y} = \begin{bmatrix} 21.4823 & 8.9704 & -14.4544 & -24.4131 \\ 8.9704 & 26.5530 & 5.9548 & -17.8372 \\ -14.4544 & 5.9548 & 32.7799 & 22.9354 \\ -24.4131 & -17.8372 & 22.9354 & 37.2378 \end{bmatrix}$$

The closed loop Lyapunov matrix was

$$Y = \begin{bmatrix} 21.4823 & 8.9704 & -14.4544 & -24.4131 & 36.1777 & -0.3680 \\ 8.9704 & 26.5530 & 5.9548 & -17.8372 & 25.5281 & 25.4431 \\ -14.4544 & 5.9548 & 32.7799 & 22.9354 & -29.4073 & 24.2979 \\ -24.4131 & -17.8372 & 22.9354 & 37.2378 & -51.3922 & -1.5242 \\ 36.1777 & 25.5281 & -29.4073 & -51.3922 & 72.4195 & 4.4211 \\ -0.3680 & 25.4431 & 24.2979 & -1.5242 & 4.4211 & 35.4631 \end{bmatrix}$$

with Frobenius norm 164.7 *and condition number* $2.20 \cdot 10^4$*. We note that the Lyapunov function necessary to prove the existence of a controller of second order has a higher norm than the one for a controller of third order. Seen in the light that the controller of order three was found solving 4.14 this is not so strange. However, that the condition number grows indicates that the Lyapunov function, is getting more and more stretch out.*

The found controller was

$$\begin{bmatrix} \dot{x}_c \\ u \end{bmatrix} = \left[\begin{array}{cc|c} 0.0498 & 1.8640 & 1.0353 \\ -1.8624 & -2.9322 & -3.1473 \\ \hline 1.0354 & 3.1533 & 2.4692 \end{array}\right] \begin{bmatrix} x_c \\ y \end{bmatrix}$$

with closed loop matrices placed at

$$\lambda_i(A) = \begin{bmatrix} -0.4795 \pm 1.0583j \\ -0.5060 \pm 0.8490j \\ -0.4556 \pm 0.1554j \end{bmatrix} \tag{4.18}$$

For $\alpha = .5$ *we get the behavior given in figure 4.6. The algorithm determines divergence after 64 iterations (it checks every 8), as it can be seen* M_x *reaches a level at around* -0.42.

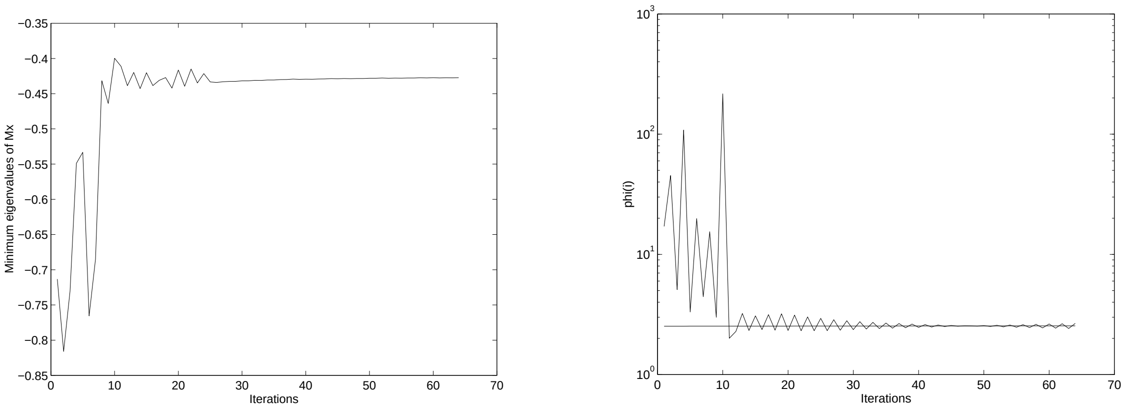


Figure 4.6: Divergence for wagon example.
To the left a plot of the minimum eigenvalue of M_x as a function of the iterations. To the right a logarithmic plot of $\phi(i)$ as a function of the iterations. On top of this a linear approximation of the $log(phi(i))$.

As a final example of the algorithm's behavior on the present example, we show the number of iterations and the maximum M_x *as a function of* α

$\overline{\alpha}$	iter.	$\max M_x$
0.4	18	-0.016
0.45	16	-0.25
0.5	64	-0.43
0.55	24	-0.55
0.6	8	-0.53
0.65	8	-0.49
0.7	8	-0.47
0.8	8	-0.43

Table 4.2: Convergence criteria as a function of α

The optimal decay rate was $.456$*. If the algorithm is run with this decay rate it does not converge, as it can be see in table 4.2. In [Iwa97b] Iwasaki obtained a decay rate of* $.4453$ *using his dual iteration algorithm. Using a discrete time approach he can get a decay rate of* $.72$ *which is much better.*

In appendix we present more experimental results. XXX summarize the results.

Chapter 5

Bilinear Matrix Inequalities

This chapter is devoted to computational methods for solving bilinear matrix inequalities (BMIs). The BMI was introduced by Safonov and his group in 1994 in the conference paper [SGL94] (journal version [GSL95]). It was shown how a μ/K_m synthesis problem could be formulated as a BMI *feasibility* problems. In this context we will use the BMI formulation to design a controller of fixed structure or/and order, see chapter 2 and chapter 3.

First we give a historical overview of previous work on optimization of BMIs. The work can be divided in two main parts - local and global methods.

A couple of local methods have been applied to the BMI problem. In [GTS$^+$94] a number of computational methods for BMI feasibility problems was considered. First it was proposed to solve the BMI as a double LMI feasibility problem, but immediately thereafter it was stated that this approach would not lead to the global optimum, and in fact not even a local optimum. Next a subgradient method was suggested, but it was problematic in implementation. Finally the *method of centers* was employed to solve the BMI feasibility problem. Method of centers had a couple of years earlier been use by Boyd and El Ghaoui to solve generalized eigenvalue problem, see [BE93].

We now proceed by summarizing work in the global optimization of BMIs problems. As pointed out earlier the BMI optimization problem is not convex - in fact the problem is highly non-convex. The feasibility region can be non-convex, and more over the feasibility region can be disconnected. For convex problems a local optimum is also a global optimum. To solve the BMI problem globally we have to examine all local minima. This problem turns out to be very hard in general. The hardness of a problem or the problems complexity has been studied in the literature, see [Vav95]. Convex problems are known to be solvable in polynomial time, that is the required time is bounded above by polynomial in the number of variables. For the global optimization of BMIs Toker and Özbay have showed in [TO95] that the problem is NP-hard. This means that computational time in general grows non-polynomial with the number of parameters. This does not omit the possibility of efficient computational methods for specific subclasses of the BMI problem.

The computational methods studied in the literature can be divided in two subparts - Branch and Bound algorithms, and cone programming techniques.

Branch and Bound (BB) algorithms have been studied extensively in the optimization literature, see [LW66] for an early survey. BB algorithms have been used in the beginning of the 1990's to find the minimum stability degree of parameter dependent systems [BBB91a, BBB91b] Balakrishnan and Boyd has the following explanation, see [BB92]: *Branch and bound algorithms derive their name from the way the proceed: They break up the parameter region into subregions* *("branching") to derive bounds for the global optimum over the original region ("bounding").* The BB techniques used on the minimum stability degree was relative efficient and problems of interesting sizes could be solved.

The earliest work on global optimization for BMIs using BB algorithms appeared in 1994 in the conference paper [GSP94] by Goh, Safonov, and Papavassilopoulos. Goh et al. used the BB algorithm by restricting the variables to be in a hyper rectangle, and then derived lower bounds using relaxation of the bilinear connection and upper bound using method of centers. Their approach was described further in [GSP95], and in deeper details in Goh's Ph.D. thesis, see [Goh95]. A number of control problems have been crunched using the techniques, but the computational time needed to be measured in days, see [KYS96, Van97]. The problem seemed to be that the partition was done in the entire parameter space, and thus not exploiting the biconvexity of the BMI problem.

Recently D.C. (difference of convex sets/function) optimization techniques [Tuy95] have been used by Horsoe, Tuy and Tuan on the BMI feasibility problem, see [THT96]. This technique exploits the biconvexity by partition in one set of the variables instead of both sets. The lower bound is still derived using relaxation of the bilinear connection, but the solution of the relaxed problem is utilized in the computation of the upper bound.

Cone programming techniques have been proposed by Mesbahi and Papavassilopoulos in the paper [MP96]. The approach reformulate the BMI problem as a linear complementarity problem over cones. Unfortunately no numerical implementation has been done yet.

We now proceed by giving the background for the technique we will apply to the BMI problem.

In [Ben62] Benders introduced an approach that exploits the structure of the problem. First Benders noted that many problems becomes much simpler, when fixing some of the variables - called *complicating variables*. The class of problems that Benders considered reduced to a linear programming problem parameterized by the values of the complicating variables. The optimum of the LP problem for each complicating variable renders a new function. Finding the optimum of this function over the set of complicating variables is equivalent to solving the original problem. Benders proposed to solve this problem by building a family of cuts that represent the new function. The cut was found by invoking the dual of the LP problem, and then the relaxed version provided a tangent if the problem was feasible for the fixed variable, otherwise a half space could be cut away. Finding the minimum over the family of cuts is also a linear program, we refer to it as a subproblem. The solution of the subproblem provides approximate solution to the original problem. The subproblem can be refined by adding more cuts, until the solution has a required accuracy.

Geoffrion extended the work of Bender to a more general case [Geo72], where the subproblem did not need to be a linear program. Geoffrion especially considered the following problem:

$$\max_{x,y} f(x,y) \text{ subject to } G(x,y) \geq 0, \;\; x \in \mathcal{X}, \;\; y \in \mathcal{Y}, \tag{5.1}$$

where y is the vector of *complicating variables*. G is an m-vector of constraint functions, defined on $\mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^{m_x} \times \mathbb{R}^{m_x}$. G was required to have special structure, and we will especially consider the situation, where $G(x,y)$ is rendered convex, when y is fixed. Geoffrion then solved the problem in the same fashion as Benders.

However, as pointed out by Sahinidis and Grossman in [SG91] this approach do not guarantee global convergence. It was pointed out that just using the tangent to a function that is non convex does not provide a valid cut. It can be shown that the obtained solution is not necessarily a local optimum.

This problem can be alleviated by deriving valid piecewise lower bounds. This solution was suggested by Floudas and Visweswaran in [FV93]. They considered problems equivalent to (5.1). However, they considered $\mathcal{X}$ to be rectangular and exploited this to partition $\mathcal{Y}$ in 2^{m_x} regions over which linear lower bounds could be derived. They showed that a series of problems of medium size could be solved in a fair amount of time. The strength of the method was the ability to exploit the bilinear structure. The work of Floudas and Visweswaran have since then been studied and described to great detail in a series of papers following [FV93], see [VF96a] and reference therein. The algorithm is refered to as the GOP algorithm. Software has been developed especially the so called cGOP package is of interest, see [VF96b].

In control the approach by Floudas and Visweswaran has been exploited to solve specific control problems. Psarris and Floudas suggested to solve robust stability analysis problems under parametric uncertainties by using the GOP algorithm [PF95]. Similar work was done in [BFHT95].

In this chapter we will devise a family of branch and bound algorithms for the BMI optimization problem. We first state the exact problem we want to solve. Next we decompose the problem as in the Benders/Geoffrion approach. A simple branch and bound algorithm is described. Different approaches for obtaining upper and lower bounds are presented. The different lower bounds poses various properties, where the BB algorithm has to be tailored in different ways. We then present three customized algorithms for each lower bound, and discuss convergence issues related to the three algorithms. An examination on how to formulate BMI formulation arising in control the best way is done thereafter. We proceed with a series of small examples to compare the different methods.

5.1 Problem statement

We will consider a formulation of the BMI optimization problem, where the convex and non-convex constraints have been separated. As our objective we only consider a linear function over the parameter space. The problem looks as follows:

$$\begin{array}{ll} \text{minimize} & f(x,y) = c^T x + d^T y \\ \text{subject to} & F(x,y) \stackrel{\Delta}{=} F_0 + \sum_{i=1}^{m_x} x_i F_i^x + \sum_{j=1}^{m_y} y_j F_j^y + \sum_{j=1}^{m_y} \sum_{i=1}^{m_x} y_j x_i F_{ij}^{xy} \geq 0 \\ & G(x,y) \stackrel{\Delta}{=} G_0 + \sum_{i=1}^{m_x} x_i G_i^x + \sum_{j=1}^{m_y} y_j G_j^y \geq 0. \end{array} \tag{5.2}$$

The variables are $x \in \mathbb{R}^{m_x}$ and $y \in \mathbb{R}^{m_y}$. The symmetric matrices $F_\bullet^\bullet$ and $G_\bullet^\bullet$ are given data. We call this problem for the *BMI optimization problem*, or just the *main problem*. If there exist x and y such that $F(x,y) \geq 0$ and $G(x,y) \geq 0$ we say that the problem is *feasible*.

We want to solve this problem to ε optimality, but before we look at a precise statement, we want to restrict our attention to a smaller class of problems. We will impose two assumption. The first relates to the feasibility and size of $G(x,y) \geq 0$:

Assumption 5.1 *The set* $\{(x,y) \mid G(x,y) \geq 0\}$ *is bounded and non-empty.*

Since our objective function is linear assumption 5.1 implies that if equation (5.2) is feasible then the optimal solution is finite. The second assumption is more restrictive, but for the algorithms we will consider later it is very essential:

Assumption 5.2 $\{y : \exists x \text{ such that } G(x,y) \geq 0\} \subset \{y : \exists x \text{ such that } F(x,y) \geq 0\}$.

This assumption might seem very restrictive, but we can still formulate a problem that solves the main problem to a given approximation. Introduce a slack variable t, with bounds $0 \leq t \leq M$, add t times identity to the BMI and ρ times t to the objective,

$$\begin{array}{ll} \text{minimize} & f(x,y) = c^T x + d^T y + \rho t \\ \text{subject to} & F(x,y) + tI \geq 0 \\ & G(x,y) \geq 0. \end{array} \tag{5.3}$$

The biggest problem is to choose ρ. It should be big enough so it does not influence the optimum significantly and small enough not to destroy the scaling of the system.

The two assumptions, 5.1 and assumption 5.2 are the only ones we will impose on our problem. These assumptions are sufficient to prove that the optimum is finite and exists. The non-emptiness in 5.1 together with assumption 5.2 implies the existence, where as the boundedness in assumption 5.1 implies that the optimum is finite. We will denote this optimum by f^*. We now give a formal definition of our goal:

Problem 5.1 (ε-optimal BMI problem) *Given an $\varepsilon > 0$ and a problem on the form (5.2) under assumption 5.1 and assumption 5.2. Let f^* be the optimum of (5.2) then find a pair (x^*, y^*), s.t. $F(x^*, y^*) \geq 0$, $G(x^*, y^*) \geq 0$ and*

$$f(x^*, y^*) - f^* \leq \varepsilon \ . \tag{5.4}$$

We will call such a solution (x^*, y^*) for ε-optimal.

The above problem 5.1 is a global optimization problem. Since the problem is non-convex in general a local optimum is not necessarily global.

For ease of notation and discussion we will define a couple of sets which relates to the BMI and LMI constraint in equation (5.2). First three sets that relate to $F(x,y) \geq 0$:

$$\begin{aligned} \mathcal{F} &\triangleq \{(x,y) : F(x,y) \geq 0\} \subset \mathbb{R}^{m_x \times m_y} \\ \mathcal{X}_F &\triangleq \{x : \exists y \text{ such that } F(x,y) \geq 0\} \subset \mathbb{R}^{m_x} \\ \mathcal{Y}_F &\triangleq \{y : \exists x \text{ such that } F(x,y) \geq 0\} \subset \mathbb{R}^{m_y} \ . \end{aligned} \tag{5.5}$$

Note that $\mathcal{F}$ in general is non-convex, however its projection $\mathcal{X}_F$ onto x-space can be convex, and also the projection $\mathcal{Y}_F$ onto y-space might be convex. Just consider the BMI constraint $0 \leq xy \leq 1$, $x, y \in \mathbb{R}$, then $\mathcal{F}$ is not convex, where as $\mathcal{Y}_F = \mathbb{R}$ and $\mathcal{X}_F = \mathbb{R}$ are both convex.

The next three sets relate to $G(x,y) \geq 0$:

$$\begin{aligned} \mathcal{G} &\triangleq \{(x,y) : G(x,y) \geq 0\} \subset \mathbb{R}^{m_x \times m_y} \\ \mathcal{X}_G &\triangleq \{x : \exists y \text{ such that } G(x,y) \geq 0\} \subset \mathbb{R}^{m_x} \\ \mathcal{Y}_G &\triangleq \{y : \exists x \text{ such that } G(x,y) \geq 0\} \subset \mathbb{R}^{m_y} \end{aligned} \tag{5.6}$$

Since $\mathcal{G}$ is described by an LMI it is convex, and both projections $\mathcal{X}_G$ onto x-space and $\mathcal{Y}_G$ onto y-space are convex.

Assumption 5.2 can now be written as $\mathcal{Y}_G$ should be a subset of $\mathcal{Y}_F$.

With the definition of these sets, we note the simple description of the general BMI optimization problem as

$$\begin{array}{ll} \underset{x,y}{\text{minimize}} & f(x,y) \\ \text{subject to} & (x,y) \in \mathcal{F} \cap \mathcal{G} \ . \end{array} \tag{5.7}$$

In other words we want to minimize a linear functional over a non-convex region. The complication is the bilinear connection between the x and y variables in the constraint $F(x,y) \geq 0$. We will try to conquer this by projecting the problem onto y-space, in which case the problem is to minimize a non-convex function over a convex region. This approach is called Benders decomposition.

5.2 Generalized Benders decomposition

The main idea in Bender's decomposition is the concept of *projection*. The projection of (5.7) onto y is the following

$$\underset{y}{\text{minimize}}\; v(y) \text{ subject to } y \in \mathcal{Y}_F \cap \mathcal{Y}_G \tag{5.8}$$

where

$$v(y) \overset{\Delta}{=} \inf_x [\, f(x,y) \text{ subject to } F(x,y) \geq 0,\; G(x,y) \geq 0] \quad . \tag{5.9}$$

We will call this problem (5.8) for the *master problem*.. The function $v(y)$ is refered to as the *optimal-value function*. The value of $v(y)$ for any fixed y can be computed as an SDP problem, which is a tractable convex problem, and for this reason y is refered to as the *complicating* variable. .

The sets $\mathcal{Y}_F$ and $\mathcal{Y}_G$ are the projections of the sets $\mathcal{F}$ and $\mathcal{G}$ onto y respectively. The union of $\mathcal{Y}_F \cap \mathcal{Y}_G$ is by assumption 5.2 equal to $\mathcal{Y}_G$. By designating y as the complicating variables evaluating $v(y)$ at point y is much easier than solving the BMI optimization problem in itself. In fact evaluating $v(y)$ in a fixed point $\hat{y}$ is an SDP problem. In [Geo71] Geoffrion showed that if y^* solves (5.8) and x^* solves (5.9) for $y = y^*$ then (x^*, y^*) solves (5.2).

By reformulating the problem in terms of minimizing a non-convex function $v(y)$ over a region $\mathcal{Y}_G$ we have formulated the problem as an infinite set of SDP problems parameterized by the value of the complicating variable y. The difficulty with the master problem is that $v(y)$ is only given implicit in terms of (5.9). Geoffrion resolved this problem by invoking the dual representation of v as a point-wise supremum of a collection of functions which underestimate it. In this case it is very important that the functions are in fact global underestimaters. Geoffrion only used a tangent approximation (called Benders cut) obtained via the Lagrangian dual, but for non-convex problems the tangent might not even be a local underestimater. In a paper by Sahinidis and Grossman[SG91] this issue is treated, and it is shown that just using the tangent will lead to a solution that is not the global optimum. To assure that we always find a global solution we will only consider global underestimaters. The master problem can then be solved by evaluating the point-wise supremum of these underestimaters. The supremum of the underestimaters might only be convex in certain regions. However, by calculating the underestimater, directions for a set of partitioning rules are usually obtained. These partitioning rules can be used to split the original region $\mathcal{Y}_G$ into smaller region over which the underestimater is convex. If we can partition $\mathcal{Y}_G$ in smaller and smaller regions over which we obtain better and better upper and lower bounds, we can solve the ε-optimal BMI problem, in finite time. To do this we need two basic concepts, first computable upper/lower bounds and partitioning rules, and second an algorithm that can exploit this partition and bounding to obtain the major goal. The algorithm we will consider is called branch and bound, and is described in the next section.

5.3 Branch and Bound Algorithm

We consider the following problem

$$\min_y v(y) \text{ subject to } y \in \mathcal{Y} \tag{5.10}$$

where v is continuous and bounded, and $\mathcal{Y}$ is convex, non-empty, closed and bounded. In this section we drop the subscript on $\mathcal{Y}_G$. The requirement on v and $\mathcal{Y}$ guarantees that the global optimum exists. We denote the optimum by v^*. Let ε be a given accuracy then we want to find y^*, (and a certificate) such that

$$v(y^*) - v^* \leq \varepsilon \tag{5.11}$$

We say that we solve the problem to ε-optimality. The goal in (5.11) implies that we during our search for y^* build a set of upper and lower bounds, that guarantees ε-optimality.

To solve this problem we consider a branch and bound scheme. The branch and bound scheme relies on the existence of computable upper and lower bounds for v. In fact we assume the existence of a *branch and bound-operator* (BB-operator). Given a region $\mathcal{Q}$ this operator should supply us with the following

i) An upper bound $\overline{f}_{\mathcal{Q}}$ for v over $\mathcal{Q}$ fulfilling

$$\overline{f}_{\mathcal{Q}} \geq \min_{y \in \mathcal{Q}} v(y)$$

and a $y \in \mathcal{Q}$, that achieves $\overline{f}_{\mathcal{Q}} = v(y)$.

ii) A partition of $\mathcal{Q}$ in $p_{\mathcal{Q}}$ regions, $\mathcal{Q}_i$, $i = 1, \cdots, p_{\mathcal{Q}}$, obeying

$$\bigcup_{i=1}^{p_{\mathcal{Q}}} \mathcal{Q}_i = \mathcal{Q}$$
$$\operatorname{Int} \mathcal{Q}_i \bigcap \operatorname{Int} \mathcal{Q}_j = \emptyset, \; i \neq j$$

and lower bounds $\underline{f}_{\mathcal{Q}_i}$ over each new region $\mathcal{Q}_i$ with

$$\underline{f}_{\mathcal{Q}_i} \leq \min_{y \in \mathcal{Q}_i} v(y), \; i = 1, \cdots, p_{\mathcal{Q}}$$

Restriction are needed on the branch and bound operator, if we want to prove convergence of the propose algorithm.

As the main tool to solve the ε-optimality problem, we consider a branch and bound tree. We will use terms from the graph theory to describe the tree. Each node in the tree corresponds to a region, and associated with this node is valid upper and lower bounds for the region. The branches of each node corresponds to a partition of the associated region. The root of the tree is the original region $\mathcal{Y}$. As the tree grows the leaves of the trees represent a partition of $\mathcal{Y}$. The minimum lower bound over the leaves provides a global lower bound on the main problem. The minimum upper bound over all parents gives a global upper bound.

The branch and bound algorithm gives direction how to grow the tree to solve the ε problem. The tree is grown in the following way. At each iteration k a region is picked (selection rule) and the BB operator is applied. The selection rule is to picked the leaf with the lowest lower bound. This provides a more fine partition of $\mathcal{Y}$ and a better bounding of the problem.

Branch and bound algorithm

1. **Initialization:** Set $k = 1$, and let $\mathcal{Q}^{(1)} \triangleq \mathcal{Y}$.

2. **Branching and local bounding:** Apply the BB operator to $\mathcal{Q}^{(k)}$. Obtain

   (a) Upper bound $\overline{f}^{(k)}$, and $y^{(k)}$ such that $\overline{f}^{(k)} = v(y^{(k)})$.

   (b) Partition $\mathcal{Q}_i^{(k)}, i = 1, \cdots, p^{(k)}$ and lower bounds $\underline{f}_i^{(k)}$.

   (c) Insert $\mathcal{Q}_i^{(k)}$, $i = 1, \cdots, p^{(k)}$ in the BB tree as leaves of $\mathcal{Q}^{(k)}$.

3. **Global bounding:**

   (a) Compute best upper bound $\overline{f}^{(k)} = \min_{j=1,\cdots,k} u^{(j)}$ and let y^u be the y achieving it, $\overline{f}^{(k)} = v(y^u)$.

   (b) Compute the best lower bound $\underline{f}$, as

$$\underline{f}^{(k)} = \min_{l=1,\cdots,k} \min_{\substack{i=1,\cdots,p^{(l)} \\ \mathcal{Q}_i^{(l)} \text{leaf}}} \underline{f}_i^{(l)}$$

4. **Conditioning:** If $\overline{f}^{(k)} - \underline{f}^{(k)} \leq \varepsilon$ then set $y^* = y^u$ and EXIT.

5. **Selection:** Update $k \triangleq k+1$, and let $\mathcal{Q}^{(k)} \triangleq \mathcal{Q}_i^{(j)}$ be one of the regions where $\overline{f} = \overline{f}_i^{(j)}$.

6. Goto 2.

Conditions for convergence of the above algorithm can be stated. First let $\mathcal{R}$ be a rectangle defined by l, u such that $\mathcal{R} = \{y : \ l \leq y \leq u\}$, where the diameter of $\mathcal{R}$ is $||u - l||_2$. The following conditions from [HJ95] XXX are sufficient for proof of convergence

H1 As the diameter of the minimal rectangle $\mathcal{R} \supset \mathcal{Q} \to 0$ then $\overline{f}_{\mathcal{Q}} - \min_{i=1,\cdots,p_{\mathcal{Q}}} \underline{f}_{\mathcal{Q}_i} \to 0$.

H2 The diameter of the minimal rectangle $\mathcal{R} \supset \mathcal{Q}$ must go to zero when the number of parents go to infinity.

H3 The subproblem with the lowest lower bound must be selected in at least every K'th iteration.

Condition H3 is fulfilled by the "selection"-rule above, where as H1 and H2 depend on the bounding and partition from the BB operator.

The above description of the branch and bound algorithm is very schematic, and in a practical implementation the algorithm will have to be customized to exploit the bounding and partition rules.

5.4 Upper bound

In this section we will show how to compute an upper bound for the master problem over the region $\mathcal{Q}_F$. Fix $y = \hat{y} \in \mathcal{Y}_G$ then $v(\hat{y})$ can be computed as an SDP problem in x:

$$\begin{array}{ll} \text{minimize} & c^T x + d^T \hat{y} \\ \text{subject to} & F(x,\hat{y}) = \left(F_0 + \sum_{j=1}^{m_y} \hat{y}_j F_j^y\right) + \sum_{i=1}^{m_x} x_i \left(F_i^x + \sum_{j=1}^{m_y} \hat{y}_j F_{ij}^{xy}\right) \geq 0 \\ & G(x,\hat{y}) = \left(G_0 + \sum_{j=1}^{m_y} \hat{y}_j G_j^y\right) + \sum_{i=1}^{m_x} x_i G_i^x \geq 0 \ . \end{array} \tag{5.12}$$

Since equation (5.12) is a restriction of $\mathcal{Y}_G$ then

$$v(\hat{y}) \geq \min_{y \in \mathcal{Y}_G} v(y).$$

The naturally question how do we choose the y we want to fix. Usually a y is available from the solution to the lower bound over the region.

Another possibility to obtain or even improve the upper bound is to use method of centers, see for instance [GSP95].

5.5 Lower bound via relaxation of $x_i y_j$

In this section we provide point-wise lower bounds on $v(y)$ over a region $\mathcal{Q}$. We assume in the next two sections that $\mathcal{Q}$ have been absorbed in $G(x,y) \geq 0$. Apart from giving lower bounds we also give directions for partitioning the region $\mathcal{Q}$ such that the lower bound can be refined. More over we provide a point where it is good to compute the upper bound. We call such a point a center point. The relaxation presented here has been examined in [GSP94, THT96], but for the problem of the form

$$\begin{array}{ll} \underset{x,y}{\text{minimize}} & \lambda_{\max}(F(x,y)) \\ \text{subject to} & l^x \leq x \leq u^x \\ & l^y \leq y \leq u^y \end{array} \tag{5.13}$$

which is an instant of the *BMI feasibility problem* of finding (x,y) such that $F(x,y) \leq 0$.

In [BV97b] a more general problem is studied, that includes quadratic connections in the matrix inequality, and connections between all variables. That is the matrix inequality has the form

$$D(x) = D_{00} + \sum_{i=1}^{m_x} x_i D_{i0} + \sum_{j=1}^{m_x} \sum_{i=1}^{m_x} x_i D_{ij} x_j \ , \tag{5.14}$$

where $D_\bullet \in \mathcal{S}^n$. The matrix inequality (5.14) is not biconvex, even if we fix some of the variables the rest of the problem will still render non convex in general.

5.5.1 The Method of Tuan

The complicating part in the BMI $F(x,y) \geq 0$ is is the bilinear connection between $x_i y_j$. We can find a lower bound for $v(y)$ over a specific region, by region relaxing the bilinear connection $x_i y_j$

by introducing a new variable w_{ij}. The new parameter w_{ij} emulates the bilinear term x_iy_j. The BMI constraint is now relaxed to an LMI in x, y and W ($\{W\}_{ij} = w_{ij}$):

$$\tilde{F}(x,y,W) = F_0 + \sum_{i=1}^{m_x} x_i F_i^x + \sum_{j=1}^{m_y} y_j F_j^y + \sum_{j=1}^{m_y}\sum_{i=1}^{m_x} w_{ij} F_{ij}^{xy} \geq 0 \quad .$$

From the convex bounds on x and y from $G(x,y) \geq 0$ it is possible to derive bounds on w_{ij}. First relax $\mathcal{X}_G$ and $\mathcal{Y}_G$ to be rectangular. By a rectangle we mean a set on the form $\mathcal{R}_\infty = \{x : x \in \mathbb{R}^{m_x},\ l \leq x \leq u\}$[1], where l, u are upper and lower bounds on each variable in x. We are interested in the rectangle with smallest volume surrounding $\mathcal{X}_G$. Such a rectangle can be found by solving $2m_x$ SDP problems. To find the best lower bound for variable x_i solve the following problem: minimize x_i subject to $x \in \mathcal{X}_G$. Denote the two best rectangles relaxing $\mathcal{X}_G$ and $\mathcal{Y}_G$ by $\mathcal{R}_x$ and $\mathcal{R}_y$ respectively, and the lower and upper bound by l^x, u^x, l^x, u^x respectively. On the variables x_i and y_j we have the bounds

$$\begin{aligned} x_i - l_i^x &\geq 0 \qquad & y_j - l_j^y &\geq 0 \\ u_i^x - x_i &\geq 0 \qquad & u_j^y - y_j &\geq 0 \end{aligned}$$

Multiplying the two bounds on x_i with the two bounds on y_j we get four bounds on x_iy_j:

$$\begin{aligned} (x_i - l_i^x)\left(y_j - l_j^y\right) &\geq 0 \\ (x_i - l_i^x)\left(u_j^y - y_j\right) &\geq 0 \\ (u_i^x - x_i)\left(y_j - l_j^y\right) &\geq 0 \\ (u_i^x - x_i)\left(u_j^y - y_j\right) &\geq 0 \end{aligned} \tag{5.15}$$

By replacing x_iy_j by w_{ij} we get bounds on w_{ij}. We now present a lemma from [THT96]:

Lemma 5.1 *Assume $l_i^x \leq u_i^x$ and $l_j^x \leq u_j^y$ then $l_i^x \leq x_i \leq u_i^x, l_j^y \leq y_j \leq u_j^y$ if and only if there exists w_{ij} such that*

$$\begin{aligned} w_{ij} &\geq l_j^y x_i + l_i^x y_j - l_i l_j^y \\ w_{ij} &\leq u_j^y x_i + l_i^x y_j - l_i^x u_j^y \\ w_{ij} &\leq l_j^y x_i + u_i^x y_j - u_i^x l_j^y \\ w_{ij} &\geq u_j^y x_i + u_i^x y_j - u_i^x u_j^y \quad . \end{aligned}$$

For convenience we will define the set of $W_{\mathcal{R}_x,\mathcal{R}_y}(x,y)$ fulfilling (5.15) by

$$\mathcal{W}_{\mathcal{R}_x,\mathcal{R}_y}(x,y) \triangleq \left\{W : W \in \mathbb{R}^{m_x \times m_y}, \left.\begin{aligned} w_{ij} &\geq l_j^y x_i + l_i^x y_j - l_i l_j^y \\ w_{ij} &\leq u_j^y x_i + l_i^x y_j - l_i^x u_j^y \\ w_{ij} &\leq l_j^y x_i + u_i^x y_j - u_i^x l_j^y \\ w_{ij} &\geq u_j^y x_i + u_i^x y_j - u_i^x u_j^y \end{aligned}\right\}, i = 1, \cdots, m_x, j = 1, \cdots, m_y\right\}$$

The set $W_{\mathcal{R}_x,\mathcal{R}_y}(x,y)$ defines a polytope in W-space. Given the rectangular sets $\mathcal{R}_x$, and $\mathcal{R}_y$ then for each x, y the variable w_{ij} in W is contained in an interval, i.e.

$$w_{ij} \in [\max\left\{l_j^y x_i + l_i^x y_j - l_i l_j^y, u_j^y x_i + u_i^x y_j - u_i^x u_j^y\right\}, \min\left\{u_j^y x_i + l_i^x y_j - l_i^x u_j^y, l_j^y x_i + u_i^x y_j - u_i^x l_j^y]\right\} \tag{5.16}$$

[1]We will later consider different types of hyper rectangles $\mathcal{R}_p$ dependent of the vector p- norm

by $W \in \mathcal{W}_{\mathcal{R}_x,\mathcal{R}_y}(x,y)$. Vice versa, by lemma 5.1 if there exists $W \in W_{\mathcal{R}_x,\mathcal{R}_y}(x,y)$ then $x,y \in \mathcal{R}_x \times \mathcal{R}_y$. Let us study the behavior of $x_i y_j$ and the relaxation w_{ij} by a simple example.

Example 5.1 (Bounds on w from x and y) *Consider bounds on x and y as*

$$
\begin{aligned}
0 &\le x \le 6\\
0 &\le y \le 4
\end{aligned}
$$

The product of x and y in the given regions can be seen in figure 5.1. If w is the relaxation of xy

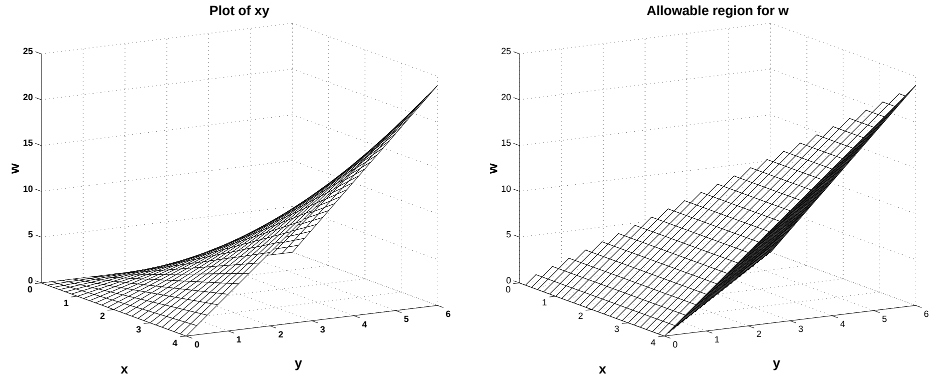


Figure 5.1: Bilinear connection and its relaxation
To the left a plot of the bilinear term xy and to the right the allowable region of w if it is an relaxation of xy.

then we have the following bounds on w:

$$
\begin{aligned}
w &\ge 0x + 0y - 0 = 0\\
w &\le 4x + 0y - 0 = 4x\\
w &\le 0x + 6y - 0 = 6y\\
w &\ge 4x + 6y - 24.
\end{aligned}
$$

which can be seen in figure 5.1. The interval (5.16) is plotted for each x,y in figure 5.2. It should be evident that the interval (5.16) goes to zero, when x,y approaches the boundary of $\mathcal{R}_x,\mathcal{R}_y$.

The example showed, especially figure 5.2, that the difference between the relaxation w and xy becomes zero at the boundary of $\mathcal{R}_x$ and $\mathcal{R}_y$.

We can now find a lower bound for the BMI optimization problem over $\mathcal{Y}_G$ by solving the following SDP problem

$$
\begin{array}{ll}
\underset{x,y,W}{\text{minimize}} & f(x,y)\\
\text{subject to} & \tilde{F}(x,y,W) \ge 0\\
 & G(x,y) \ge 0\\
 & W \in \mathcal{W}_{\mathcal{R}_x,\mathcal{R}_y}(x,y)
\end{array}
\tag{5.17}
$$

Let the optimal variables of the above problem (5.17) be denoted as $\tilde{x},\tilde{y},\tilde{W}$. The above problem (5.17) is a convexification of original non-convex problem (5.2), by making the non-convex constraint convex. This renders the problem (5.17) as a optimization problem with a linear constraint

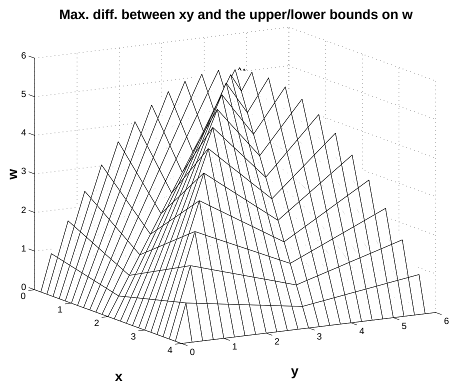


Figure 5.2: Absolute difference between xy and its convexification.
The above is a plot of the maximum absolute difference between xy and (5.16) for each x, y.

over a convex region. The optimum over (x,W) of (5.17) provides a convex function χ

$$\chi(y) = \min_{x,W} \left[f(x,y) \text{ subject to } \tilde{F}(x,y,W) \geq 0, G(x,y) \geq 0, W \in \mathcal{W}_{\mathcal{R}_x,\mathcal{R}_y} \right], \text{ for } y \in \mathcal{Y}_G \tag{5.18}$$

that under estimates $v(y)$. Basically $\chi(y)$ is the projection of (5.17) onto y, and since (5.17) is convex then so is $\chi(y)$. We will call χ a lower bounding function.

We now examine the lower bound by a simple example.

Example 5.2 (Sahinidis and Grossman I) *The following example is from [SG91]:*

$$\begin{array}{ll} \underset{x,y}{\textit{Minimize}} & -x-y \\ \textit{subject to} & xy \leq 4 \\ & 0 \leq x \leq 6 \\ & 0 \leq y \leq 4 \end{array} \tag{5.19}$$

There are two local minima at points $(x,y) = (1,4)$ and $(x,y) = (6,2/3)$ with optima -5 and $-20/3 \approx -6.67$. Sahinidis and Grossman provides the optimal-value function as

$$v(y) = \begin{cases} -6-y & \textit{if } 0 \leq y \leq 2/3 \\ -4/y-y & \textit{if } 2/3 \leq y \leq 4 \end{cases} \tag{5.20}$$

The optimal-value function can be seen in figure 5.3

The relaxation (5.17) becomes the following LP problem:

$$\begin{array}{ll} \underset{x,y}{\textit{minimize}} & -x-y \\ \textit{subject to} & w \leq 4 \\ & 0 \leq x \leq 6 \\ & 0 \leq y \leq 4 \\ & w \geq 0x+0y-0 = 0 \\ & w \leq 4x+0y-0 = 4x \\ & w \leq 0x+6y-0 = 6y \\ & w \geq 4x+6y-24 \end{array} \tag{5.21}$$

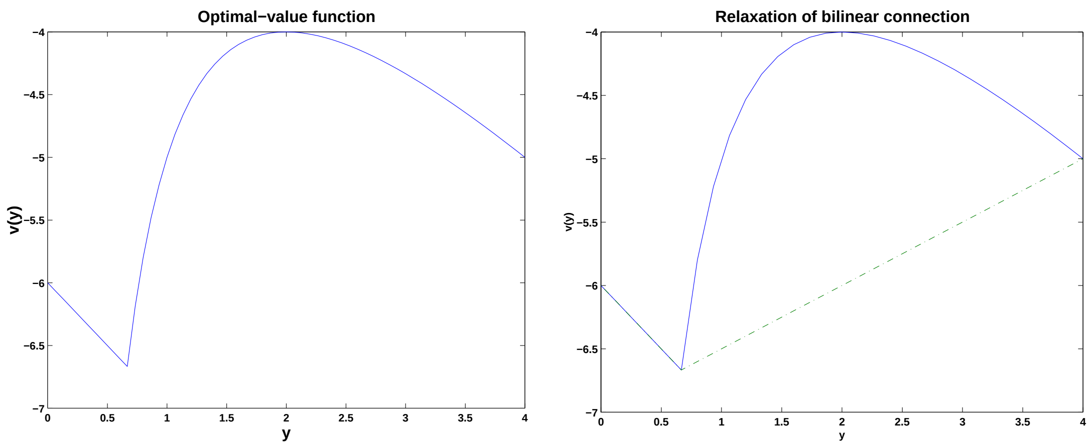


Figure 5.3: Optimal-value function and relaxation for the Sahinidis-Grossman problem
To the left the optimal-value function for the Sahinidis-Grossman problem. To the right the solution of the relaxation of (5.21) as a function of y, that is the function $\chi(y)$.

The optimum is $-20/3$ *just as the global optimum of the problem. The optimum is achieved at* $(x,y,w) = (6,2/3,4)$. *In 5.3 we have plotted (5.21) as a function of* y. *By looking at the figure it is possible to derive the following from (5.21):*

$$\chi(y) = \begin{cases} -6-y & \textit{if } 0 \le y \le 2/3 \\ -7+\frac{1}{2}y & \textit{if } 2/3 \le y \le 4 \end{cases} \tag{5.22}$$

with $v(y) \ge \chi(y), 0 \le y \le 4$. *The optimum of* $\min \chi(y)$ *subject to* $0 \le y \le 4$ *is* $-20/3$. *In other words, by solving the above relaxation we actually solve the original problem. We stress that a generalization to other problems of this fact cannot be made. In figure 5.3 the solution of the relaxation (5.21) is plotted together with* $v(y)$. *The relaxation is actually the convexification of* $v(y)$.

Partition

We will now examine how we should branch, that is divide the current rectangle $\mathcal{R}_y$ in smaller pieces. A simple approach would be to partition $\mathcal{R}_y$ in half along the longest edge. Another approach is to base the partition $\mathcal{R}_y$ on the bilinear connection between x and y related to the relaxation W. If $\tilde{w}_{ij} - \tilde{x}_i\tilde{y}_j = 0$ for all i, j then $\tilde{x}, \tilde{y}$ is a solution to the BMI, and the lower bound is equal to the upper bound. Thus, to ensure that we achieve this goal we want to minimize the maximum difference $|\hat{w}_{ij} - \hat{x}_i\hat{y}_j|$. We do this by finding indices such that

$$(\tilde{i}, \tilde{j}) \in \arg\max_{i,j} \left\{ |\tilde{w}_{ij} - \tilde{x}_i\tilde{y}_j| : i = 1,\cdots,m_x, j = 1,\cdots,m_y \right\} \tag{5.23}$$

then split the rectangle $\mathcal{R}_y$ with the plane

$$\hat{y}_{\tilde{j}} = \begin{cases} \frac{1}{2}\left(\tilde{y}_{\tilde{j}} + \frac{\tilde{w}_{\tilde{i}\tilde{j}}}{\tilde{x}_{\tilde{i}}}\right), & \text{if } \tilde{x}_{\tilde{i}} \ne 0 \\ \tilde{y}_{\tilde{j}} & \text{if } \tilde{x}_{\tilde{i}} = 0 \end{cases} \tag{5.24}$$

We note that $\hat{y}_{\tilde{j}}$ will usually lye in the middle of $\mathcal{R}_y$, because the constraints on w become tight when y is close to the boundary of $\mathcal{R}_y$, and thus making $|\tilde{w}_{ij} - \tilde{x}_i\tilde{y}_j|$ small.

5.5.2 The method of Goh

The first method used to solve the BMI feasibility problem by Goh et al [GSP95, Goh95] used the same relaxation technique as explained in the previous subsection, but was different at a few points.

Instead of restricting W to the set $\mathcal{W}_{\mathcal{R}_x,\mathcal{R}_y}(x,y)$ Goh restricted W to the rectangular relaxation of $\mathcal{W}_{\mathcal{R}_x,\mathcal{R}_y}(x,y)$. Each w_{ij} was restricted to the set $\{w_{ij} : w_{ij} \in [\underline{w}_{ij}, \overline{w}_{ij}]\}$, where

$$\begin{aligned}\underline{w}_{ij} &= \min\{l^x l^y, l^x u^y, u^x l^y, u^x u^y\}\\ \overline{w}_{ij} &= \max\{l^x l^y, l^x u^y, u^x l^y, u^x u^y\} \quad .\end{aligned}$$

This gives only half the constraints on w_{ij} as of (5.16), but the constraints are weaker.

Partition

The partition rule is to split along the longest edge of $\mathcal{R}_x \times \mathcal{R}_y$, instead of just splitting in $\mathcal{R}_y$ by the hyper-plane (5.24).

5.5.3 Alternative method of Tuan

The following lower bound can be found in [THT96]. For the BMI feasibility problem (5.13) the relaxation (5.17) gets the following form: Minimize the function

$$\varphi(x,y,W) \stackrel{\Delta}{=} \lambda_{\max}\tilde{F}(x,y,W) \tag{5.25}$$

over $x \in \mathcal{R}_x$, $y \in \mathcal{R}_y$ and $W \in \mathcal{W}_{\mathcal{R}_x,\mathcal{R}_y}(x,y)$. Instead of solving this problem an approximation can be build up using a linearization to $\varphi(x,y,W)$ at a given point $\hat{x},\hat{y},\hat{W}$. A linearization of $\varphi(x,y,W)$ at a given point can be found using *generalized gradients*, see [Cla83]. Let $\partial\varphi(\hat{x},\hat{y},\hat{W})$ be the set of generalized gradients at the point $(\hat{x},\hat{y},\hat{W})$. We then have the linearization of φ as have

$$h_{\hat{x},\hat{y},\hat{W}}(x,y,W) \stackrel{\Delta}{=} \langle \zeta, \left(x-\hat{x}, y-\hat{y}, W-\hat{W}\right) \rangle + \varphi(\hat{x},\hat{y},\hat{W})$$

where $\zeta \in \partial\varphi(\hat{x},\hat{y},\hat{W})$. In [THT96] Tuan et al give one generalized gradient. It can be found in the following way. For simplicity we consider a linear matrix function $A(x) = A_0 + \sum_{i=1}^m x_i A_i$, and the function $\varphi_x(x) = \lambda_{\max} A(x)$. Let λ be the maximum eigenvalue of $A(\hat{x})$ and v the corresponding eigenvector. The eigenvalue λ and the eigenvector v fulfill by definition

$$(A(x) - \lambda I)\, v = 0 \tag{5.26}$$

at $x = \hat{x}$. Even though the function φ_x is not even differentiable, we use the derivative rules to derived a generalized gradient. Using the partial derivative rules for x_i on both sides we get the following

$$\begin{aligned} 0 &= \frac{\partial}{\partial x_i}\{(A(x) - \lambda I)\, v\}\\ &= \frac{\partial}{\partial x_i}\{A(x) - \lambda I\}\, v + (A(x) - \lambda I)\frac{\partial}{\partial x_i} v\\ &= \left(A_i - \frac{\partial}{\partial x_i}\lambda I\right) v + (A(x) - \lambda I)\frac{\partial}{\partial x_i} v \end{aligned}$$

Multiplying with v^T from the left we get

$$\begin{aligned} 0 &= v^T \left(A_i - \frac{\partial}{\partial x_i}\lambda I \right) v + \underbrace{v^T \left(A(x) - \lambda I\right)}_{=0 \text{ by } (5.26)} \frac{\partial}{\partial x_i} v \\ &= v^T A_i v - \frac{\partial}{\partial x_i}\lambda I v^T v \ \ . \end{aligned}$$

which gives

$$\frac{\partial}{\partial x_i}\lambda = \frac{v^T A_i v}{v^T v}$$

It can be shown [THT96] that

$$\zeta = \left(\frac{v^T A_1 v}{v^T v}, \frac{v^T A_2 v}{v^T v}, \cdots \frac{v^T A_m v}{v^T v} \right) \in \zeta \in \partial\varphi(\hat{x}) \ \ . \tag{5.27}$$

Since $\varphi(x,y,W)$ is convex we have

$$\varphi(x,y,W) \geq h_{\hat{x},\hat{y},\hat{W}}(x,y,W) \ \forall, x, y, W$$

A collection of points $\hat{x}^k, \hat{y}^k, \hat{W}^k$ will provide an approximation of $\varphi(x,y,W)$ by

$$\varphi(x,y,W) \geq \max_k h_{\hat{x}^k,\hat{y}^k,\hat{W}^k}(x,y,W) \ \ .$$

Partition

The partition rule follows the same as the partition rules given above for the method of Tuan.

5.6 Lower bound via Lagrangian duality

In this section we shall develop lower bounds using Lagrangian duality. The theory behind Lagrangian duality can be found in for instance [Ber95] or [BV97a]. For the general BMI problem equation (5.2) the Lagrangian looks like the following

$$L(x,y,\Gamma,\Delta) \triangleq c^T x + d^T y - \operatorname{Tr}\Gamma F(x,y) - \operatorname{Tr}\Delta G(x,y), \tag{5.28}$$

for $F(x,y) \geq 0$, $G(x,y) \geq 0$, where $\Gamma \geq 0$, $\Delta \geq 0$. We refer to Γ and Δ as the *Lagrange multipliers* associated with the constraint $F(x,y) \geq 0$, and $G(x,y) \geq 0$ respectively.

Since $\Gamma \geq 0$ and $F(x,y) \geq 0$ then $\operatorname{Tr}\Gamma F(x,y) \geq 0$. The same holds for $\Delta G(x,y)$ and we have the relation

$$c^T x + d^T y \geq L(x,y,\Gamma,\Delta), \quad \forall x,y \in \mathcal{F} \cap \mathcal{G} \ \ . \tag{5.29}$$

5.6.1 Relaxation of regions

Going back to $v(y)$ then we have the following lower bound on $v(y)$:

$$\begin{aligned} v(y) &= \inf_x [f(x,y) \text{ subject to } F(x,y) \geq 0,\ G(x,y) \geq 0] \\ &\geq \inf_x [L(x,y,\Gamma,\Delta) \text{ subject to } F(x,y) \geq 0,\ G(x,y) \geq 0], \forall \Gamma, \Delta \geq 0 \end{aligned} \tag{5.30}$$

In the above we have the constraints $(x,y) \in \mathcal{F} \cap \mathcal{G}$. If we relax this to $(x,y) \in \mathcal{X}_G \times \mathcal{Y}_G$, we still get a valid lower bound:

$$v(y) \geq \inf_{x \in \mathcal{X}_G} L(x,y,\Gamma,\Delta), \forall y \in \mathcal{Y}_G, \Gamma, \Delta \geq 0 \tag{5.31}$$

In the following the Lagrange variables are fixed in the positive semidefinite cone. With this in mind we take a closer look at the Lagrangian $L(x,y,\Gamma,\Delta)$. Since we want to find a lower bound for $v(y)$ we want to eleminate the variables x. For this reason rewrite the Lagrangian as:

$$\begin{aligned} L(x,y,\Gamma,\Delta) &= c^T x + d^T y - \operatorname{Tr}\Gamma F(x,y) - \operatorname{Tr}\Delta G(x,y) \\ &= d^T y - \operatorname{Tr}\Gamma\left(F_0 + \sum_{j=1}^{m_y} y_j F_j^y\right) - \operatorname{Tr}\Delta\left(G_0 + \sum_{j=1}^{m_y} y_j G_j^y\right) \\ &+ \sum_{i=1}^{m_x} x_i \left(c_i - \operatorname{Tr}\Gamma\left(F_i^x + \sum_{j=1}^{m_y} y_j F_{ij}^{xy}\right) - \operatorname{Tr}\Delta G_i^x\right) \end{aligned} \tag{5.32}$$

Define the following to functions

$$a(y,\Gamma,\Delta) \triangleq d^T y - \operatorname{Tr}\Gamma\left(F_0 + \sum_{j=1}^{m_y} y_j F_j^y\right) - \operatorname{Tr}\Delta\left(G_0 + \sum_{j=1}^{m_y} y_j G_j^y\right) \tag{5.33}$$

$$b_i(y,\Gamma,\Delta) \triangleq c_i - \operatorname{Tr}\Gamma\left(F_i^x + \sum_{j=1}^{m_y} y_j F_{ij}^{xy}\right) - \operatorname{Tr}\Delta G_i^x,\ i = 1,\cdots,m_x, \tag{5.34}$$

then we can write the Lagrangian in the more simple form

$$L(x,y,\Gamma,\Delta) = a(y,\Gamma,\Delta) + \sum_{i=1}^{m_x} x_i b_i(y,\Gamma,\Delta)$$

Recall that we are interested in the infimum of $L(x,y,\Gamma,\Delta)$ over $\mathcal{X}_G$, but since the functions a and b are independent of x we have the following

$$\begin{aligned} \inf_{x \in \mathcal{X}_G} L(x,y,\Gamma,\Delta) &= \inf_{x \in \mathcal{X}_G} \left\{ a(y,\Gamma,\Delta) + \sum_{i=1}^{m_x} x_i b_i(y,\Gamma,\Delta) \right\} \\ &= a(y,\Gamma,\Delta) + \inf_{x \in \mathcal{X}_G} \sum_{i=1}^{m_x} x_i b_i(y,\Gamma,\Delta) \end{aligned} \tag{5.35}$$

Our goal is to find computable lower bounds on $v(y)$ for $y \in \mathcal{G}$. Since a is an affine function of y for fixed Lagrange variables Γ, Δ finding the minimum over $y \in \mathcal{G}$ is a convex problem and therefore tractable. However, finding the minimum over $y \in \mathcal{Q}$ for $\inf_{x \in \mathcal{X}_G} \sum_{i=1}^{m_x} x_i b_i(y,\Gamma,\Delta)$, is not a convex problem due to the bilinear connection between x and y. We will resolve this problem by relaxing the region of $\mathcal{X}_G$ to be a region of the form

$$\begin{aligned} \mathcal{R}_p &\triangleq \{x: \left\|\Lambda^{-1}(x-\hat{x})\right\|_p \leq 1\} \subset \mathbb{R}^{m_x}, \\ &\text{where } \Lambda = \mathrm{Diag}\left(\begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_{m_x} \end{bmatrix}\right) > 0, p = 1,2,\infty \ . \end{aligned} \tag{5.36}$$

We are interested in the region with smallest volume. For $p = \infty$ this was shown in last subsection. For $p = 1$ it can be done approximately by solving the same $2m_x$ SDP problems, as for obtaining $p = \infty$ and then using that if

$$\mathcal{C} \supset \left\{x: \left\|\Lambda^{-1}(x-\hat{x})\right\|_\infty \leq 1\right\} \Rightarrow \mathcal{C} \supset \left\{x: \left\|(2\Lambda)^{-1}(x-\hat{x})\right\|_1 \leq 1\right\} \ .$$

By doing this relaxation of $\mathcal{X}_G$ to $\mathcal{R}_p$ we obtain the following

$$\begin{aligned} v(y) &\geq a(y,\Gamma,\Delta) + \inf_{x\in\mathcal{X}_G} \sum_{i=1}^{m_x} x_i b_i(y,\Gamma,\Delta) \\ &\geq a(y,\Gamma,\Delta) + \inf_{x\in\mathcal{R}_p} \sum_{i=1}^{m_x} x_i b_i(y,\Gamma,\Delta),\, y\in\mathcal{Y}_G,\ p = 1,2,\infty, \end{aligned} \tag{5.37}$$

The map $\tilde{x} = \Lambda^{-1}(x-\hat{x})$ is bijective since $\Lambda > 0$ and have the feature: if $\tilde{x} \in \left\{\tilde{x} : \|\tilde{x}\|_p \leq 1\right\}$ then $x \in \mathcal{R}_p$. Using the change of variables $\tilde{x} = \Lambda^{-1}(x-\hat{x})$ and introducing

$$\begin{aligned} \tilde{a}(y,\Gamma,\Delta) &\triangleq a(y,\Gamma,\Delta) - \sum_{i=1}^{m_x} \hat{x}_i \lambda_i^{-1} b_i(y,\Gamma,\Delta) \\ \tilde{b}_i(y,\Gamma,\Delta) &\triangleq \lambda_i^{-1} b_i(y,\Gamma,\Delta), i = 1,\cdots,m_x \end{aligned} \tag{5.38}$$

we get the following reformulation of the lower bound

$$v(y) \geq \tilde{a}(y,\Gamma,\Delta) + \inf_{\|\tilde{x}\|_p\leq 1} \sum_{i=1}^{m_x} \tilde{x}_i \tilde{b}_i(y,\Gamma,\Delta),\, y\in\mathcal{X}_G,\ p = 1,2,\infty \ . \tag{5.39}$$

By this we have justified that we can put the problem on the form 5.39 independently of the desired norm bound on x. In the following we will simplify our notation to $a(y) = \tilde{a}(y,\Gamma,\Delta)$ and each $b_i(y) = \tilde{b}_i(y,\Gamma,\Delta),\ i = 1,\cdots,m_x$. Stacking $b_i(y)$ in a column vector results in the variable we will naturally denote $b(y)$. We will also remove the tilde on x. We now consider the relation

$$v(y) \geq a(y) + \inf_{\|x\|\leq 1} x_i b_i(y) \tag{5.40}$$

The next step in our derivation of lower bounds is to rewrite the sum of $x_i b_i(y)$, as an inner product between x and b. We can restrict our self to the infimum over x so we look at

$$\inf_{\|\tilde{x}\|_p\leq 1} \sum_{i=1}^{m_x} \tilde{x}_i \tilde{b}_i(y) = \inf_{\|\tilde{x}\|_p\leq 1} \langle x, b(y)\rangle \ . \tag{5.41}$$

We now have the following relations due to Hölders inequality

$$\begin{aligned} \inf_{\|x\|_p\leq 1} \langle x, b(y)\rangle &\geq -\sup_{\|x\|_p\leq 1} |\langle x, b(y)\rangle| \\ &\geq -\|b(y)\|_q, \frac{1}{q} + \frac{1}{p} = 1. \end{aligned} \tag{5.42}$$

Using the result of equation (5.42) on equation (5.39) we obtain the following

$$v(y) \geq a(y) - \|b(y)\|_q, \forall y \in \mathcal{Y}_G \tag{5.43}$$

where q relates to the bound on x by $\frac{1}{q} + \frac{1}{p} = 1$. To find a valid lower bound over $\mathcal{Y}_G$ we have to compute the infimum over $\mathcal{Y}_G$ for the right hand side in equation (5.43). We will need different approaches for each p,q to compute this lower bound using convex problems.

Lower bound for x bounded in max norm

For $p = \infty$ we have $q = 1$ and we need to compute the 1-norm of $b(y)$. We recall the 1-norm

$$\|b(y)\|_1 = \sum_{i=1}^{m_x} |b_i(y)|$$

Recall that any norm is a convex function, but we basically want to maximize the norm, or more precisely minimize $-\|b(y)\|_1$, this is a non-convex problem. The problem is the absolute value $|b_i(y)|$. If $b_i(y) \geq 0$ we do not have to take the absolute value, this restrict our attention to one half space of $\mathbb{R}^{m_y}$. In the opposite half space $-b_i(y) \geq 0$ we have the opposite sign of b_i in the sum. For this reason introduce a sign vector $s \in \{-1,1\}^{m_x}$, and we can write an equivalent formulation of the 1-norm. Given y find s such that $s_i b_i(y) \geq 0,\ i = 1,\cdots,m_x$ then

$$\|b(y)\|_1 = \sum_{i=1}^{m_x} s_i b_i(y). \tag{5.44}$$

That is we have reformulated the computation of the 1-norm as a computation of a linear sum of m_x elements in one of the corners of the unit cube. Which corner we have to consider depends on the sign of $b_i(y)$. Inserting this result in equation (5.43) and taking the infimum over $y \in \mathcal{Y}_G$ we get the following valid lower bound

$$\inf_{y\in\mathcal{Y}_G} v(y) \geq \inf_{s\in\{-1,1\}^{m_x}} \inf_{y\in\mathcal{Y}_G} \left[\tilde{a}(y,\Gamma,\Delta) - \sum_{i=1}^{m_x} s_i \tilde{b}_i(y,\Gamma,\Delta), s_i\tilde{b}_i(y,\Gamma,\Delta) \geq 0, i = 1,\cdots,m_x\right] \tag{5.45}$$

Thus by solving 2^{m_x} subproblems we can find a lower bound for f^*. Each subproblem is an SDP on the form

$$\begin{array}{ll} \text{minimize} & \tilde{a}(y,\Gamma,\Delta) - \sum_{i=1}^{m_x} s_i \tilde{b}_i(y,\Gamma,\Delta) \\ \text{subject to} & s_i\tilde{b}_i(y,\Gamma,\Delta) \geq 0, i = 1,\cdots,m_x \\ & y \in \mathcal{Y}_G \end{array} \tag{5.46}$$

Denote the optimum of each subproblem by ϕ_s, and the y achieving it by y_s. The lower bound for $v(y)$ over $\mathcal{Y}_G$ is the following

$$f^* \geq \min_s [\phi_s, \text{ subject to } s \in \{-1,1\}^{m_x}] \tag{5.47}$$

The drawback here is the big number of SDP problems we have to solve. The number grows exponentially with the number of x variables. The same is not true if we choose to bound $\mathcal{X}_G$ in the 1-norm instead.

Lower bound for x bounded in 1-norm

For $p = 1$ we have $q = \infty$ and we consider the max-norm:

$$||b(y)||_\infty = \max_{i=1,\cdots,m_x} |b_i(y)|$$

Since we are only interested in the maximum absolute value of all $b_i(y)$ we can find this by looking at $2m_x$ situations. This means that we can find a lower bound for f^* by solving the $2m_x$ SDP problems. For each entry in b consider the two possible signs. Suppose $\tilde{b}(y,\Gamma,\Delta) \geq 0$ solve the following problem

$$\begin{array}{ll} \text{minimize} & \tilde{a}(y,\Gamma,\Delta) - \tilde{b}_i(y,\Gamma,\Delta) \\ \text{subject to} & \tilde{b}_i(y,\Gamma,\Delta) \geq 0 \\ & y \in \mathcal{Y}_G \end{array} \tag{5.48}$$

denote the optimum by ψ_i^+ and the optimal y by $\hat{y}_i^+$, similarly solve the problem

$$\begin{array}{ll} \text{minimize} & \tilde{a}(y,\Gamma,\Delta) + \tilde{b}_i(y,\Gamma,\Delta) \\ \text{subject to} & -\tilde{b}_i(y,\Gamma,\Delta) \geq 0 \\ & y \in \mathcal{Y}_G \end{array} \tag{5.49}$$

and denote the optimum by ψ_i^- and the optimal y by $\hat{y}_i^+$. We now have the following simple relation

$$f^* \geq \min_i \min[\psi_i^+, \psi_i^-] \tag{5.50}$$

5.6.2 Optimal Lagrange multipliers and partition

In the previous subsection we considered Γ and Δ to be fixed. Here we will show that a specific choice lead to a nice structure of $a(y,\Gamma,\Delta)$ and $b_i(y,\Gamma,\Delta)$, which again lead to a set of nice properties of the lower bounds.

Consider the problem (5.12) with $y = \hat{y}$ fixed:

$$\begin{array}{ll} \text{minimize} & c^T x + d^T \hat{y} \\ \text{subject to} & F(x,\hat{y}) = \left(F_0 + \sum_{j=1}^{m_y} \hat{y}_j F_j^y\right) + \sum_{i=1}^{m_x} x_i \left(F_i^x + \sum_{j=1}^{m_y} \hat{y}_j F_{ij}^{xy}\right) \geq 0 \\ & G(x,\hat{y}) = \left(G_0 + \sum_{j=1}^{m_y} \hat{y}_j G_j^y\right) + \sum_{i=1}^{m_x} x_i G_i^x \geq 0 \ \ . \end{array} \tag{5.51}$$

We will call this problem the *primal restricted problem*. Note, that due to assumption 5.2 the above problem is always feasible.

The Lagrangian associated with the primal restricted problem is

$$L(x,\hat{y},\Gamma,\Delta) = c^T x + d^T \hat{y} - \operatorname{Tr}\Gamma F(x,\hat{y}) - \operatorname{Tr}\Delta G(x,\hat{y}), \tag{5.52}$$

The so called Lagrange dual function

$$g(\hat{y},\Gamma,\Delta) = \inf_x \left[c^T x + d^T \hat{y} - \operatorname{Tr}\Gamma F(x,\hat{y}) - \operatorname{Tr}\Delta G(x,\hat{y})\right], \ \ \Gamma > 0, \Delta > 0 \tag{5.53}$$

provides a lower bound on the primal problem. Looking at (5.32) it is straight forward to rewrite the Lagrange dual function as

$$\begin{aligned} g(\hat{y},\Gamma,\Delta) &= d^T y - \operatorname{Tr}\Gamma\left(F_0 + \sum_{j=1}^{m_y} y_j F_j^y\right) - \operatorname{Tr}\Delta\left(G_0 + \sum_{j=1}^{m_y} y_j G_j^y\right) \\ &\quad + \inf_x \sum_{i=1}^{m_x} x_i \left(c_i - \operatorname{Tr}\Gamma\left(F_i^x + \sum_{j=1}^{m_y} y_j F_{ij}^{xy}\right) - \operatorname{Tr}\Delta G_i^x\right) \\ &= a(\hat{y},\Gamma,\Delta) + \inf_x \sum_{i=1}^{m_x} x_i b_i(\hat{y},\Gamma,\Delta) \end{aligned} \tag{5.54}$$

where the last equivalence follows from the definition of a and b_i in (5.33) and (5.34). Since the infimum over x in (5.54) is an unconstrained optimization problem the optimum is finite only if the partial derivatives with respect to x are zero, which lead to

$$g(\Gamma,\Delta) = \begin{cases} a(\hat{y},\Gamma,\Delta) & \text{if } b_i(\hat{y},\Gamma,\Delta) = 0, i = 1,\cdots,m_x,\ \Gamma > 0, \Delta > 0 \\ -\infty & \text{otherwise} \end{cases} \tag{5.55}$$

Note that $g(\Gamma,\Delta)$ provides a lower bound on the primal restricted problem. It is therefore interesting to find the best lower bound. We can do this using the following problem

$$\begin{array}{ll} \text{maximize} & a(\hat{y},\Gamma,\Delta) \\ \text{subject to} & b_i(\hat{y},\Gamma,\Delta) = 0, i = 1,\cdots,m_x, \\ & \Gamma > 0, \Delta > 0 \end{array} \tag{5.56}$$

Denote the optimum Lagrange multipliers in (5.56) by $\hat{\Gamma}$ and $\hat{\Delta}$. The optimum is $a(\hat{y},\hat{\Gamma},\hat{\Delta})$, which is a lower bound on $v(\hat{y})$, that is *weak duality* holds. Since the $\mathcal{X}_G$ is bounded, the dual is strictly feasible, and *strong duality* holds, and we get $v(\hat{y}) = a(\hat{y},\hat{\Gamma},\hat{\Delta})$. This particular solution to Γ and Δ gives a very nice structure in $a(y,\Gamma,\Delta)$ and $b_i((y,\Gamma,\Delta)$ First consider a:

$$\begin{aligned} a(y,\hat{\Gamma},\hat{\Delta}) &= d^T y - \operatorname{Tr}\hat{\Gamma}\left(F_0 + \sum_{j=1}^{m_y} y_j F_j^y\right) - \operatorname{Tr}\hat{\Delta}\left(G_0 + \sum_{j=1}^{m_y} y_j G_j^y\right) + v(\hat{y}) - a(\hat{y},\hat{\Gamma},\hat{\Delta}) \\ &= v(\hat{y}) + d^T(y-\hat{y}) - \sum_{j=1}^{m_y}\left(\operatorname{Tr}\hat{\Gamma}F_j^y + \operatorname{Tr}\hat{\Delta}G_j^y\right)(y_j - \hat{y}_j) \end{aligned} \tag{5.57}$$

Introducing the vector $p \in \mathbb{R}^{m_y}$ defined by

$$p_j \triangleq d_j - \operatorname{Tr}\hat{\Gamma}F_j^y + \operatorname{Tr}\hat{\Delta}G_j^y \ ,$$

we can write a as

$$a(y,\hat{\Gamma},\hat{\Delta}) = v(\hat{y}) + p^T(y-\hat{y}) \ . \tag{5.58}$$

There is a couple of things to say about the function $a(y,\hat{\Gamma},\hat{\Delta})$. First of all it is a hyper plane and secondly it touches $v(y)$ at $y = \hat{y}$. Note, that p is independent of F_0 and G_0.

Similarly we can simplify b_i as

$$b_i(y,\Gamma,\Delta) = q_i^T(y-\hat{y}), i = 1,\cdots,m_x \ . \tag{5.59}$$

where q_i is the i'th column in the matrix Q defined by

$$\{Q\}_{ji} \stackrel{\Delta}{=} \mathrm{Tr}\,\Gamma F_{ij}^{xy} \quad . \tag{5.60}$$

Note the indices of Q and F_{ij}^{xy} are swapped.

By looking at q_i it is possible to detect if there is a bilinear connection between a variable x_i in x and the variables in y. In other words is x_i a complicating variable. We will call a variable x_i where $q_i \neq 0$ for a *connected variable*.

Inserting the result of (5.58) and (5.59) in (5.38) we obtain the following

$$\tilde{a}(y,\hat{\Gamma},\hat{\Delta}) = v(\hat{y}) + p^T(y-\hat{y}) - \sum_{i=1}^{m_x} \hat{x}_i \lambda_i^{-1} q_i^T (y-\hat{y})$$
$$\tilde{b}_i(y,\hat{\Gamma},\hat{\Delta}) = \lambda_i^{-1} q_i^T (y-\hat{y})$$

Note that $\tilde{a}(y,\hat{\Gamma},\hat{\Delta})$ also is a hyper plane touching $v(y)$ at $v(\hat{y})$.

Lower bound for x bounded in max-norm

To obtain a lower bound using the max-norm, it was necessary to check all 2^{m_x} corners in the unit cube. The number of corners can be reduced if the number of connected variables is smaller than m_x. For each corner s the subproblem (5.46) was solved with the constraints $s_i\tilde{b}_i(y,\Gamma,\Delta) \geq 0, i = 1,\cdots,m_x$. If x_i is a connected variable $b_i(y,\Gamma,\Delta) = 0$ and the sign vector can be chosen arbitrarily. With the choice of $\Gamma = \hat{\Gamma}$ and $\Delta = \hat{\Delta}$ these constraints define a polytope in $\mathbb{R}^{m_y}$. We will define this polytope as

$$\mathcal{Q}_s \stackrel{\Delta}{=} \{y : \; s_i q_i^T (y-\hat{y}) \geq 0, i = 1,\cdots,m_x\} \tag{5.61}$$

An example with $m_x = 2$ can be seen in figure 5.4. Note that $\mathrm{Int}\,\mathcal{Q}_s \cap \mathrm{Int}\,\mathcal{Q}_r = \emptyset, r \neq s$, and

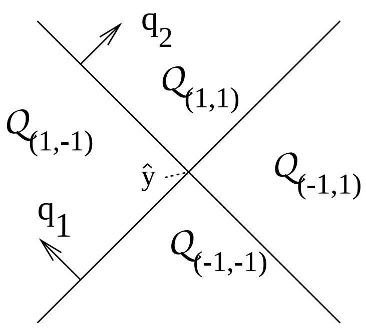


Figure 5.4: A star-shaped partition.
The partition has four regions, since $m_x = 2$ and $y \in \mathbb{R}^2$.

$\bigcup_{s\in\{-1,1\}^{m_x}} \mathcal{Q}_s = \mathbb{R}^{m_x}$, thus the collection $\{\mathcal{Q}_s \cap \mathcal{Y}_G\}$ fulfills the condition for a partition as required in the branch and bound algorithm. Note that the partition $\{\mathcal{Q}_s\}$ has a star-shaped form, with $\hat{y}$ as the center point.

We will now consider a more direct way of finding the lower bounds when x is bounded in $\mathcal{R}_\infty$. We consider a formulation of $\mathcal{R}_\infty$ in terms of lower left $l \in \mathbb{R}^{m_x}$ and upper right $l \in \mathbb{R}^{m_x}$

corners, where

$$\begin{aligned} l_i &\triangleq \hat{x}_i - \lambda_i,\ i = 1,\cdots,m_x \\ u_i &\triangleq \hat{x}_i + \lambda_i,\ i = 1,\cdots,m_x \end{aligned} \tag{5.62}$$

and

$$\begin{aligned} \lambda_i &= \frac{u_i - l_i}{2},\ i = 1,\cdots,m_x \\ \hat{x}_i &= \frac{u_i + l_i}{2},\ i = 1,\cdots,m_x \end{aligned} \tag{5.63}$$

This gives the following formulation:

$$\mathcal{R}_\infty = \{x:\ l_i \le x_i \le u_i,\ \ i = 1,\cdots,m_x\}$$

If we go back to the objective of finding the minimum over $\mathcal{X}_G$ of the Lagrangian (5.35), we can obtain a result more directly. For each corner in $\mathcal{R}_\infty$ defined by $s \in \{-1,1\}^{m_x}$ as $\hat{x} + \Lambda s$, we have to consider the region Q_s, and we get the following lower bound valid

$$\begin{aligned} \inf_{x\in\mathcal{X}_G} L(x,y,\Gamma,\Delta) &\ge a(y,\hat{\Gamma},\hat{\Delta}) + \inf_{x\in\mathcal{R}_\infty} \sum_{i=1}^{m_x} x_i b_i(y,\hat{\Gamma},\hat{\Delta}) \\ &= a(y,\hat{\Gamma},\hat{\Delta}) + \inf_{s\in\{-1,1\}^{m_x}} \sum_{i=1}^{m_x} \left(\frac{u_i+l_i}{2} + \frac{u_i-l_i}{2}s_i\right) b_i(y,\hat{\Gamma},\hat{\Delta}) \end{aligned} \tag{5.64}$$

For each sign vector $s \in \{-1,1\}^{m_x}$ we have to solve the following subproblem:

$$\begin{array}{ll} \text{minimize} & v(\hat{y}) + p^T(y-\hat{y}) + \sum_{i=1}^{m_x} \left(\frac{u_i-l_i}{2} - \frac{u_i-l_i}{2}s_i\right) q_i^T(y-\hat{y}) \\ \text{subject to} & s_i q_i^T(y-\hat{y}) \ge 0, i = 1,\cdots,m_x \\ & y \in \mathcal{Y}_G \end{array} \tag{5.65}$$

Again the optimum is denoted by ϕ_s, and the y that achieves it by y_s.

For each $s \in \{-1,1\}^{m_x}$ we will define a function $\phi_s:\ Q_s \to \mathbb{R}$ by

$$\phi_s(y) \triangleq v(\hat{y}) + p^T(y-\hat{y}) + \sum_{i=1}^{m_x} \left(\frac{u_i+l_i}{2} - \frac{u_i-l_i}{2}s_i\right) q_i^T(y-\hat{y}), y \in Q_s.$$

Putting all $\phi_s(y), s \in \{-1,1\}^{m_x}$ together we get a function $\phi:\ \mathbb{R}^{m_y} \to \mathbb{R}$, that we denote $\phi(y)$. Going back to the definition of the 1-norm we can write $\phi(y)$ as

$$\phi(y) = v(\hat{y}) + p^T(y-\hat{y}) + \sum_{i=1}^{m_x} \frac{u_i+l_i}{2} q_i^T(y-\hat{y}) - \left\| \frac{u_i-l_i}{2} Q(y-\hat{y}) \right\|_1, y \in Q. \tag{5.66}$$

Note, that since $\phi_s(y)$ is linear, then $\phi(y)$ is piecewise linear.

The function $\phi(y)$ consists of two parts, a linear function minus the norm of linear function. Since the norm is convex a convex function, the norm of a linear function of is convex. Putting a minus in front of the norm makes it a concave function. Since a linear function is concave, and the sum of two concave function is also concave we have that $\phi(y)$ is concave. The function ϕ is also continuous.

We summarize the properties of ϕ as follows

$$\phi \begin{cases} \text{is piece-wise linear} \\ \text{is concave} \\ \text{is continuous} \\ \text{touches } v \text{ at } \hat{y}\text{, i.e. } \phi(\hat{y}) = v(\hat{y}) \\ \text{is a lower bounding (LB) function } \phi(\hat{y}) \leq v(y),\ \forall y \in \mathcal{Y}_G \end{cases} \tag{5.67}$$

We will call ϕ for *lower bounding function* refering to the fifth property above. In fact one can think of ϕ as a pyramid shaped function lying below $v(y)$. We will now add a superscript on ϕ to reflect the point $\hat{y}$ and we get the notation $\phi^{\hat{y}}$.

Example 5.3 (Simple ϕ function) *Suppose $m_y = 2$, $\hat{y} = [0\ 0]^T$, $p = [0\ 0]^T$, x bounded in the unit cube in max norm, and $Q = \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}$, then*

$$\phi(y) = -|1y_1| - |3y_2| \quad .$$

A plot can be seen in figure 5.5.

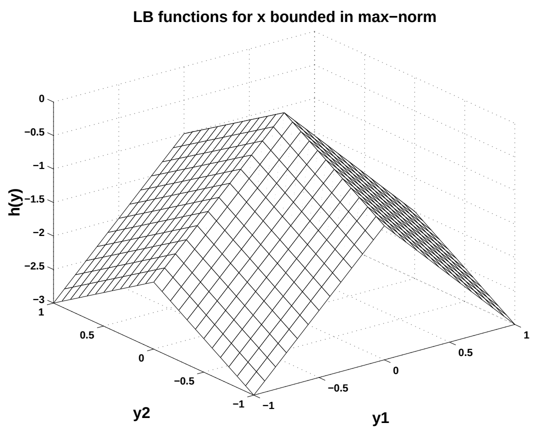


Figure 5.5: A simple LB function

Partition

We have now derived a star-shaped partition of the region $\mathcal{Y}_G$, with $\hat{y}$ as the center point. we have showed how we can derive lower bounding functions in each region of the partition. The lower bounding functions can be used to find lower bounds, that can be used in the branch and bound algorithm. One might object to split up the region in the 2^{m_x} subregions, the star-shaped partition suggest. There is one major advantages the function $\phi(y)$ that underestimates $v(y)$ over all $\mathcal{Y}_G$ is convex in each subregion Q_s. This mean that if we in future iterations should consider to split Q_s again, and calculate upper and lower bounds, we can use $\phi_s(y)$ that is linear to improve the lower bound. That is the lower bounding functions for a region Q from previous iterations can be inherited down to any subpartition of Q. Before we describe this type of inheritance more deeply, we will study a couple of lower bounding function for a simple BMI problem:

Example 5.4 (Sahinidis and Grossman II) *Again we consider the example from [SG91], see example 5.2 We now consider the Lagrangian of the above problem. Associated with the lower bounds $x \geq 0$, $y \geq 0$ we use the Lagrange multiplier $\mu \in \mathbb{R}^2$, and with the upper bounds $\nu \in \mathbb{R}^2$. The multiplier associated with the bilinear constraint is denoted by $w \in \mathbb{R}$. We get the following Lagrangian:*

$$\begin{aligned} L(x,y,w,\mu,\nu) &= -x-y-w(4-xy) - \begin{bmatrix} \mu_1 & \mu_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} \nu_1 & \nu_2 \end{bmatrix} \begin{bmatrix} 6-x \\ 4-y \end{bmatrix} \\ &= -4w - 6\nu_1 - 4\nu_2 + y(-1-\mu_2+\nu_2) \\ &\quad + x(-1+wy-\mu_1+\nu_1), \;\; w \geq 0, \nu \geq 0, \mu \geq 0 \end{aligned} \tag{5.68}$$

We consider different $\hat{y}$. Due to complementary slackness a number of Lagrange multipliers will be zero. We get the following values of $v(\hat{y})$, optimal $\hat{x}$, and optimal Lagrange multipliers:

$\hat{y}$	$\hat{x}$	$v(\hat{y})$	$\hat{w}$	$\hat{\mu}_1$	$\hat{\mu}_2$	$\hat{\nu}_1$	$\hat{\nu}_2$	$a(y,\hat{w},\hat{\mu},\hat{\nu})$	$b_1(y,\hat{w},\hat{\mu},\hat{\nu})$
0	6	-6	0	0	≥ 0	1	0	$-6+(-1-\hat{\mu}_2)(y-0)$	0
2/3	6	$-20/3$	3/2	0	0	0	0	$-20/3+(-1)(y-2/3)$	$(3/2)(y-2/3)$
2	2	-4	1/2	0	0	1	0	$-4+(-2)(y-2)$	$1/2(y-2)$
4	1	-5	0	0	0	0	≥ 0	$-5+(-1+\hat{\nu}_2)(y-4)$	$1/4(y-4)$

≥ 0 *indicates that the multiplier was unconstrained.*

Using this we can obtain the following lower bounding functions on $v(y)$

$$\phi^0(y) = -6+(-1-\hat{\mu}_2)(y-0) \textit{ for } y \geq 0$$

$$\phi^{2/3}(y) = \begin{cases} -20/3+(-1)(y-2/3)+0(3/2)(y-2/3) & \textit{for } y-2/3 \geq 0 \\ -20/3+(-1)(y-2/3)+6(3/2)(y-2/3) & \textit{for } y-2/3 \leq 0 \end{cases}$$

$$\phi^2(y) = \begin{cases} -4+(-1)(y-2)+0(1/2)(y-2) & \textit{for } y-2 \geq 0 \\ -4+(-1)(y-2)+6(1/2)(y-2) & \textit{for } y-2 \leq 0 \end{cases}$$

$$\phi^4(y) = \begin{cases} -5+(-1+\hat{\nu}_2)(y-4)+6(1/4)(y-4) & \textit{for } y-4 \leq 0 \end{cases}$$

These lower bounding functions are plotted in figure 5.6. From the figures the lower bound on f^* can be seen. We get the following lower bounds by solving the subproblems

LB function	ϕ_{-1}	ϕ_1
ϕ^0	*NC*	$-10 = \phi(4)$
$\phi^{2/3}$	$-12 = \phi(0)$	$-10 = \phi(4)$
ϕ^2	$-8 = \phi(0)$	$-6 = \phi(4)$
ϕ^4	$-7 = \phi(0)$	*NC*

where NC denotes Not Calculated, since the corresponding region lies outside $0 \leq y \leq 4$.

Inheritance

Suppose we have a collection of k points $\hat{y}^l \in Q$, $l = 1, \cdots, k$. By calculating the LB functions extending from these points, we get a function approximating $v(y)$ from below. This function consist of the point-wise infimum of all computed LB functions.

$$v(y) \geq \sup_{l=1,\cdots,k} \phi^{\hat{y}^{(l)}}(y) \triangleq \Phi(y) \quad . \tag{5.69}$$

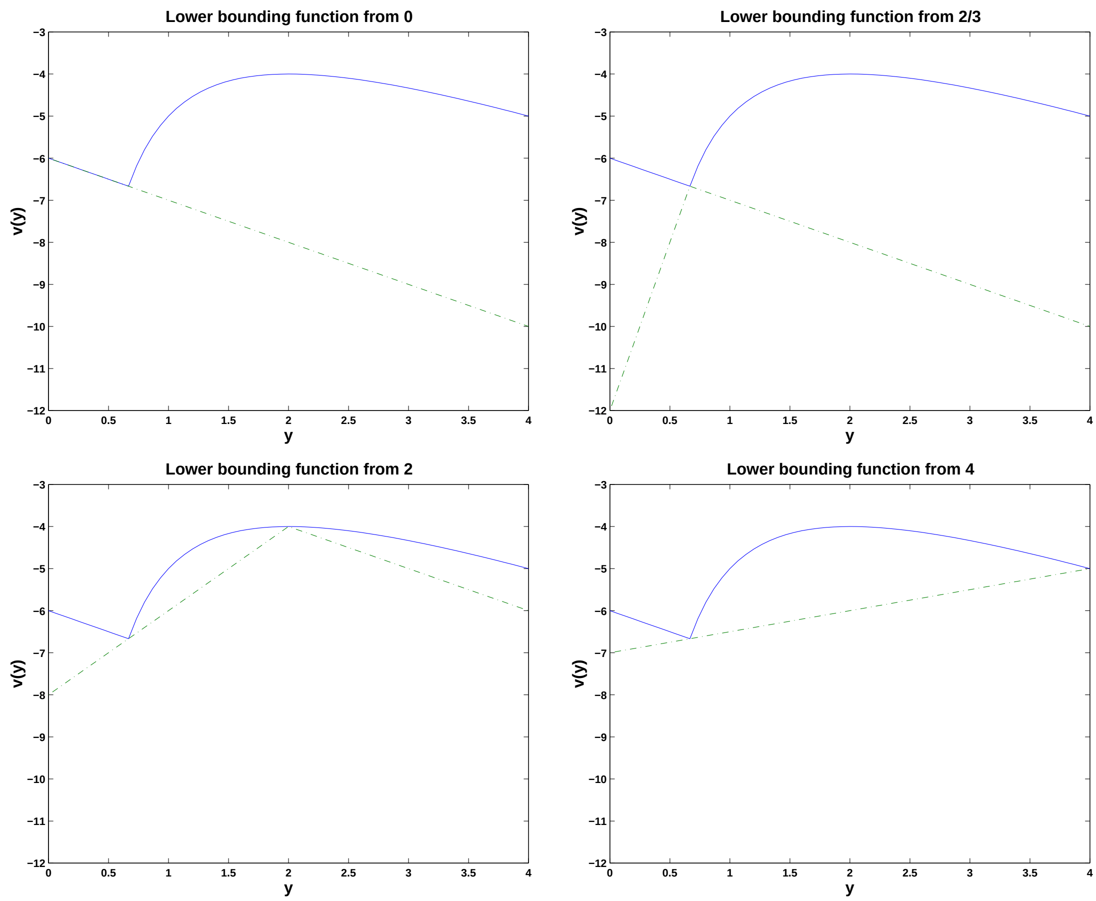


Figure 5.6: Lower bounding functions for the Sahinidis-Grossman problem.
Lower bounding functions around four different center points are plotted together with the optimal-value function (solid line).

The infimum of the function $\Phi(y)$ now represent the best lower bound (denoted $\underline{f}$) we can find over the region. The infimum of $v(\hat{y}^{(l)})$ over $l = 1, \cdots, k$ gives an upper bound $\overline{f}$ on $v(y)$ over the region Q. We have the following

$$\underline{f} = \inf_{y \in \mathcal{Y}_G} \Phi(y) \leq \inf_{y \in \mathcal{Y}_G} v(y) \leq \overline{f} \ . \tag{5.70}$$

The algorithm that we consider for solving the ε-optimal BMI optimization problem basically builds the function Φ until $\overline{f} - \underline{f} \leq \varepsilon$.

We now return to the Sahinidis Grossmann example.

Example 5.5 (Sahinidis Grossmann III) *Again we consider the example by Sahinidis and Grossmann. The following combination of ϕ^y provides exact upper and lower bounds for the global optimum:*

$$\begin{aligned} f^* &= \inf_{0 \leq y \leq 4} \inf \left[\phi^0(y), \phi^2(y)\right] = -20/3 \\ f^* &= \inf_{0 \leq y \leq 4} \inf \left[\phi^0(y), \phi^4(y)\right] = -20/3 \ , \end{aligned} \tag{5.71}$$

see figure 5.6.

Splitting point

When a new region is considered there should be a splitting point $\hat{y}$ available, that can be used both to solve the restricted problem, and to split the region around. Usually the optimal point y_s, from the subproblem over the present region can be used to split around. However, the solution to the subproblem is likely to lye on the boundary in the first couple of iterations. This will leave some of the regions in the partition big, and the algorithm might not converge, we will return to this issue on page 86. However, if we in each iteration use a point in the middle of the region then we are guaranteed that the volume of the regions in the partition goes to zero, which will ensure convergence. How do find a point in the middle of the region $\mathcal{Y}_G$? We shall here give two possibilities. Let $H(y)$ be an LMI formulation of the region $\mathcal{Y}_G$, such that $\mathcal{Y}_G = \{y : H(y) \geq 0\}$.

The following problem will find the most feasible point in the region:

$$\begin{array}{ll} \underset{y,t}{\text{minimize}} & t \\ \text{subject to} & H(y) + tI \geq 0 \end{array} \tag{5.72}$$

Also consider to solve the following maximum determinant problem [VBW96]:

$$\begin{array}{ll} \underset{y}{\text{minimize}} & \log\det H(y)^{-1} \\ \text{subject to} & H(y) \geq 0 \end{array} \tag{5.73}$$

The above problem (5.73) is a convex problem, and the optimal solution is usually refered to as the *analytical center*. More over the objective $\log\det H(y)^{-1}$ is a barrier function, that is it goes to infinity when $H(y)$ approaches the boundary of the positive semidefinite cone. There fore minimizing the objective $\log\det H(y)^{-1}$ centers y in the feasibility region $\{y : H(y) \geq 0\}$. Note however that the analytical center is dependent on the LMI formulation. So for instance given two constraint multiplying one of the constraints with a factor $\lambda \neq 1$ will move the analytical center.

Active variables

The number of subproblems is in general 2^{m_x} but can in some cases be reduced significantly. First of all if not all variables in x are connected with bilinear terms to y, then we only have to consider those that are. Let $\mathcal{A}$ be the set of indices to x where there is a bilinear connection with some variable in y. By considering Q we get

$$\mathcal{A}^{(k)} \triangleq \{i : q_i \neq 0\},$$

are the variables in x that do not have a bilinear connection with y. In other words Q can tell if there is a bilinear connection between x and y.

Dead half spaces

The number of subproblems can also be reduced by checking if $b_i(y) \geq 0$ or $b_i(y) \leq 0$ for the entire region $\mathcal{Y}_G$. If for instance the set $\mathcal{Y}_G \cap \{y : q_i(y - \hat{y}) \geq 0\}$ is empty, then all the problems with $s_i q_i^T (y - \hat{y}) \geq 0$, $s_i = 1$ are infeasible, and only problems with $s_i = -1$ can be feasible. This will cut down the number of subproblems to be solved by a factor of a half, and just by solving one simple feasibility problem. In other words if $m_x > 2$ then this might cause a significant reduction in the number of subproblems to be solved. An example can be seen in the Grossman-Sahinidis problem, we have two occasions denoted NC, where the corresponding regions lies outside $\mathcal{Y}_G$.

Lower bound for x bounded in 1-norm

When x was bounded in 1-norm, we had to calculate the max norm of $b(y,\Gamma,\Delta)$, which could be done by considering each entry $b_i(y,\Gamma,\Delta)$ separately. Inserting the optimal formulation derived above for $\tilde{a}(y,\Gamma,\Delta)$ and $\tilde{b}_i(y,\Gamma,\Delta)$ we get lower bounds, that have some additional properties.

To find the lower bound over $\mathcal{Y}_G$ we have to compute $2m_x$ SDP problems. For each entry $i=1,\cdots,m_x$ we to compute two problems. First the one corresponding to $b_i(y,\Gamma,\Delta)=q_i^T(y-\hat{y})$ being negative definite,

$$\begin{array}{ll} \underset{y}{\text{minimize}} & v(\hat{y})+p^T(y-\hat{y})+q_i^T(y-\hat{y}) \\ \text{subject to} & -q_i^T(y-\hat{y})\geq 0 \\ & y\in\mathcal{Y}_G \end{array} \tag{5.74}$$

denote the optimum by ψ_i^+ and the optimal y by $\hat{y}_i^+$, similarly for $b_i(y,\Gamma,\Delta)$ solve the problem

$$\begin{array}{ll} \underset{y}{\text{minimize}} & v(\hat{y})+p^T(y-\hat{y})-q_i^T(y-\hat{y}) \\ \text{subject to} & q_i^T(y-\hat{y})\geq 0 \\ & y\in\mathcal{Y}_G \end{array} \tag{5.75}$$

and denote the optimum by ψ_i^- and the optimal y by $\hat{y}_i^+$.

As with the x bounded in max norm we can define a complete lower bounding function over $\mathcal{Y}_G$, here denoted ψ. It gets the form

$$\psi(y)\stackrel{\Delta}{=}v(\hat{y})+p^T(y-\hat{y})-\|Q(y-\hat{y})\|_\infty \tag{5.76}$$

Again it can be shown, that $\psi(y)$ for x bounded in 1-norm, has almost the same properties, as ϕ for x bounded in max-norm:

$$\psi\begin{cases} \text{is piece-wise linear} \\ \text{is concave} \\ \text{is continuous} \\ \text{touches } v \text{ at } \hat{y}, \text{ i.e. } \phi(\hat{y})=v(\hat{y}) \end{cases} \tag{5.77}$$

The only property that is lacking in comparison with ϕ is the partition of the linear parts by Q_s. Even though ϕ also gets a pyramid shape, the edges of the pyramid are not aligned with $q_i(y-\hat{y})=0$, but with the lines given by

$$\|Q(y-\hat{y})\|_\infty=\left|q_j^T(y-\hat{y})\right|=\left|q_i^T(y-\hat{y})\right|, i\neq j, i,j=1,\cdots,m_x \ . \tag{5.78}$$

We examine this by an example:

Example 5.6 (Simple ψ functions) *Suppose $m_y=2$, $\hat{y}=[0\ 0]^T$, $p=[0\ 0]^T$, x bounded in the unit cube in 1 norm, and $Q=\begin{bmatrix}1 & 0\\ 0 & 2\end{bmatrix}$, then*

$$\phi(y)=\min_{y_1,y_2}(-|1y_1|,-|2y_2|) \ .$$

A plot can be seen in figure 5.7. The corners in the pyramid is aligned with

$$|1y_1|=|2y_2|=\min_{y_1,y_2}(-|1y_1|,-|2y_2|)$$

which constitutes to the lines $y_1=2y_2$ and $y_1=-2y_2$.

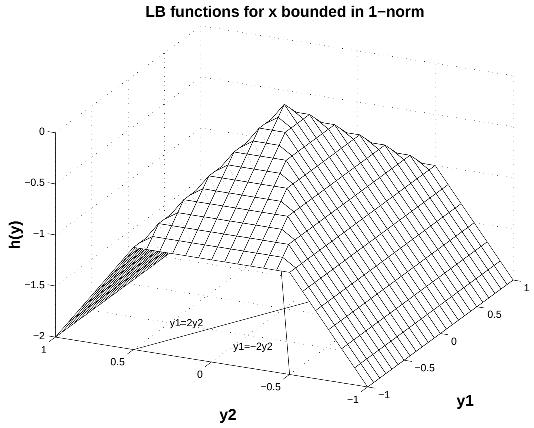


Figure 5.7: A simple ψ function

Partition

In this case the partition should not be defined by the lines $q_i(y-\hat{y}) = 0$, since we then get 2^{m_x} new regions. Instead we will give two suggestions for splitting $\mathcal{Y}_G$ in two or four. The suggestions are heuristic.

The first partition rule is based on the wish to improve the lowest bound fastest: that is minimize the biggest difference between $v(\hat{y})$ and ψ^+/ψ^-.

The second partition rule is based on the corners of the pyramid. Recall the condition (5.78) then we have the condition $|q_i(y-\hat{y})| = |q_i(y-\hat{y})|$. The bounds on y has not form of a rectangle, so we will assume that y is bounded in a circle, that is the 2-norm. We can now use Cauchy-Schwartz inequality to get $|q_i(y-\hat{y})| = \left|\langle q_i^T, (y-\hat{y})\rangle\right| \leq \|q_i\|_2 \|y-\hat{y}\|_2$. Therefore we conject that the two columns q_i and q_j with largest 2-norm constitutes to the smallest lower bound, and splitting according to these will leave us with the best partition. This partition should now be performed by $q_i(y-\hat{y}) = \pm q_i(y-\hat{y})$, which will partition the region in four regions.

We summarize these partition rules in the following list:

1. Let $\tilde{j}$ be one of the entries, which have minimal lower bound:

$$\tilde{j} \in \arg\min_{i\in\mathcal{A}} \min\{\psi_i^+, \psi_i^-\} \tag{5.79}$$

   then use $q_j(y-\hat{y}) = 0$ to split the region in two.

2. Let i, j be two entries with largest Euclidean norm of q_j, then partition using $(q_i - q_j)(y-\hat{y}) = 0$ and $(q_i + q_j)(y-\hat{y}) = 0$. If there is only one entry q_i split using $q_j(y-\hat{y}) = 0$.

No inheritance

Due to the form of the pyramid inheritance is difficult, and time consuming and therefore neglected

Splitting point

The candidate for a splitting point available from the lower bounding subproblem is not really adequate, since it most likely will lye on the boundary. Therefore, a point in the center should be determined using either (5.72) or (5.73).

5.7 Customized Branch and Bound algorithms

In the above sections we have discussed how we could obtain upper and lower bound, which could be used together with the Branch and Bound algorithm. Even though the BB algorithm could be used directly the BB algorithm should be customized to exploit special features of the upper and lower bounds.

The upper and lower bounds are denoted differently for each proposed algorithm. For the bilinear connection approach, χ is used, for the socalled pyramid algorithm ϕ denotes the bounds, where as ψ is used for the socalled house algorithm.

5.7.1 Relaxation via bilinear connection

We will now give a more extensive description of an algorithm, that can exploit the relaxation of the bilinear connection. The lower bound over a region could be computed as an SDP problem, and the solution to this SDP problem also gave directions for the splitting and a could point to use in the restricted problem (upper bound).

The algorithm graduately builds a branch and bound tree, where each node in the tree corresponds to a region, in the original set $\mathcal{Y}_G$. We will refer to each node by the corresponding region. Each node gets superscript to reflect the iteration it is has been split in, such that $Q^{(l)}$, has been split in l'th iteration. Subscript $s = 1,2$ are indices for the two regions that was split in a iteration.

$\overline{f}$ and $\underline{f}$ are used to denote global upper and lower bound respectively. A tilde denotes that the result was obtained from the relaxation problem, where as a $\hat{y}$ denotes the center point, and $\hat{x}$ is the optimal point in the restricted problem solved at $\hat{y}$.

We call the algorithm for the Relaxed Bilinear Connection Branch and Bound algorithm, abbreviated RBCBB algorithm.

Relaxed Bilinear Connection Branch and bound algorithm

1. **Initialization:**

   Set $k = 1$.

   Compute $\mathcal{R}_x \supset \mathcal{X}_G$ and $\mathcal{R}_y \supset \mathcal{Y}_G$ and let $\mathcal{R}_y^{(1)} \triangleq Q^{(1)} = \mathcal{R}_y$.

   Let $Q^{(1)}$ be the root in the BB tree.

   Set the splitting point to $\tilde{x}^{(1)} \triangleq (l_x + u_x)/2$, $\tilde{y}^{(1)} \triangleq (l_y + u_y)/2$, and $\tilde{W}_{ij}^{(1)} = \tilde{x}_i^{(1)} \tilde{y}_j^{(1)}$, $i = 1, \cdots, m_x$, $j = 1, \cdots, m_y$.

2. **Branching and local bounding:**

   (a) **Center point**

   Find indices $(\tilde{\imath}, \tilde{\jmath}) \in \arg\max_{i,j} \left\{ \left| \tilde{w}_{ij}^{(k)} - \tilde{x}_i^{(k)} \tilde{y}_j^{(k)} \right| : \tilde{x}_i \neq 0, i = 1, \cdots, m_x, j = 1, \cdots, m_y \right\}$ and split $\mathcal{R}_y^{(k)}$ in two by the plane $\hat{y}_{\tilde{\jmath}} = \frac{1}{2}\left(\tilde{y}_{\tilde{\jmath}} + \frac{\tilde{w}_{\tilde{\imath}\tilde{\jmath}}}{\tilde{x}_{\tilde{\jmath}}}\right)$

   (b) **Upper bounding**

Solve the restricted problem

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f(x,\hat{y}^{(k)}) \\ \text{subject to} & F(x,\hat{y}^{(k)}) \geq 0 \\ & G(x,\hat{y}^{(k)}) \geq 0 \end{array}$$

Optimum is the upper bound $\overline{\chi}^{(k)}$, and the optimal variable $\hat{x}^{(k)}$.

(c) **Branching**

Set $Q_1^{(k)} \triangleq Q^{(k)} \cap \{y : y \leq \hat{y}_{\tilde{i}}\}$ and $Q_2^{(k)} \triangleq Q^{(k)} \cap \{y : y \geq \hat{y}_{\tilde{i}}\}$ in the BB tree as children to $Q^{(k)}$.

(d) **Lower bounding** Generate

$$\mathcal{W}_1^{(k)}(x,y) = \mathcal{W}_{\mathcal{R}_x,Q_1^{(k)}}(x,y)\,\mathcal{W}_2^{(k)}(x,y) = \mathcal{W}_{\mathcal{R}_x,Q_1^{(k)}}(x,y)$$

from $\mathcal{R}_x$ and $Q_1^{(k)}/Q_2^{(k)}$ and solve the two relaxed problems:

$$\begin{array}{ll} \underset{x,y,W}{\text{minimize}} & f(x,y) \\ \text{subject to} & \tilde{F}(x,y,W) \geq 0 \\ & G(x,y) \geq 0 \\ & y \in Q_1^{(k)} \\ & W \in \mathcal{W}_1^{(k)}(x,y) \end{array} \qquad \begin{array}{ll} \underset{x,y,W}{\text{minimize}} & f(x,y) \\ \text{subject to} & \tilde{F}(x,y,W) \geq 0 \\ & G(x,y) \geq 0 \\ & y \in Q_1^{(k)} \\ & W \in \mathcal{W}_2^{(k)}(x,y) \end{array}$$

Denote the optima by $\underline{\chi}_1^{(k)}/\underline{\chi}_2^{(k)}$ respectively, and let $\tilde{x}_1^{(k)}, \tilde{y}_1^{(k)}, \tilde{W}_1^{(k)}/\ \tilde{x}_2^{(k)}, \tilde{y}_2^{(k)}, \tilde{W}_2^{(k)}$ be the optimal variables

3. **Global bounding:**

(a) Compute the index for the best global upper bound $l = \arg\min_{j=1,\cdots,k} \overline{\chi}^{(j)}$ and let $x^* = \hat{x}^l, y^* = \hat{y}^l$, and the global bound is $\overline{f}^{(k)} = f(x^*,y^*)$.

(b) Compute the best global lower bound $\underline{f}^{(k)}$, as

$$\underline{f}^{(k)} = \min_{j=1,\cdots,k} \min_{\substack{s=1,2 \\ Q_s^{(l)} \text{ leave}}} \underline{\chi}_s^{(l)}$$

and let p,r belong be one of the sets achieving $\underline{f} = \underline{f}_r^{(p)}$, where $Q_r^{(p)}$ is a leave.

4. **Conditioning:** If $\overline{f}^{(k)} - \underline{f}^{(k)} \leq \varepsilon$ then x^*, y^* are ε optimal, EXIT.

5. **Selection:** Update $k \triangleq k+1$, and let $Q^{(k)} \triangleq Q_q^{(p)}$, $x^{(k)} \triangleq \tilde{x}_q^{(p)}$, $y^{(k)} \triangleq \tilde{y}_q^{(p)}$, $W^{(k)} \triangleq \tilde{W}_q^{(p)}$.

6. Goto 2.

The above algorithm is described such that aspects important for computation are considered. The description is not adequate for direct implementation in for instance MATLAB , C++ or Java++. These include what is called *pruning*, that is removing regions where the lower bound on a region is above the global upper bound. We discuss these implementation issues in appendix.

These implementation related matters do not affect the number of iterations. However, tricks can be performed. For instance after some iterations k we know, that the optimum will lye between $\underline{f}^{(k)}$ and $\overline{f}^{(k)}$ so we can add an constraint like

$$\underline{f}^{(k)} \leq f(x,y) \leq \overline{f}^{(k)} \tag{5.80}$$

to the two relaxed problems. As discussed earlier this will improve the lower bounds, since the region over which relaxed problem is solved is compressed, and due to this reduce the required number of iterations.

The constraint (5.80) could also be used to compress the rectangle $\mathcal{R}_x$, if $c \neq 0$, and in this way improve the lower bound. Recall that a calculation of $\mathcal{R}_x$ requires $2m_x$ SDP problems. Although these SDP problems are simple, the problems might require more computational time, than the amount that is saved due to the better bounds. This drawback could be solved by only recalculating $\mathcal{R}_x$, at certain iterations (every tenth) or when the upper and lower bounds have changed significantly. An improvement can also be obtained by compressing the regions $Q^{(k)}$ before steb 2.b. The drawback here is even stronger, since the calculations can only be used for the children of that region, in oppose to $\mathcal{R}_x$ that works on all regions.

5.7.2 Relaxation via Lagrangian duality

In this subsection we present two algorithms based on the relaxation using Lagrangian duality presented above. We first show an algorithm that is customized to exploit the lower bounds obtained when x is bounded in max norm. We then proceed with the customized algorithm when x is bounded in 1 norm.

Lagrangian Pyramid Branch and bound algorithm

When x is bounded the lower bounding function $\phi(y)$ we could derive, have a shape like a generalized pyramid. We will therefore call the algorithm for the Lagrangian Pyramid Branch and Bound algorithm, abbreviated LPBB algorithm.

Pyramid Branch and bound algorithm

1. **Initialization:**

   Set $k = 1$.

   Compute $\mathcal{R}_x \supset \mathcal{X}_G$ and let $\mathcal{R}_y^{(1)} \triangleq Q^{(1)} = \mathcal{R}_y$.

   Let $Q^{(1)}$ be the root in the BB tree.

2. **Branching and local bounding:**

   (a) **Upper bounding**
   Solve the restricted problem

   $$\begin{array}{ll} \underset{x}{\text{minimize}} & f(x, \hat{y}^{(k)}) \\ \text{subject to} & F(x, \hat{y}^{(k)}) \geq 0 \\ & G(x, \hat{y}^{(k)}) \geq 0 \end{array}$$

   Optimum is the upper bound $\overline{\phi}^{(k)}$, and the optimal variable $\hat{x}^{(k)}$, and the dual optimal variables $\Gamma^{(k)}, \Delta^{(k)}$.

   (b) **Key constants and active variables**
   Calculate

   $$p_j^{(k)} \triangleq d_j - \mathrm{Tr}\,\hat{\Gamma} F_j^y + \mathrm{Tr}\,\hat{\Delta} G_j^y \ ,$$
   $$\left\{Q^{(k)}\right\}_{ji} \triangleq \mathrm{Tr}\,\Gamma F_{ij}^{xy} \ .$$

where each row in Q is denoted q_i.

Let $\mathcal{A}^{(k)} \triangleq \{i:\ q_i \neq 0\}$ be the set of indices for the active x-variables.

(c) **Inheritance**

$Q^{(k)}$ is just a subregion of all its parents, and the computed lower bounding functions from the parents, can be used to refine the information over the current region. Trace the BB tree for the parents. Let I denote the set of iterations numbers corresponding to the parents of $Q^{(k)}$. Note that $Q^{(k)} = \bigcap_{l\in I} \mathcal{B}^{(l)}$.

(d) **Bounding of the x variables**

Find the optimal interval $[l_i^{(k)}, u_i^{(k)}]$ such that there exists x, y with $G(x,y) \geq 0$ and $\underline{\phi} \leq f(x,y) \leq \overline{\phi}$. This is done by solving

$$\begin{array}{lll} \underset{x,y}{\text{Minimize}} & x_i & \\ \text{subject to} & \underline{f}^{(k-1)} \leq f(x,y) \leq \overline{\phi}^{(k)} & \\ & G(x,y) \geq 0 & \\ & y \in \mathcal{B}^{(l)} & \} \, l \in I \end{array} \qquad \begin{array}{lll} \underset{x,y}{\text{Minimize}} & -x_i & \\ \text{subject to} & \underline{f}^{(k-1)} \leq f(x,y) \leq \overline{\phi}^{(k)} & \\ & G(x,y) \geq 0 & \\ & y \in \mathcal{B}^{(l)} & \} \, l \in I \end{array}$$

for each $i \in \mathcal{A}$. Denote the solution of the left side by $l_i^{(k)}$,and the right side by $u_i^{(k)}$

(e) **Lower bounding**

For each

$$s \in \left\{ s:\ s \in \{-1,1\}^{m_x},\ s_i = 1 \text{ for } i \notin \mathcal{A}^{(k)} \right\} \triangleq \mathcal{T}^{(k)}$$

generate

$$\mathcal{B}_s^{(k)} = \left\{ y:\ y \in \mathbb{R}^{m_y}, s_i q_i^{(k)}(y - \hat{y}^{(k)}) \geq 0, i \in \mathcal{A}^{(k)} \right\}$$

$$h_s^{(k)}(y) = \overline{\phi}^{(k)} + p^{(k)}(y - \hat{y}^{(k)}) + \sum_{\substack{i=1 \\ i\in\mathcal{A}^{(k)}}} \left(\frac{u_i^{(k)} + l_i^{(k)}}{2} + s_i \frac{u_i^{(k)} - l_i^{(k)}}{2} \right) q_i(y - \hat{y}^{(k)})$$

and solve the subproblem

$$\begin{array}{lll} \underset{x,y}{\text{minimize}} & \phi & \\ \text{subject to} & G(x,y) \geq 0 & \\ & y \in \mathcal{B}_s^{(k)} & \\ & \phi \geq h_s^{(k)}(y) & \\ & y \in \mathcal{B}^{(l)} & \multirow{2}{*}{$\Big\} \, l \in I$} \\ & \phi \geq h^{(l)}(y) & \end{array}$$

If the above problem is feasible denote the optimum by $\underline{\phi}_s^{(k)}$, the optimal y as $\hat{y}_s^{(k)}$, put $Q_s^{(k)} = Q^{(k)} \cap \mathcal{B}_s^{(k)}$ in the BB tree.

3. **Global bounding:**

(a) Compute the index for the best global upper bound $l = \arg\min_{j=1,\cdots,k} \overline{\phi}^{(j)}$ and let $x^* = \hat{x}^l, y^* = \hat{y}^l$, and the global bound is $\overline{f}^{(k)} = f(x^*, y^*)$.

(b) Compute the best global lower bound $\underline{f}^{(k)}$, as

$$\underline{f}^{(k)} = \min_{j=1,\cdots,k} \ \min_{\substack{s\in\mathcal{T}^{(l)} \\ Q_s^{(l)} \text{ leave}}} \underline{\phi}_s^{(l)}$$

and let p, r belong be one of the sets achieving $\underline{f} = \underline{\psi}_r^{(p)}$, where $Q_r^{(p)}$ is a leave.

4. **Conditioning:** If $\overline{f}^{(k)} - \underline{f}^{(k)} \le \varepsilon$ then x^*, y^* are ε optimal, EXIT.

5. **Selection:** Update $k \triangleq k+1$, and let $Q^{(k)} \triangleq Q_r^{(p)}$, $\mathcal{B}^{(k)} \triangleq \mathcal{B}_r^{(p)}$, $y^{(k)} \triangleq \tilde{y}_r^{(l)}$, $h^{(k)} = h_r^{(p)}$.

6. Goto 2.

Lagrangian House Branch and bound algorithm

When x is bounded in 1 norm, we have to solve $2m_x$ subproblems. For each variable x_i we get a house like lower bound, which combined gave a pyramid shape. However to distinguish this approach with the pyramid algorithm above, we will call this algorithm the Lagrangian House Branch and Bound (LHBB) algorithm.

Lagrangian House Branch and bound algorithm

1. **Initialization:**

   Set $k = 1$.

   Find some Λ and $\hat{x}$, such that $\mathcal{R}_1^x = \{x : \ ||\Lambda(x-\hat{x})||_1 \le 1\} \supset \mathcal{X}_G$.

   Let $Q^{(1)}$ be the root in the BB tree.

2. **Branching and local bounding:**

   (a) $Q^{(k)}$ is just a subregion of all its parents, and the computed lower bounding functions from the parents, can be used to refine the information over the current region. Trace the BB tree for the parents. Let I be the set of iterations numbers corresponding to the parents of $Q^{(k)}$. Let $q^{(l)}(y-\hat{y}^{(l)}) = 0, l \in I$ be the cutting planes with appropriate sign such that

   $$Q^{(k)} = \left\{ y : \ \exists x, G(x,y) \ge 0, q^{(l)}(y-\hat{y}^{(l)}) \ge 0, l \in I \right\}$$

   Let $H(x,y)$ be the LMI constraint consisting of $G(x,y) \ge 0$ and $q^{(l)}(y-\hat{y}^{(l)}) \ge 0, l \in I$.

   (b) Find the analytical center of $Q^{(k)}$, that is solve the problem

   $$\begin{array}{ll} \underset{x,y}{\text{minimize}} & \log\det H(x,y)^{-1} \\ \text{subject to} & H(x,y) \ge 0 \end{array}$$

   and denote the optimal variable by $y^{(k)}$.

   (c) **Upper bounding**

   Solve the restricted problem

   $$\begin{array}{ll} \underset{x}{\text{minimize}} & f(x,\hat{y}^{(k)}) \\ \text{subject to} & F(x,\hat{y}^{(k)}) \ge 0 \\ & G(x,\hat{y}^{(k)}) \ge 0 \end{array}$$

   Optimum is the upper bound $\overline{\Psi}^{(k)}$, and the optimal variable $\hat{x}^{(k)}$, and the dual optimal variables $\Gamma^{(k)}, \Delta^{(k)}$.

   (d) **Key constants and active variables**

   Calculate

   $$p_j^{(k)} \triangleq d_j - \mathrm{Tr}\,\hat{\Gamma} F_j^y + \mathrm{Tr}\,\hat{\Delta} G_j^y \ ,$$
   $$\left\{Q^{(k)}\right\}_{ji} \triangleq \mathrm{Tr}\,\Gamma F_{ij}^{xy} \ .$$

   where each row in Q is denoted q_i.

   Let $\mathcal{A}^{(k)} \triangleq= \{i : \ q_i \ne 0\}$ be the set of indices for the active x-variables.

(e) **Lower bounding**
For each index i in $\mathcal{A}^{(k)}$ solve the two problems

$$\begin{array}{ll} \underset{x,y}{\text{minimize}}\ \overline{\Psi}^{(k)} + p^{(k)}(y-\hat{y}^{(k)}) - q_i(y-\hat{y}^{(k)}) & \underset{x,y}{\text{minimize}}\ \overline{\Psi}^{(k)} + p^{(k)}(y-\hat{y}^{(k)}) + q_i(y-\hat{y}^{(k)}) \\ \text{subject to } G(x,y) \geq 0 & \text{subject to } G(x,y) \geq 0 \\ \qquad\qquad +q_i(y-\hat{y}^{(k)}) \geq 0 & \qquad\qquad -q_i(y-\hat{y}^{(k)}) \geq 0 \\ \qquad\qquad q^{(l)}(y-\hat{y}^{(l)}) \geq 0,\ l \in I & \qquad\qquad q^{(l)}(y-\hat{y}^{(l)}) \geq 0,\ l \in I \end{array}$$

Denote the optimum of the left problem by $\underline{\psi}_i^{+}$ and the right by $\underline{\psi}_i^{-}$.

(f) **partition** Compute

$$\tilde{j} = \arg\min_{i\in\mathcal{A}} \left[\min\{\underline{\psi}_i^{+}, \underline{\psi}_i^{-}\}\right] \tag{5.81}$$

partition with the plane $q_{\tilde{j}}(y-\hat{y}^{(k)}) = 0$. Define $q^{(k)} = q_{\tilde{j}}$. Insert the two regions $Q_1 = Q^{(k)} \cap \left\{y:\ q_j(y-\hat{y}^{(k)}) \leq 0\right\}$, and $Q = Q_2^{(k)} \cap \left\{y:\ q_j(y-\hat{y}^{(k)}) \leq 0\right\}$. As lower bound $\underline{\psi}_1^{(k)}/\underline{\psi}_2^{(k)}$ to $Q_1^{(k)}/Q_2^{(k)}$ is

$$\underline{\psi}_1^{(k)} = \min\left[\underline{\psi}_i^{+}, (\min_{\substack{i\in\mathcal{A}\\ i\neq\tilde{j}}}\{\underline{\psi}_i^{+}, \underline{\psi}_i^{-}\}), \underline{\psi}^{(k)}\right]$$

$$\underline{\psi}_2^{(k)} = \min\left[\underline{\psi}_i^{-}, (\min_{\substack{i\in\mathcal{A}\\ i\neq\tilde{j}}}\{\underline{\psi}_i^{+}, \underline{\psi}_i^{-}\}), \underline{\psi}^{(k)}\right]$$

3. **Global bounding:**

   (a) Compute the index for the best global upper bound $l = \arg\min_{j=1,\cdots,k} \overline{\Psi}^{(j)}$ and let $x^* = \hat{x}^{(l)}, y^* = \hat{y}^{(l)}$, and the global bound is $\overline{f}^{(k)} = f(x^*, y^*)$.

   (b) Compute the best global lower bound $\underline{f}^{(k)}$, as

$$\underline{f}^{(k)} = \min_{l=1,\cdots,k}\ \min_{\substack{s\in\{1,2\}\\ Q_s^{(l)}\ \text{leave}}} \underline{\psi}_s^{(l)}$$

and let p, q belong be one of the sets achieving $\underline{f}^{(k)} = \underline{\psi}_r^{(p)}$, where $Q_r^{(p)}$ is a leave.

4. **Conditioning:** If $\overline{f}^{(k)} - \underline{f}^{(k)} \leq \varepsilon$ then x^*, y^* are ε optimal, EXIT.

5. **Selection:** Update $k \triangleq k+1$, and let $Q^{(k)} \triangleq Q_q^{(r)}$, $q^{(k)} \triangleq q_r^{(p)}$, $y^{(k)} \triangleq \tilde{y}_q^{(r)}$, $h^{(k)} = h_s^{(p)}$, and $\underline{\psi}^{(k)} = \underline{\psi}_r^{(p)}$.

6. Goto 2.

5.8 Convergence properties

XXX needs to be rewritten to be more stringent..

Convergence of the above algorithms is of course an interesting issue. However more important is that if the algorithm converges then the solution is indeed the global optimum. In fact most often a specific time is available for solving a given problem, this could be minutes, hours, days, or even weeks. If the problem has not been solved to the required accuracy within the available time, then we have two possibilities. Either we use the solution, provided by the (so far) optimal upper bound, or we accept that no solution has been found.

With this in mind we proceed with a discussion of the convergence properties of the above mentioned algorithms. All algorithms is based on a branch and bound frame work. We have the following sufficient conditions for proof of convergence (see [HJ95]):

H1 As the diameter of the minimal rectangle $\mathcal{R} \supset \mathcal{Q} \to 0$ then $\overline{f}_{\mathcal{Q}} - \min_{i=1,\cdots,p_{\mathcal{Q}}} \underline{f}_{\mathcal{Q}_i} \to 0$.

H2 The diameter of the minimal rectangle $\mathcal{R} \supset \mathcal{Q}$ must go to zero when the number of parents go to infinity.

H3 The subproblem with the lowest lower bound must be selected in at least every K'th iteration.

We will now go through these conditions for each algorithm given above.

5.8.1 Relaxation of bilinear connection

For the RBCBB algorithm the first condition H1 is fulfilled, because when the diameter $||u^y - l^y||$ of $\mathcal{Q} = \{y : u^y - l^y\} \supset \mathcal{Q}$ goes to zero, then the bounds on W becomes more and more tight, and thus forcing $w_{ij} = x_i y_j$ yielding that the solution to the lower bound must be equal to the upper bound. The partition by (5.24) guarantees that the region $\mathcal{Q}$ is split in such away that the constraints on the relaxed variables w_{ij} that are furthest from $x_i y_j$ are tighten. Exactly beacuse the maximizer of $\left|w_{ij} - x_i y_j\right|$ is used to split the y-space. Condition H3 is fulfilled by step 3.b and 5 in the description.

5.8.2 Lagrangian relaxation

H1 Condition H1 is fulfilled by the tightness of the ϕ and ψ functions. Both functions touches $v(y)$ at $\hat{y}$, and thus when the diameter of the minimal superior rectanlge $\mathcal{R}$ for $\mathcal{Q}$ goes to zero the difference between the upper and lower bounds goes to zero.

H2 The second conditon H2 is for the Lagrangian House algorithm easily fulfilled by step 2.b and 2f.

For the Lagrangian Pyramid algorithm the splitting point $\hat{y}$ is given from the solution of the subproblem. This solution $\hat{y}$ might lye on the boundary of the set $\mathcal{Q}$, thus the partition given by the planes $q_i(y - \hat{y}) = 0, i \in \mathcal{A}$ might not partition $\mathcal{Q}$ at all, see figure 5.8. However, at each iteration the tightness of the LB function ϕ guarantees, that around $\hat{y}$ in a certain area the optimal value function has been approximated by the required amount ε. Or more precisely there exist a region (convex in fact) Ω around $\hat{y}$, such that

$$\Omega = \{y : v(y) - \phi(y) \leq \varepsilon\} \quad ,$$

where ε is the required accuracy. In a region $\mathcal{Q}^{(k)}$, that is a child of the regions $\mathcal{Q}^{(l)}, l \in I$, there exists a number of regions $\Omega^{(l)}$, where the optimal function $v(y)$ has been approximately sufficiently. However, it remains to be shown that with a limited amount of iterations I, we can get

$$\mathcal{Q}^{(k)} \subset \bigcup_{l \in I} \Omega^{(l)} \quad . \tag{5.82}$$

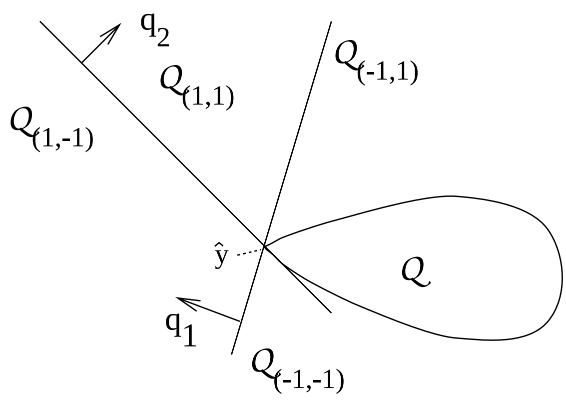


Figure 5.8: A partition on the boundary of Q.

Floudas and Visweswaran [FV93] talk about "accumulation of constraints", and tend to prove that the algorithm converges using this. But it is the authors opion, that the proof is not complete. Even though the union of $\Omega^{(k)}$ and $\bigcup_{l\in I}\Omega^{(l)}$ might be slightly bigger than $\bigcup_{l\in I}\Omega^{(l)}$, there is no guarantee that we have uniform convergence. In other words the sequence

$$\Theta^{(k)} = \Omega^{(k)} \backslash \bigcup_{l\in I}\Omega^{(l)} \quad , \tag{5.83}$$

might approach the $\emptyset$. There might be a BMI problem, where the sequence of partitions generates, smaller and smaller regions, but at the same time generates more and more new regions, that have to be split in a similar fashion. The problem with this can be seen in example 5.9, where a huge amount of regions, are generated, and the improvement of the lower bound is negligible. The author stress though, that this is not a contradiction to convergence of the algorithm. The problem with the partition can be alleviated, by using the techniques given in (5.72) or (5.73).

H3 For both cases x bounded in 1 and max-norm, condition H3 is trivially fulfilled, by step 3.b and step 5 in both algorithms.

5.9 Solving Control problems with Global Optimization

We conclude this chapter by examine how to formulate control problems such that they can be solved using the above proposed algorithms. We shall see that is a lot of structure in the control problems, can be used to reduce the computational time. First of all reducing volume of the set $\mathcal{Y}_G$ implies that the lower bound will be better.

Let us first consider the BMI formulation of the fixed-order optimal α stabilizing control problem. It was stated as the BMI problem (3.4). We repeat it here for better readability.

$$\begin{array}{ll} \underset{Y,G,\alpha}{\text{Maximize}} & \alpha \\ \text{subject to} & Y > 0,\ G \in \mathbb{R}^{(n_c+n_y)\times(n_c+n_u)} \\ & \left(\tilde{A}+\tilde{B}_u G\tilde{C}_y+\alpha I\right)^T Y + Y\left(\tilde{A}+\tilde{B}_u G\tilde{C}_y+\alpha I\right) < 0. \end{array} \tag{5.84}$$

We assume that the reader knows how to transform the above problem (5.84) to the form (5.2). Otherwise an example is given in example 2.1. We introduce the variables $y = \mathrm{SVec}\,Y$, $g = \mathrm{Vec}\,G$. There are three variables y, g, α, where we will combine g and α in the variable x.

We have the following BMI constraint:

$$\left(\tilde{A}+\tilde{B}_u G\tilde{C}_y+\alpha I\right)^T Y + Y\left(\tilde{A}+\tilde{B}_u G\tilde{C}_y+\alpha I\right) < 0$$

which we denote $F(x,y)$ and the following LMI constraint

$$Y > 0$$

There are no constraints on either g or α, and $\mathrm{SVec}^{-1}y$ is only constraint to the positive definite cone, that is unbounded. We therefore have to add constraints on g, α, and y to ensure that assumption 5.1 hold.

We will add the two constraints $\mathrm{Tr}Y \leq n(1+\varepsilon), \mathrm{Tr}Y \geq n(1-\varepsilon)$ to Y. This does not introduce any conservatism, since if Y solves the BMI then so does $\lambda Y, \lambda > 0$.

However, we can add more constraints on Y. Suppose we know that the optimal α is bigger than $\overline{\alpha}$, then we can restrict our attention to the set of Y's, that corresponds to the existence of a full order controller with stability degree $\overline{\alpha}$. Recall that the existence of G solving the above BMI is guaranteed by $Y^{-1} \in \mathcal{X}_{\overline{\alpha},n_c}$ and $Y \in \mathcal{Y}_{\overline{\alpha},n_c}$. The last condition is convex, so we can add this to the constraints on Y. We can add stronger constraints on Y. Y and Y^{-1} was related by (3.18), here repeated,

$$Y = \begin{bmatrix} \mathsf{Y} & Y_{12} \\ Y_{12}^T & Y_{22} \end{bmatrix} \text{ and } Y^{-1} = \begin{bmatrix} \mathsf{X} & * \\ * & * \end{bmatrix} . \tag{5.85}$$

We can restrict our search in Y to the the set

$$\{\mathsf{Y} : \exists(\mathsf{X},\mathsf{Y}) \in \mathcal{X}_{\overline{\alpha},0} \times \mathcal{Y}_{\overline{\alpha},0} \cap \mathcal{Z}\} \quad ,$$

which guarantees that Y is the upper left corner of a Lyapunov function that proves the existence of a full order controller. Combining the constraints on Y we get

$$\begin{gathered} Y > 0 \\ \mathrm{Tr}Y \leq n(1+\varepsilon) \\ \mathrm{Tr}Y \geq n(1-\varepsilon) \\ Y = \begin{bmatrix} \mathsf{Y} & * \\ * & * \end{bmatrix} \\ \mathsf{C}_y^{T\perp}\left(\mathsf{A}^T\mathsf{Y}+\mathsf{Y}\mathsf{A}+2\overline{\alpha}\mathsf{Y}\right)\mathsf{C}_y^{T\perp T} < 0 \\ \begin{bmatrix} \mathsf{X} & I \\ I & \mathsf{Y} \end{bmatrix} \geq 0 \\ \mathsf{B}_u^{\perp}\left(\mathsf{A}\mathsf{X}+\mathsf{X}\mathsf{A}+2\overline{\alpha}\mathsf{X}\right)\mathsf{B}_u^{\perp T} < 0 \end{gathered} \tag{5.86}$$

The last two constraints requires an introduction of a new variable X. However the variable is not completely bounded we need to add an upper bound, $\mathrm{Tr}\mathsf{X} \leq M$ will do. The coupling constraint is equivalent to $\mathsf{Y} > 0$ and $\mathsf{X} \geq \mathsf{Y}^{-1}$, combining this with $Y > 0$ we get $Y > 0$ and $\mathsf{X} \geq \begin{bmatrix} I & 0 \end{bmatrix} Y \begin{bmatrix} I \\ 0 \end{bmatrix}$, which is equivalent to

$$\begin{bmatrix} \mathsf{X} & \begin{bmatrix} I & 0 \end{bmatrix} \\ \begin{bmatrix} I \\ 0 \end{bmatrix} & Y \end{bmatrix} \geq 0$$

We now write our final suggestion for the constraints on Y, as

$$\begin{gathered} Y = \begin{bmatrix} \mathsf{Y} & * \\ * & * \end{bmatrix} \\ \mathsf{B}_u^{\perp} \left(\mathsf{AX} + \mathsf{XA} + 2\overline{\alpha}\mathsf{X}\right) \mathsf{B}_u^{\perp T} < 0 \\ \mathsf{C}_y^{T\perp} \left(\mathsf{A}^T\mathsf{Y} + \mathsf{YA} + 2\overline{\alpha}\mathsf{Y}\right) \mathsf{C}_y^{T\perp T} < 0 \\ \begin{bmatrix} \mathsf{X} & \begin{bmatrix} I & 0 \end{bmatrix} \\ \begin{bmatrix} I \\ 0 \end{bmatrix} & Y \end{bmatrix} \geq 0 \\ \mathrm{Tr}\, Y \leq n(1+\varepsilon) \\ \mathrm{Tr}\, Y \geq n(1-\varepsilon) \\ \mathrm{Tr}\, \mathsf{X} \leq M \end{gathered} \tag{5.87}$$

This reduces the volume of $\mathcal{Y}_G$ significantly. Especially we only search over the Lyapunov functions that proves the existence of a full order $\overline{\alpha}$ stabilizing controller. $G \in \mathbb{R}^{(n_c+n_y)\times(n_c+n_u)}$

We also need to put constraints on the controller parameter G. The controller parameter G has $(n_c+n_y)\times(n_c+n_u)$ free variables. If $n_c > 1$ this is more than sufficient for parameterizing all possible input/output mappings for the controller. Given a controller

$$\begin{bmatrix} \dot{x}_c \\ u \end{bmatrix} = \underbrace{\begin{bmatrix} \mathsf{A}_c & \mathsf{B}_c \\ \mathsf{C}_c & \mathsf{D}_c \end{bmatrix}}_{G} \begin{bmatrix} x_c \\ y \end{bmatrix}, \tag{5.88}$$

defined by A_c, B_c, C_c, D_c then we want to find a parameterization A such that we can place poles and zeros of the transfer function from u to y freely. A general approach would be to consider the Jordan form of A, but this leads to a complicated parameterization. However, if we restrict our attention to diagonizable A_c then we can get a simple parameterization[2]. B_c and C_c determines the zeros... If n_c is even we consider blocks on the form

$$A_c = \mathrm{Diag}\left(J_1, J_2, \cdots, J_q\right) \tag{5.89}$$

where $q = n_c/2$ and J are two by two blocks on the form

$$J_i = \begin{bmatrix} \alpha_1^i & \omega^i \\ \omega^i & \alpha_2^i \end{bmatrix}$$

Each block give eigenvalues J_i on the form

$$\lambda = \begin{cases} \alpha \pm j\omega & \text{if } \alpha \triangleq \alpha_1 = \alpha_2 \\ \frac{-\alpha_1-\alpha_2\pm\sqrt{(\alpha_1+\alpha_2)^2-4(\alpha_1\alpha_2+\omega^2)}}{2} & \text{otherwise} \end{cases}$$

This allows us to place the poles freely, but not the eigenvectors. With B_c and C_c we can place the zeros. Note that with the form of A_c we have $3q = 3n_c/2$ free parameters instead fo n_c^2.

If n_c is odd, we get a similar parameterization. Let q be the biggest integer, such that $q \leq n_c/2$, then we get

$$A_c = \mathrm{Diag}\left(J_1, J_2, \cdots, J_q, \alpha^{q+1}\right) \tag{5.90}$$

[2]This omit impulse responses on the form $y(t) = \ldots t^q e^{\lambda t} \ldots$

5.10 Experimental results with the Pyramid algorithm

The above pyramid algorithm have been implemented in MATLAB . The SP package by Boyd and Vandenberghe [VB94] was used to solve the SDP problems since it provides the primary as well as the Lagrange variables. SP is coded in the C language and is therefore very fast, where as the bookkeeping code for the algorithm is in plain MATLAB and not specially optimized. The code is does not solve the generalized problem discussed above, but is a more specialized code. The algorithm solves the following problem

$$\begin{array}{ll} \text{minimize} & f(x,y) = c^T x \\ \text{subject to} & F(x,y) \geq 0 \\ & A(y) = A_0 + \sum_{j=1}^{m_y} y_j A_j \geq 0 \\ & -l_i \leq x_i \leq u_i,\ i = 1,\dots,m_x, \end{array} \tag{5.91}$$

The above restriction of the general BMI optimization cannot exploit any convex connections between x and y. At each iteration the superior rectangle $\mathcal{R}$ for

$$\left\{ x : \begin{array}{l} \max[\underline{f},\underline{\phi}] \leq c^T x \leq \max[\overline{f},\overline{\phi}] \\ -l_i \leq x_i \leq u_i,\ i = 1,\dots,m_x \end{array} \right\} \tag{5.92}$$

is found, and used for the lower bound functions, where $\underline{f}$ and $\overline{f}$ are expected upper and lower bounds. This has only effect if $c_i \neq 0$ and $F_{ij}^{xy} \neq 0$, for some i, j.

The algorithm also performs pruning, that is it removes regions with lower bound bigger than upper bound. Also the algorithm finds a feasible point itself before starting the algorithm, so even a slight chance in $A(y)$ might cause a complete different initial y. The code is written to see how well the algorithm performs. Therefore we only give number of iterations and number SDP's.

5.10.1 A classical example

In the literature the following simple BMI has been a bench-mark example.

Example 5.7 (Goh example) *Consider the simple problem from [GSP94]*

$$\begin{array}{ll} \textit{minimize} & t \\ \textit{subject to} & F_0 + xF_1^x + yF_1^y + xyF_{11}^{xy} + tI \geq 0 \\ & -1 \leq t \leq 7 \\ & -3 \leq x \leq 7 \\ & -0.5 \leq y \leq 2 \end{array} \tag{5.93}$$

where

$$F_0 = \begin{bmatrix} -10 & -0.5 & -2 \\ -0.5 & 4.5 & 0 \\ -2 & 0 & 0 \end{bmatrix} \quad F_1^x = \begin{bmatrix} -1.8 & -.1 & -.4 \\ -.1 & 1.2 & -1 \\ -.4 & -1 & 0 \end{bmatrix}$$

$$F_2^y = \begin{bmatrix} 9 & 0.5 & 0 \\ 0.5 & 0 & -3 \\ 0 & -3 & -1 \end{bmatrix} \quad F_{11}^{xy} = \begin{bmatrix} 0 & 0 & 2 \\ 0 & -5.5 & 3 \\ 2 & 3 & 0 \end{bmatrix}$$

The above problem have three local minima, with the optimum as $f^* = -0.9565$ *with an accuracy of* 10^{-4} *and is achieved by* $x^* = 1.049565$ *and* $y^* = 1.416605$.

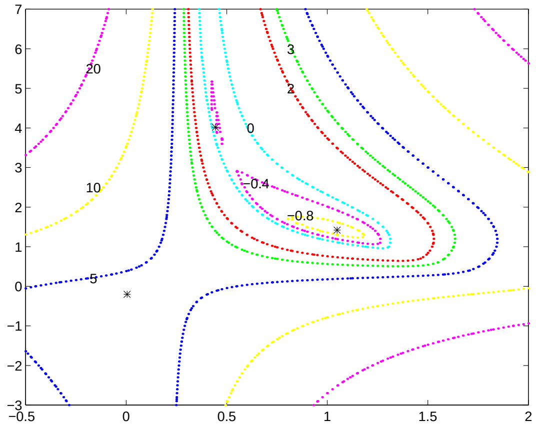


Figure 5.9: Contour plot of the Goh problem

Denote $F(x,y) \triangleq F_0 + xF_1^x + yF_1^y + xyF_{11}^{xy}$*, and consider the maximal eigenvalue of* $F(x,y)$*, as* $h(x,y) = \lambda_{min}(F(x,y))$*. A plot of* $h(x,y)$ *for fixed x and y can be seen in figure 5.9. It should be evident, that* $h(x,y)$ *is not convex. If we project the above problem on x-space, we get the function*

$$v_x(y) = \min_x [h(x,y) \text{ subject to } -3 \leq x \leq 7]$$

and semilarly we can get the projection on y-space as

$$v_y(x) = \min_y [h(x,y) \text{ subject to } -0.5 \leq y \leq 2]$$

These functions are plotted in figure 5.10.

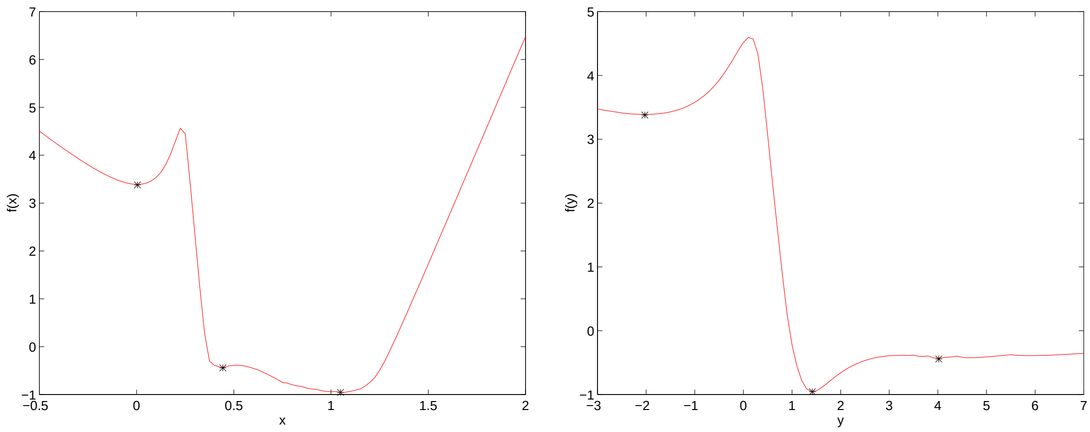


Figure 5.10: Optimal-value functions for the Goh BMI optimization problem
To the left is the minima of the Goh BMI optimization problem for each x, $v_x(y)$, and to the right for each y, $v_y(x)$. Local minima are indicated with a '*'.

The pyramid algorithm solves this problem in 16 iterations to an 10^{-3} *accuracy, and needs 15 iterations more to get an accuracy of* 10^{-4}*. In figure 5.12 is an evolution of the lower bound for*

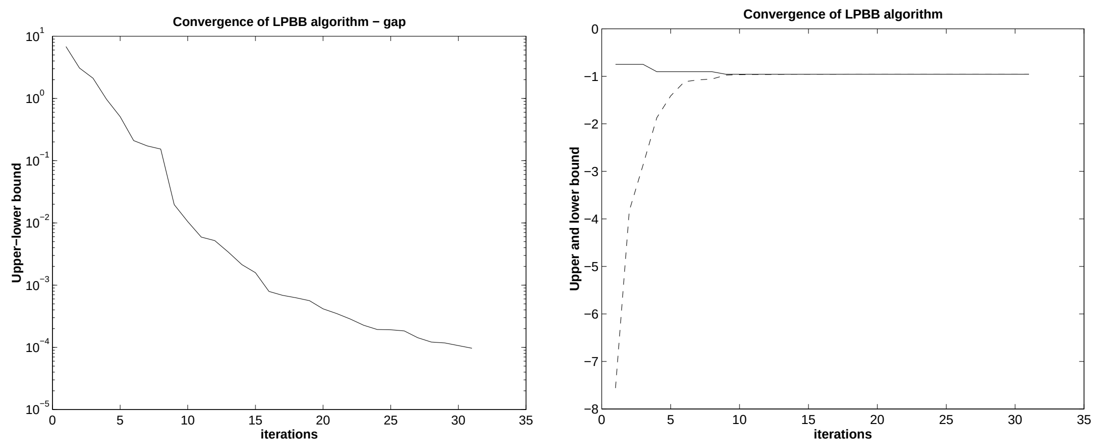


Figure 5.11: Goh example - gap and upper/lower bounds.

all y's over the course of iterations. The upper and lower bounds as a function of the iterations can be seen in figure 5.11, next to a plot of the logarithm of the gap.

The same problem has been solved by Goh et al [GSP95] and by Tuan et al [THT96]. We give a comparison in table 5.1. We only give a comparison in terms of the number of iterations required for obtaining a given accuracy. For all algorithms presented the table the region is split in half in each iteration. The number of iteration therefore give an indication how well the lower bound works.

The number of iterations is for a low accuracy smallest with the 3. method of Tuan et al [THT96], which is equivalent to the RBCBB algorithm presented above. With higher accuracy the number of iterations is smallest for the pyramid algorithms. In all three algorithms three SDP's are solved in each iteration, one for obtaining an upper bound[3], and two SDP's to compute the lower bound over the two new regions. For both Goh, Tuan 3 and Tuan 4 the SDP's are bigger than the pyramid algorithm for the first number of for the lower bound grows with the depth of the Branch and Bound tree for Tuan 4 and Pyramid.

Accuracy	Goh	Tuan 3	Tuan 4	Pyramid
0.5 %	24	7	16	12
0.1 %	NC	25	19	16
0.01 %	NC	NC	NC	31

Table 5.1: Iterations for different algorithms for the Goh example
NC means not calculated. Tuan gives two algorithms.

[3]Goh uses method of centers to improve the upper bound.

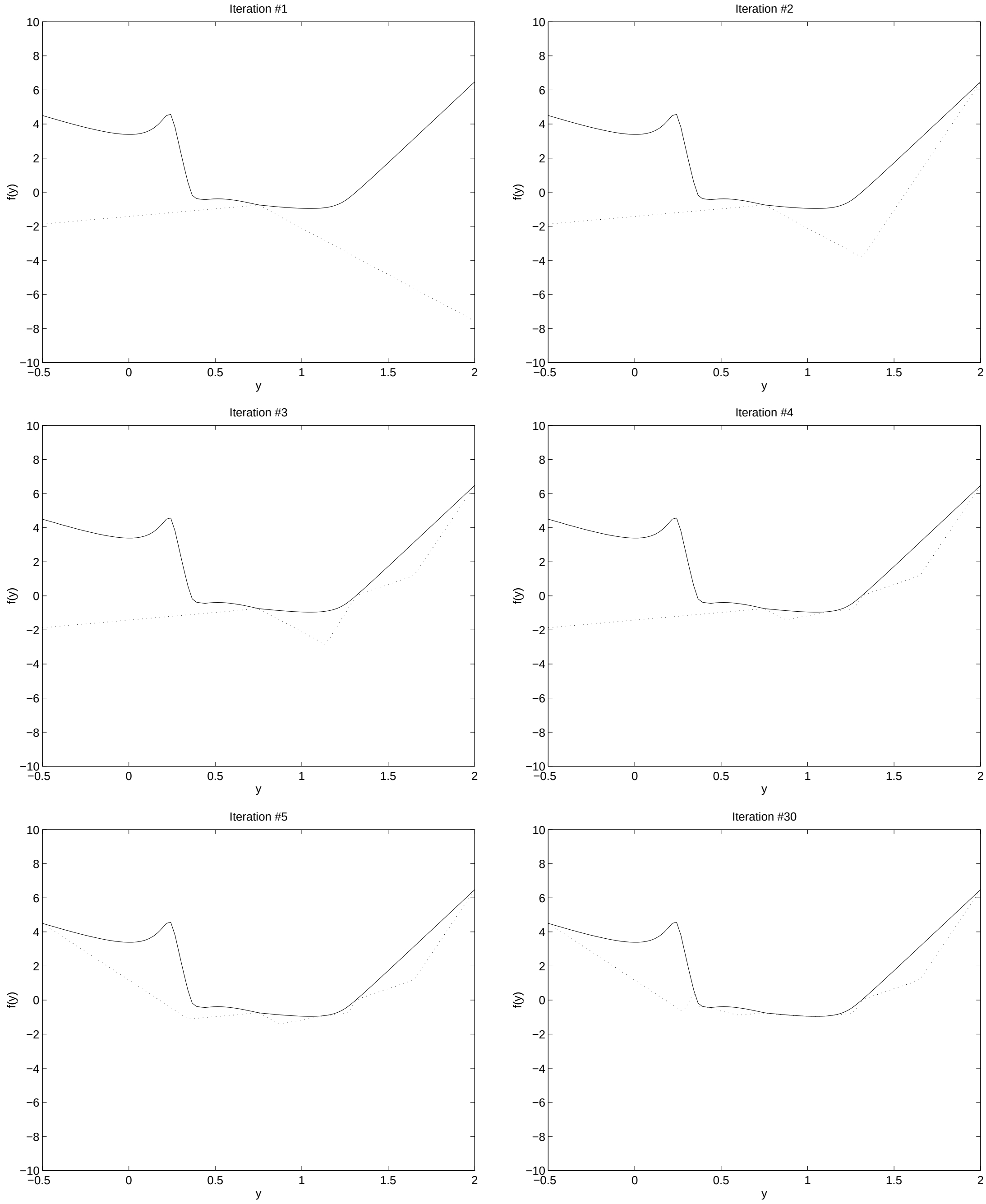


Figure 5.12: Evolution of Φ for the Goh Example
The function $\Phi^{(k)}(y) = \sup_{l=1,\cdots,k} \phi^{\hat{y}^l}(y)$ (dashed) for some iterations and $v(y)$ solid. $\Phi^{(k)}$ slowly approximates the lowest regions of $v(y)$.

5.10.2 Control problems

Example 5.8 (A simple α stabilizing control problem) *We consider the α stability of the following system*

$$\begin{bmatrix} A & B_u \\ C_y & 0 \end{bmatrix} = \left[\begin{array}{cc|c} 0 & 1 & 1 \\ 1 & -1 & 0 \\ \hline 1 & 1 & 0 \end{array}\right]. \tag{5.94}$$

with a zero'th order controller. A root locus plot of the above system is given in figure 4.3. The optimal controller has gain -5 and places bot the closed loop poles at -3. In fact we cannot find a Lyapunov function that proves that $A_{cl}(-5) = A + B_u(-5)C_y$ has an α stability of -3, since $A_{cl}(-5) + 3I$ is singular.

We formulated the problem in the following way:

$$\begin{array}{ll} \underset{\alpha,G,t,Y}{\text{Minimize}} & -\alpha + \rho t \\ \text{subject to} & (\mathsf{A} + \mathsf{B}_u G \mathsf{C}_y + \alpha I)^T Y + Y(\mathsf{A} + \mathsf{B}_u G \mathsf{C}_y + \alpha I) < 0 \\ & \mathsf{C}_y^{T\perp}\left(\mathsf{A}^T Y + Y\mathsf{A} + 2\underline{\alpha} Y\right)\mathsf{C}_y^{T\perp T} < 0 \\ & Y > 1/\kappa I \\ & \operatorname{Tr} Y \leq n(1+\delta) \\ & \operatorname{Tr} Y \geq n(1-\delta) \\ & 0 \leq \alpha \leq 3 \\ & -6 \leq G \leq -1 \\ & 0 \leq t \leq 100 \end{array} \tag{5.95}$$

where $\rho = 100$, $\delta = 1e-3$, and $\overline{\alpha}$ has been varied. κ was chosen to 50 which allows the Lyapunov matrix to have condition number approximately $50n = 100$ in this case. Looking at figure 4.4 the optimal α is expected to be between 2.8 and 3.0. We will solve the problem to accuracy $\varepsilon = 1e-2$. We have solved the above problem for different $\overline{f}/\overline{f}$ and $\overline{\alpha}$ to see the influence on a number of key variables. We will consider the following key variables at each iteration:

1. *The best upper bound $\overline{\phi}^{(k)}$ and best lower bound $\underline{\phi}^{(k)}$.*
2. *The difference between $\overline{\phi}^{(k)} - \underline{\phi}^{(k)}$.*
3. *The optimum of the restricted problem.*
4. *The number of regions.*
5. *Two of the free variables of the Lyapunov matrix, Y_{11} and Y_{12}. ($Y_{22} \approx n - Y_{11}$).*

We will also consider the lower bounds, on the remaining regions.

We consider the following three examples:

$$CP\,I: \begin{array}{l} \underline{\alpha} = 0, \\ \overline{f} = 0, \\ \underline{f} = -3. \end{array} \quad CP\,II: \begin{array}{l} \underline{\alpha} = 2, \\ \overline{f} = 0, \\ \underline{f} = -3. \end{array} \quad CP\,III: \begin{array}{l} \underline{\alpha} = 2.8, \\ \overline{f} = -2.8, \\ \underline{f} = -2.9. \end{array} \tag{5.96}$$

The number of iterations and SDP calls are given in table 5.2 In figure 5.13 the best upper and lower bounds can be seen together with the gap. In figure 5.14 the upper bound for the

	# iter.	# SDP	$\bar{\phi}$	G
CP I	78	378	2.8794	-4.7622
CP II	41	198	2.8803	-4.7618
CP III	47	226	2.8805	-4.7617

Table 5.2: Iterations, SDP calls, upper bound, and optimal G for the simple problem.

restricted problem, can be viewed next to the Lyapunov function and the feasibility region of Y*. In figure 5.15 the number of regions at each iterations is plotted, next to the lower bounds of the regions remaining when the algorithm terminates.*

The algorithm behaves fairly different on the different problems CP I/II/III. The optimum is approximately the same, but the evolution of the algorithm is different. The most open problem, namely CPI, requires the highest number of iterations, which was to be expected. Restricting with $\underline{\alpha} = 2$ *the required iterations were almost halfed. Supprisingly restricting even further, and adding more strict bounds for* $\overline{f}$ *and* $\underline{f}$*, made the algorithm use more iterations.*

XXX discussion.

Example 5.9 (Two wagon example II) *We consider the same example as example 4.2 as in chapter 4. The controller we search for is of order 2. We will limit our search for controllers to the ones having the parameter:*

$$G = \begin{bmatrix} g_1 & g_2 & g_3 \\ -g_2 & -g_4 & -g_5 \\ g_3 & g_5 & g_6 \end{bmatrix}$$

$g_i \in [0;4], i = 1, \cdots, 6.$

XXX discussion

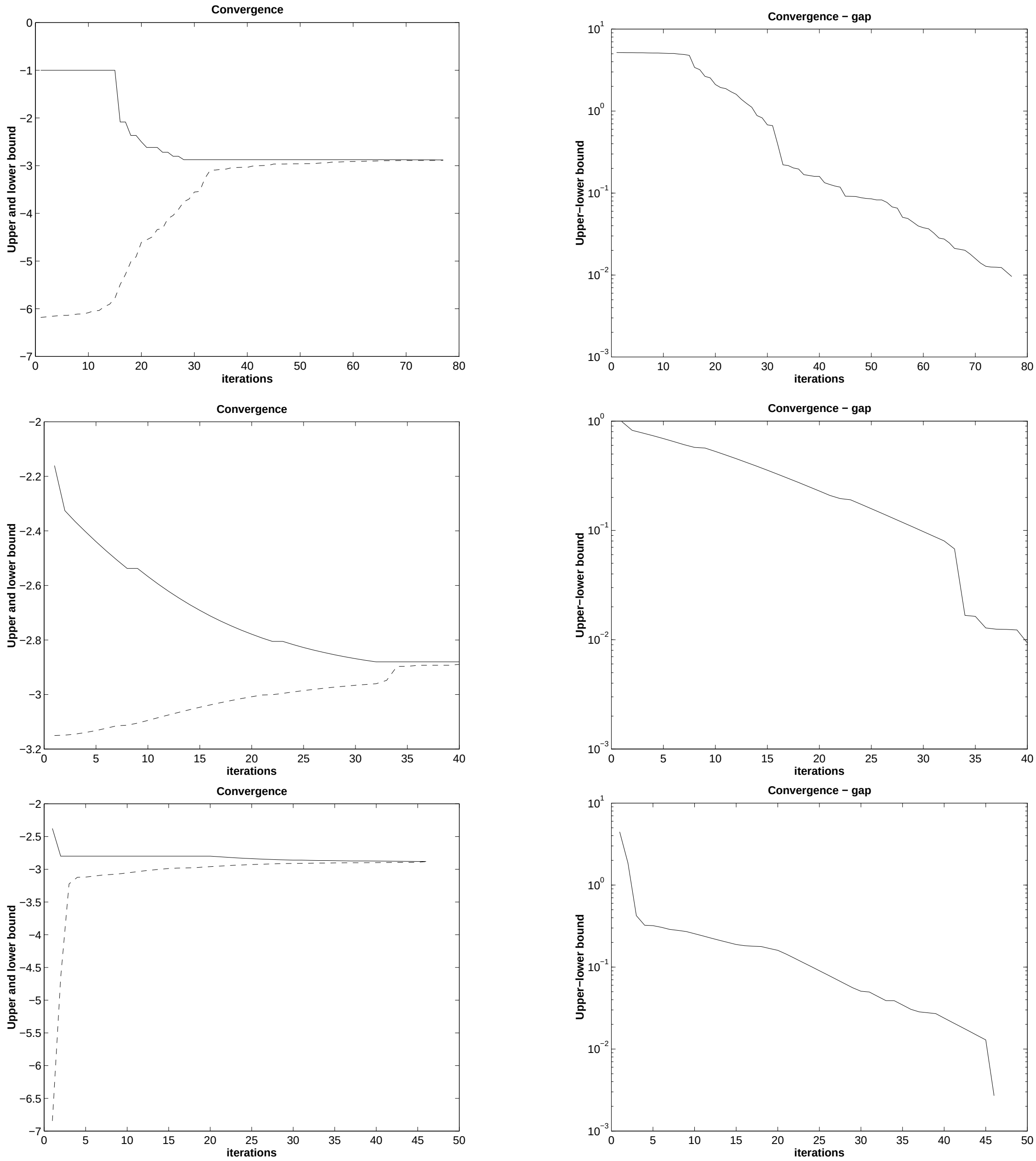


Figure 5.13: Upper and lower bounds for CP I/II/III.

First row is CP I, second CP II, and third CPIII. To the left a plot of the best upper bound $\overline{\phi}^{(l)}$ and best lower bound $\underline{\phi}^{(l)}$ for iterations $l = 1, \cdots, k$, to the right a logarithmic plot of the difference between the two.

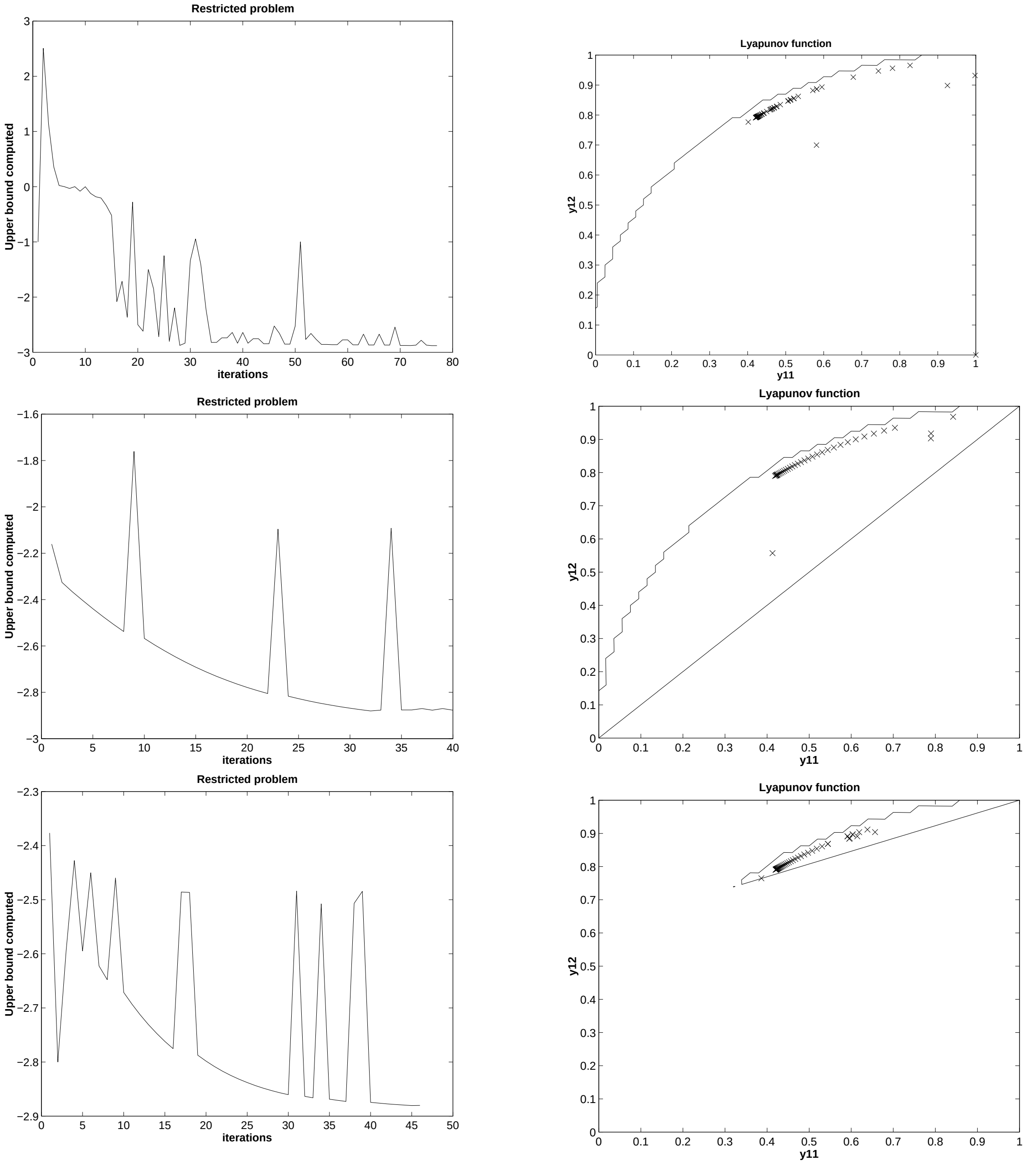


Figure 5.14: Upper bound/Lyapunov function for CP I/II/III.
First row is CP I, second CP II, and third CPIII. To the left a plot of the solution of the restricted problem $\overline{\phi}^{(k)}$. To the right a plot of the Lyapunov function, that is Y_{11} and Y_{12}, for each iteration. Each iteration is marked with a 'x', whereas the solid line surrounds the feasibility region $\{y : A(y) \geq 0\}$, which is computed using the MATLAB routine `contour` and therefore slightly inaccurate.

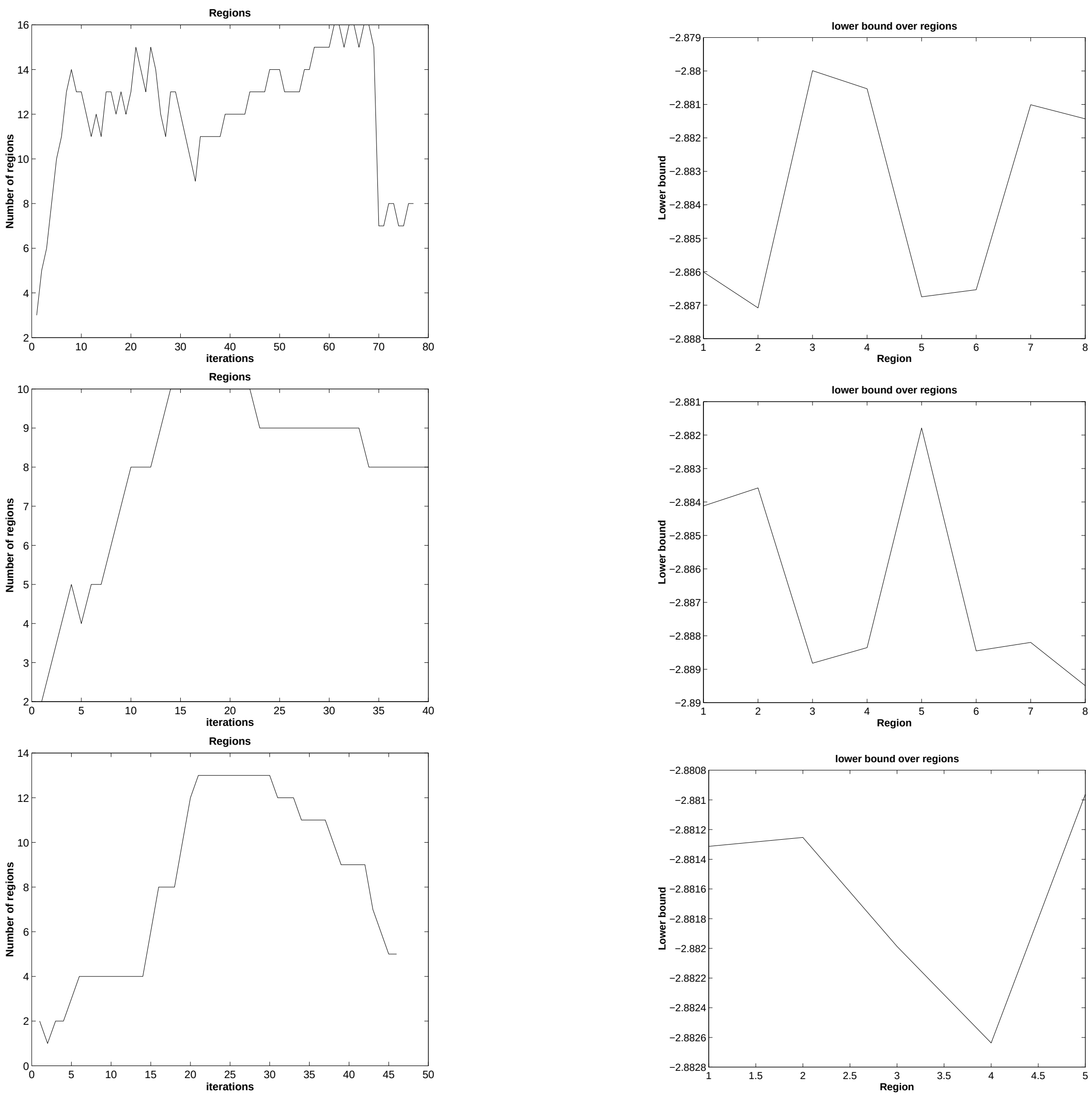


Figure 5.15: Number of subregions/lower bounds for regions remaining for CPI/II/III. First row is CP I, second CP II, and third CPIII. To the left the number of subregions at each iteration. To the right the lower bounds for the remaining regions at the end of the algorithm.

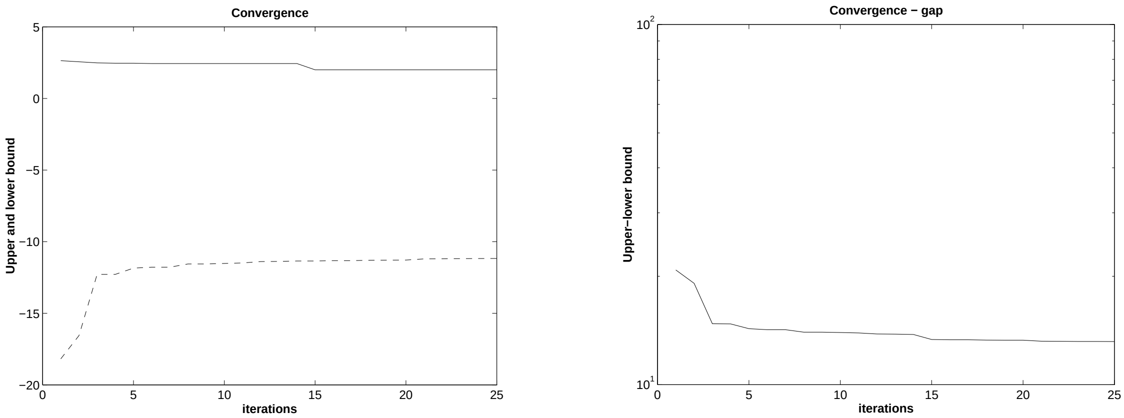


Figure 5.16: Upper and lower bounds/gap for Two wagon example.
To the left a plot of the best upper bound $\overline{\phi}^{(l)}$ and best lower bound $\underline{\phi}^{(l)}$ for iterations $l = 1, \cdots, k$, to the right a logarithmic plot of the difference between the two.

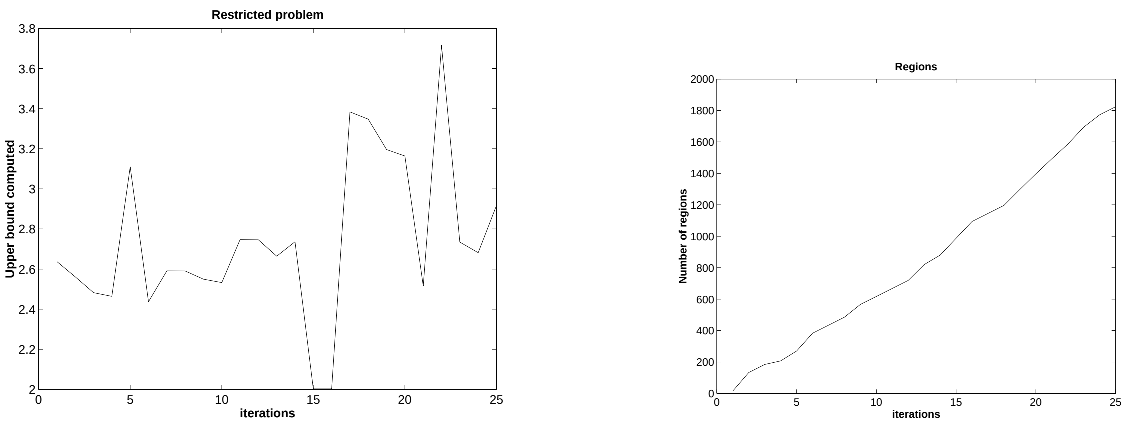


Figure 5.17: Upper bound/Number of subregions for Two Wagon example.
To the left a plot of the solution of the restricted problem $\overline{\phi}^{(k)}$. To the right the number of subregions at each iteration.

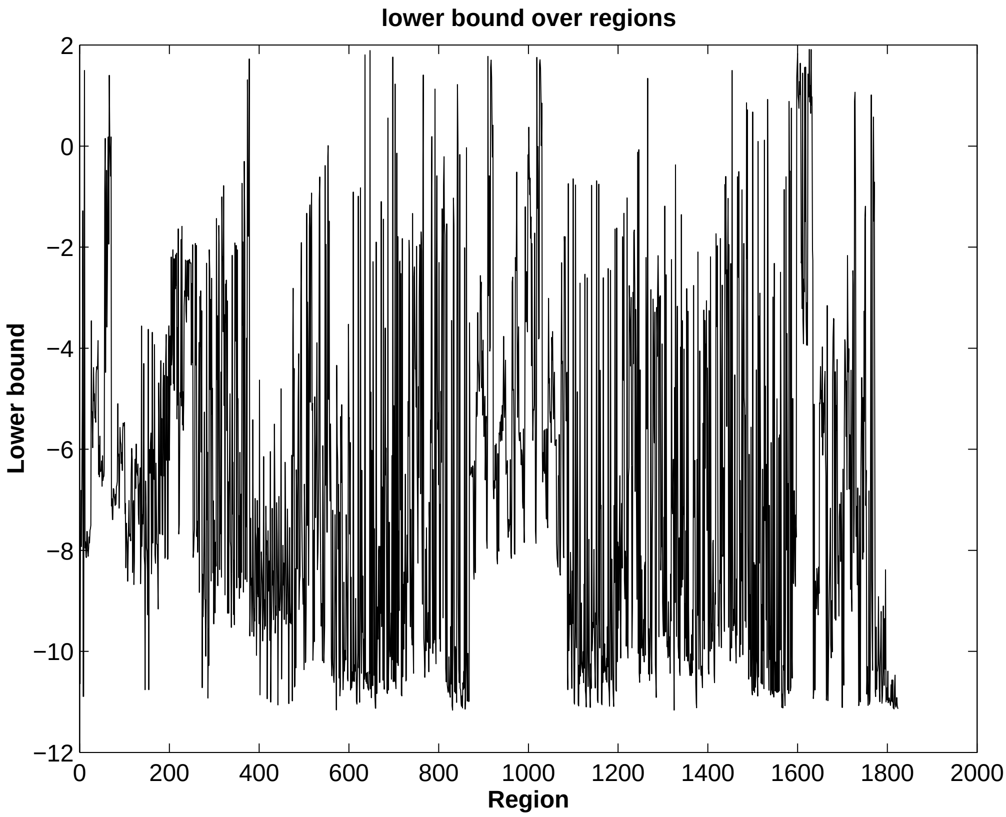


Figure 5.18: Lower bounds for regions remaining for Two Wagon example.

Part III

Summary and conclusion

Chapter 6

Discussion and future work

Here we summarize and discuss the results presented in the last two chapters.

6.1 Heuristics algorithms for low-order control design.

The algorithm presented in chapter 4 was developed together with Karolos Grigoriadis. The algorithm is heurestic and can as such not be expected to perform perfectly or even well in all cases. The algorithm has however

It seem to work very well in practice for some examples. When it came to random examples there was still some situations where the

6.2 Global Optimization of BMI problems

Chapter 7

Conclusion

Everything is nice!!!

Part IV

Appendices

Appendix A

Induced Norm Control Toolbox

The tools in the INCT can be divided into four different types of specifications. One control problem was given in the previous section with respect to the control objective of making the induced norm less than γ. In the table below the possible control objectives in the toolbox are given together with a short description and acronyms:

1)	Stability:	Poles to the left of $-\alpha + j\mathbb{R}$.	'Stab'
2)	Performance:	8 different induced norms.	'Perf'
3)	Robustness:	unstructured uncertainties($\mathcal{H}_\infty$)	'Robust'
4)	Robust Performance:	Guaranteed LQ performance under unstructured uncertainties.	'RobPerf'

Analysis is possible for all of the above specifications. Synthesis is possible for stability, performance, and robustness. A good algorithm does not exist for the robust performance design.

The toolbox contains a user friendly packing system called 'gstspace'. The main idea is that complex setups, like equation (2.14), are specified by a fairly big number of matrices when it comes to writing them down. The package provides a compact form for such system setups. This makes it easier to write the MATLAB code to specify a control problem when the system is first packed. Computing closed-loop matrices and extracting vital information becomes easier. Moreover, the package also provides scaling tools that may improve numerical stability.

The main tools are the functions 'INAnl', 'INView' and 'INSynth'. The first tool deals with analysis of the specifications listed above, whereas the second tool provides an overview of all the 8 different norms specified in table 2.1. The synthesis tool, the function 'INSynth', can take up to six arguments:

1)	The system	Packed with 'gstspace'.
2)	Control objective	Specified by acronyms, e.g. 'PerfE2EP'.
3)	Objective bound	degree, e.g. 'degree opt' or 'degree 0.2'.
4)	Controller order	optional, e.g. 'Nclow' or 'Nc 2'.
5)	Debug information	optional, e.g. '1' print out main loops.

Specifications 2) through 5) are written in one string, e.g. 'PerfEtEP degree 0.2 Ncdown'. This specifies: find a controller that solves the energy to Euclidean peak control problem with $\gamma = 0.2$. Moreover, the controller order is decreased by at least one(see next paragraph). Usually MATLAB functions take several numeric parameters, each having a special meaning. Using a string is more user friendly, because the specifications are written in words and not numbers refering to the specifications.

For all control problems the controller order can be explicitly considered. This is an advantage with the LMI approach for controller design that the controller order can be decreased to at least the system order minus one using a convex optimization approach. Moreover, methods exist, but are not included in the toolbox, to decrease the controller order more by using some heuristic method, see for instance[BG96, GS96].

A.1 Example

We will demonstrate some of the features in the toolbox by a simple example. Consider the two mass-spring benchmark example, see for instance [BG96, GS96]. A schematic figure of the model is given in figure A.1. The kernel is a system with two wagons connected with a spring, i.e. a fourth order system with four poles on the imaginary axis. The actuator signal is the force on the first wagon, and the measurement is the position of the second wagon. The actuator

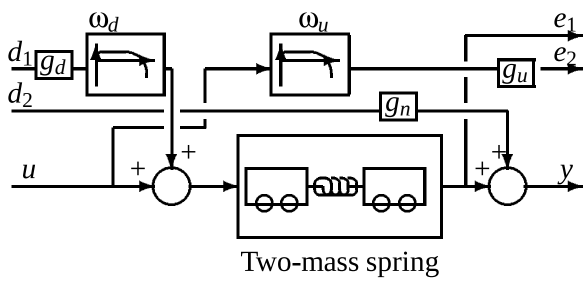


Figure A.1: Simple example

is disturbed by a Coulomb friction which we, very conservative, will approximate with a low frequency($\omega_d = 0.3$) energy bounded disturbance with amplitude $g_d = 0.1$. The measurement signal contains noise d_2 that is bounded in energy and with amplitude g_n, but may contain all frequencies. We want to track the position of the second wagon error signal e_1. For simplification we will not introduce a non-zero reference in this model. The second error signal is a low-pass filtered($\omega^u = 10$) version of the actuator signal. The control objective is to minimize the energy to peak from the two disturbances to the errors. We will also compute a controller that minimizes the energy to energy induced norm ($\mathcal{H}_\infty$). Since the model is of order 6, we expect the controller to be of order 5. For the energy to peak design we will use the feature in the toolbox that allows a trade-off between the performance and some internal parameters.

Controller	RtE	ERtE	EtP	EtEP	EtE	*BStBP*
EtP design	0.181	0.187	0.199	0.205	0.916	0.217
EtE design	1.389	1.389	1.388	1.388	0.159	1.398

Table A.1: Induced norms for controller design

The two controllers were computed using LMITOOL and SP. The MATLAB file used for the design is presented in figure A.3. We will not give the controllers in this paper, but encourage the reader to download the toolbox and try the example `mlinp.m`. We will however present some of the results, and compare the two controllers. Both controllers were of order five. The magnitude plot of the transfer functions from disturbance to error is presented next to the m-file in figure A.2. The EtP design is close to the EtE design in low frequencies. Moreover, the EtP design gives a more sharp magnitude plot against the very flat one of $\mathcal{H}_\infty$ design. To the right, the different induced norms of the closed-loop are presented for each of the two controllers. The energy to peak design gives the best performance in the designated induced norm, but it also drags down the other induced norms except the energy to energy. For the second design the energy to energy design gives the lowest performance in terms of energy to energy. The water bed effect should be evident.

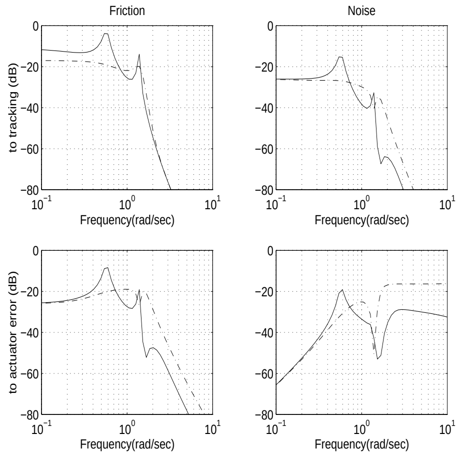


Figure A.2: Disturbance to error
The solid line is the energy to peak design, and the dash-dot line is the energy to energy design.

```
% Induced norm performance
% synthesis problem

% setting up matrices
wlp = 0.3; % LP BW for actuator disturbance
dg = 0.1;  % disturbance gain
wulp = 10; % LP filter for actuator error
ug = 0.5;
ng = 0.05; % measurement noise gain
A = [0  0 1 0 0 0;
     0  0 0 1 0 0;
     -1 1 0 0 dg 0;
     1 -1 0 0 0  0;
      0 0 0 0 -wlp 0;
     0  0 0 0 0   -wulp];
Bu = [0 0 1 0 0 wulp]';
Bd = [0 0 0 0 wlp 0;
      0 0 0 0 0  0]';
Cy = [0 1 0 0 0 0];,
Ce = [0 1 0 0 0 0;0 0 0 0 0 ug];
Ded = [0 0;0 0];, Dyd = [0 ng];
Deu = [0;0];, Dyu = 0;
% packing system
tot = [A Bd Bu;Ce Ded Deu;Cy Dyd Dyu];
G = gpck(tot,6,gstype(1,1,0),2,2);
% calling design algorithm
CPspec = 'PerfEtP';
if (syn)
[Gc,errmsg] = INSynth(G,[CPspec ...
    'degree opt Ncdown GcNorm 100'],0);
error(errmsg)
end
if (syn2)
CPspec = 'PerfEtE';
[Gc2,errmsg] = INSynth(G,[CPspec ...
    'degree 0.16 Ncdown GcNorm 100'],0);
end
error(errmsg)
% computing closed loop
Gt = gclloop(G,Gc);
Gt2 = gclloop(G,Gc2);
% Analyzing result
disp('EtP design')
INView(Gt)
disp('EtE design')
INView(Gt2)
```

Figure A.3: The MATLAB -file
that calls the INCT to design and analyze the results of the design example in section A.1.]

A.2 Experimental results

In this section extensive numerical experimentation is provided to assess the efficiency and the complexity of the proposed alternating projection schemes for the solution of LMI problems with rank constraints. The fixed-order α-stabilization problem for a nth-order spring-mass system, a helicopter system and the static output feedback stabilization problem for randomly generated systems are considered. The results were obtained on an HP735 workstation using MATLAB version 4.2 and the SP package by Boyd and Vandenberghe [BV94].

A.2.1 Randomly Generated Stabilizable Systems

The first numerical experiment considers randomly generated static output-feedback stabilizable systems. A stabilizable system is obtained by reflecting the eigenvalues of randomly generated matrices via an eigenvalue-eigenvector decomposition where the positive eigenvalues are replaced with their negative values. Then, a product of arbitrary input, feedback gain and output matrices is subtracted from this matrix to guarantee that the result is static-output-feedback stabilizable. By shifting all eigenvalues by an amount $-\alpha$, where α is a positive scalar, desired α-degree of stability can be introduced in the static output feedback problem.

Case	System Order n	Number of inputs n_u	Number of outputs n_y
a	4	1	1
b	6	2	2
c	8	3	3
d	4	3	2
e	6	4	3
f	8	5	4

Table A.2: Sizes for the random generated experiments.

We seek to obtain the lowest order α-stabilizing controller obtained via the proposed alternating projection methods. We will compare this result with the following Kimura bound k_b, that provides an upper bound on the control order for the stabilizing control problem [Kim75]

$$k_b = n - n_u - n_y + 1 \quad .$$

This bound is for $\alpha = 0$. Table A.2 shows the cases we have considered. Notice that the Kimura bound k_b is equal to $k_b = 3$ for cases a), b) and c), and $k_b = 0$ for cases d), e) and f). For each one of the above six cases 200 hundred random experiments were carried out. A degree of stability $\alpha = 0.1$ has been introduced in the randomly generated systems and the objective is to obtain the lowest order stabilizing controller that places the closed-loop poles to the left of $-\alpha$. Tables A.3 and A.4 show the results for each one of the cases considered in Table A.2.

¿From these results it is observed that in the majority of the experiments, the lowest order achievable controller is obtained in 0 iterations, that is by solving the convex problem (4.14) as described in Section 4.3. In all the experiments, the lowest order achievable controller is of order lower or equal to the Kimura bound k_b. In 5 experiments for case a), the lowest order achievable controller was 1, instead of the zeroth order that is guaranteed by the construction of the experiments.

Controller order	Outer iterations	Case a		Case b		Case c	
		#	Rate	#	Rate	#	Rate
0	0	176	88.00	177	88.50	184	92.00
	1	0	0.00	2	1.00	3	1.50
	2	3	1.50	7	3.50	7	3.50
	3	3	1.50	4	2.00	2	1.00
	4	4	2.00	3	1.50	1	0.50
	5	3	1.50	0	0.00	0	0.00
	6	0	0.00	2	1.00	1	0.50
	7-38	6	3.00	5	2.50	1	0.50
1		5	2.50	0	0.00	1	0.50
Average CPU time		18.71 s		15.93 s		26.15 s	

Table A.3: The results for the random example cases a), b), and c).
All controllers were of order 0 or 1, as expected with the Kimura bound $k_b = 3$. The CPU time is the average time used on all examples.

Outer iterations	Case d		Case e		Case f	
	#	Rate	#	Rate	#	Rate
0	200	100.00	199	99.50	196	98.00
1	0	0.00	0	0.00	0	0.00
2	0	0.00	0	0.00	3	1.50
3	0	0.00	1	0.50	0	0.00
4	0	0.00	0	0.00	1	0.50
CPU time	1.17s		3.19 s		11.18s	

Table A.4: The results for the random example cases d), e), and f)
. All controllers were of order 0, as expected with the Kimura bound $k_b = 0$.The CPU time is the average time used on all examples.

A.2.2 Helicopter Example

The following example is from [KBH88]. The goal is to obtain a static state feedback controller for the following helicopter model such that the closed-loop poles are located to the left of $-\alpha = -0.1$ at the complex plane. The system is of the form (2.14), and the data are the following

$$\mathsf{A} = \begin{bmatrix} -0.0366 & 0.0271 & 0.0188 & -0.4555 \\ 0.0482 & -1.01 & 0.0024 & -0.4555 \\ 0.1002 & 0.3681 & -0.7070 & 1.4200 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \ \mathsf{B}_u = \begin{bmatrix} 0.4422 & 0.1761 \\ 3.5446 & -7.5922 \\ -5.5200 & 4.4900 \\ 0 & 0 \end{bmatrix},$$
$$\mathsf{C}_y = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$$

The algorithm converged in zero iterations, that is the solution of the convex minimization problem (4.14) provides a static controller

$$u = \begin{bmatrix} -0.2162 \\ 2.4942 \end{bmatrix} y \tag{A.1}$$

that achieves the desired objective. The closed-loop poles are -20.8379, -0.1042 and $-0.2572 \pm 0.9738j$. The required CPU time to obtain this controller is 1.27 sec.

A.2.3 Spring-Mass Systems

In this numerical experiment interconnected spring-mass system models were generated. The order of the system is equal to twice the number of the interconnected masses. We now proceed to examine higher order spring mass systems. Tables A.5 and A.6 provide the lowest order achievable controller and the number of iterations needed for $\alpha = 0.001$ and $\alpha = 0.1$ for different numbers of the system masses.

Number of masses	Number of iterations	Lowest order achievable controller
2	0	2
3	0	3
4	0	4
5	0	5

Table A.5: Results for degree of stability $\alpha = 0.001$

Hence, either zero or one iteration of the APSP algorithm is enough to provide a static output feedback controller depending on the desired degree of stability α.

Number of masses	Number of iterations	Lowest order achievable controller
2	0	2
3	0	3
4	1	5
5	1	6

Table A.6: Results for degree of stability $\alpha = 0.1$

Appendix B

BMI optimization algorithms

B.1 Bilinear relaxation BB algorithm

Tree structure

The tree structure used to explain the algorithm can be simulated by making a set $\mathcal{V}$. This set contains all leaves of the original tree, that is the union of the sets in $\mathcal{V}$ constitutes to a partition of the original region. In the beginning $\mathcal{V} = \left\{Q^{(1)}\right\}$. In each iteration k the set $Q^{(k)}$ is removed from $\mathcal{V}$ and the two new sets $Q_1^{(k)}$ and $Q_2^{(k)}$ are added. Together with the definition of the regions should be added some additional data. From the solutions to the relaxation problems, the center points $\hat{y}_s^{(k)}$ for each new regions should be computed, and added to the list of the current regions.

Pruning The set $\mathcal{V}$ should frequently be checked for regions where the computed lower bound is above the global upper bound, and these regions can be removed. Note that this will not constitute to fewer iterations, since such a region never would have been considered by the algorithm. However, it do save memory.

For the regions already split it is only important to keep the best solution, that is ν and x^*, y^*.

Bibliography

[ANP96] J. Abedor, K. Nagpal, and K. Poolla. A linear matrix inequality approach to peak-to-peak gain minimization. *International Journal of Robust and Nonlinear Control*, 6:899–927, 1996.

[BB92] V. Balakrishnan and S. Boyd. Global optimization in control system analysis and design. In C.T. Leondes, editor, *Control and Dynamic Systems - High Performance Systems Techniques and Applications*, volume 53. Academic Press, Inc., 1992.

[BBB91a] V. Balakrishnan, S. Boyd, and S. Balami. Computing the minimum stability degree of parameter-dependent linear systems. In *Control of Uncertain Dynamic Systems*. CRC Press, 1991.

[BBB91b] V. Balakrishnan, S. Boyd, and S. Balemi. Branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems. *Int. Journal of Robust and Nonlinear Control*, 1:295–317, 1991.

[BD86] J. P. Boyle and R. L. Dykstra. A method for finding projections onto the intersection of convex sets in Hilbert space. *Lecture Notes in Statistics*, 37:28–47, 1986.

[BE93] S. Boyd and L. El Ghaoui. Method of centers for minimizing generalized eigenvalues. *Linear Algebra and Applications, special issue on Numerical Linear Algebra Methods in Control, Signals and Systems*, 188:63–111, July 1993.

[Ben62] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numer. Math.*, 4:238–252, 1962.

[Ber94] E. Beran. Robust and Optimal Control - Theory and Design. Master's thesis, Technical University of Denmark, Department of Mathematics, August 1994. Available via `http://www.iau.dtu.dk/~ebb/INCT`.

[Ber95] D.P. Bertsekas. *Nonlinear programming*. Athena Scientific, 1995.

[BFHT95] B.R. Barmish, C.A. Floudas, C.V. Hollot, and R. Tempo. A global linear progamming solution to some open robustness problems including matrix polytope stability. In *Proceedings of the American Control Conference*, pages 3871–3877, June 1995.

[BG96] E. Beran and K.M. Grigoriadis. A combined alternating projection and semidefinite programming algorithm for low-order control design. In *IFAC 96, San Francisco*, volume C, pages 85–90, July 1996.

[BG97] E. Beran and K.M. Grigoriadis. *Induced Norm Control Toolbox INCT*, 1997. Available via `http://www.iau.dtu.dk/~ebb/INCT`.

[BGFB94] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM, 1994.

[BGL95] V. Blondel, M. Gevers, and A. Lindquist. Survey on the state of systems and control. *European Journal of Control*, 1(1):5–23, 1995.

[BT95] V. Blondel and J. Tsitsiklis. NP-hardness of some linear control design problems. In *CDC*, pages 2910–2915, New Orleans, LA, 1995.

[BV94] S. Boyd and L. Vandenberghe. SP - Software for Semidefinte Programming. Users guide. Available via ftp to `isl.stanford.edu` in `pub/boyd/reports`, November 1994. Beta version.

[BV97a] S. Boyd and L. Vandenberghe. *Convex optimization*. in preparation, 1997.

[BV97b] S. Boyd and L. Vandenberghe. Semidefinite programming relaxations of non-convex problems in control and combinatorial optimization. Communications, Computation, Control and Signal Processing: a tribute to Thomas Kailath, Kluwer, 1997.

[BW95] S. Boyd and S. Wu. *A Parser/Solver for Semidefinite Programs with Matrix Structure*. Stanford University, November 1995.

[CG59] W. Cheney and A. A. Goldstein. Proximity maps for convex set. *Proc. Amer. Math. Society*, 12(2):448–450, 1959.

[CG96] M. Chilali and P. Gahinet. $\mathcal{H}_\infty$ design with pole placement constraints: An LMI approach. *IEEE Transactions on Automatic Control*, 41(3):358–367, March 1996.

[Cla83] F.H. Clarke. *Optimization and Nonsmooth Analysis*. John Wiley & Sons, 1983.

[CT90] P.L. Combettes and H. J. Trussell. Method of successive projections for finding a common point of sets in metric spaces. *Journal of Optimization Theory and Applications*, 1990.

[DGKF89] J. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis. State-space solutions to standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems. *IEEE Transactions on Automatic Control*, AC-34(8):831–847, August 1989.

[FL97] M. Fu and Z.-Q. Luo. Computational complexity of a problem arising in fixed order output feedback design. *Systems & Control Letters*, 1997. (To appear).

[FV93] C.A. Floudas and V. Visweswaran. A primal-relaxed dual global optimization approach. *Journal on Optimization Theory and Application*, 78(2):187, 1993.

[GA93] P. Gahinet and P. Arkarian. An LMI-based parameterization of all $\mathcal{H}_\infty$ with applications. In *Proceedings of the Conference on Decision and Control*, San Antonio, Texas,USA, 1993.

[GA94] P. Gahinet and P. Apkarian. A linear matrix inequality approach to $\mathcal{H}_\infty$ control. *International Journal Robust and Nonlinear Control*, 4, 1994.

[GDN95] L. El Ghaoui, F. Delebecque, and R. Nikoukhah. LMITOOL: A User-Friendly Interface for LMI Optimization. Available via ftp to `ftp.ensta.fr` in `pub/elghaoui/lmitool`, February 1995. Beta version.

[Geo71] A. M. Geoffrion. Duality in nonlinear programming: a simplified applications-oriented development. *SIAM review*, 13(1):1–37, January 1971.

[Geo72] A. M. Geoffrion. Generalized Benders decomposition. *JOTA*, 10:237–260, 1972.

[GLS96] K.M. Grigoriadis, J. Lu, and R.E. Skelton. $\mathcal{L}_2$ and $\mathcal{L}_2$-$\mathcal{L}_2$ model reduction via linear matrix inequalities. In *Proceedings of the Conference on Decision and Control*, Kobe, Japan, December 1996.

[GNLC95] P. Gahinet, A. Nemirovski, A. Laub, and M. Chilali. LMI control toolbox, 1995. MATLAB.

[GOA95] L. El Ghaoui, F. Oustry, and M. Aitrami. An LMI-based linearization algorithm for static output-feedback and related problems. *IEEE Transactions on Automatic Control*, December 1995. Submitted to.

[Goh95] K.C. Goh. *Robust Control Synthesis via Bilinear Matrix Inequalities*. PhD thesis, University of Southern California, May 1995.

[GPR67] L. G. Gubin, B. T. Polyak, and E. V. Raik. The method of projections for finding the common point of convex sets. *USSR Comp. Math. Physics*, 1967.

[Gri95] K.M. Grigoriadis. Optimal $\mathcal{H}_\infty$ model reduction via linear matrix inequalities continuous- and discrete-time cases. *Systems & Control Letters*, 26:321–333, 1995.

[GS94] K. M. Grigoriadis and R.E. Skelton. Alternating convex projection methods for covariance control design. *International Journal of Control*, 60:1083–1106, 1994.

[GS96] K. M. Grigoriadis and R.E. Skelton. Low-order control design for LMI problems using alternating projection methods. *Automatica*, 32(8):117–1125, August 1996.

[GSL95] K.C. Goh, M.G. Safonov, and J.H. Ly. Robust synthesis with bilinear matrix inequalities. *Accepted for the International Journal of Robust and Nonlinear Control*, 1995.

[GSP94] K.-C. Goh, M.G. Safonov, and G.P. Papavassilopoulos. A global optimization approach for the BMI problem. In *Proceedings of the Conference on Decision and Control*, Orlando, Florida, 1994.

[GSP95] K.C. Goh, M.G. Safonov, and G.P. Papavassilopoulos. A global optimization approach for the BMI problem. *J. of Global Optimization*, 7:365–380, 1995.

[GTS+94] K.C. Goh, L. Turan, M.G. Safonov, G.P. Papavassilopoulos, and J.H. Ly. Biaffine matrix inequality propertieas and computational methods. *Proceedings of the American Control Conference*, pages 850–855, 1994.

[GW96] K.M. Grigoriadis and J.T. Watson. Suboptimal $\mathcal{H}_\infty$ filtering via linear matrix inequalities. In *IFAC*, volume G, pages 197–202, July 1996.

[Han88] S. P. Han. A successive projection method. *Mathematical Programming*, 40:1–14, 1988.

[Hig88] N. J. Higham. Computing the nearest symmetric positive semidefinite matrix. *Linear Algebra and its Applications*, 103:103–118, 1988.

[HJ91] R.A. Horn and C.R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[HJ95] P. Hansen and B. Jaumard. Lipschitz optimization. In R. Horst and P. Pardalos, editors, *Handbook of Global Optimization*, pages 407–493. Kluwer Acedemic Publishers, 1995.

[IS93] T. Iwasaki and R.E. Skelton. A complete solution to the general $\mathcal{H}_\infty$ control problem: LMI existence conditions and state space formulas. In *Proceedings of the American Control Conference*, San Francisco., California, 1993.

[IS94] T. Iwasaki and R.E. Skelton. All controllers for the general $\mathcal{H}_\infty$ control problem: LMI existence conditions and state space formulas. *Automatica*, 30, 1994.

[IS95] T. Iwasaki and R.E. Skelton. The XY-centering algorithm for the dual LMI problem: A new approach to fixed-order control design. *International Journal of Control*, 62(6):1257–1272, 1995.

[Iwa93] T. Iwasaki. *A Unified Matrix Inequality Approach to Linear Controller Design*. PhD thesis, Purdue University, 1993.

[Iwa97a] T. Iwasaki. Control synthesis for well-posedness of feedback systems. In L. El Ghaoui and S.-I. Niculescu, editors, *Recent advances on LMI methods in Control*. SIAM, 1997.

[Iwa97b] T. Iwasaki. The dual iteration for fixed order control. In *Proceedings of the American Control Conference*, Albuquerque, New Mexico, June 1997.

[KB60] R. E. Kalman and J. E. Bertram. Control system analysis and design via the "second method" of Lyapunov: Part I, continuous-time systems. *Transactions of the ASME, Journal of Basic Engineering*, 82(59-NAC-2):371–393, June 1960.

[KBH88] L.H. Keel, S.P. Bhattacharyya, and J.W. Howze. Robust control with structured perturbations. *IEEE Transactions on Automatic Control*, 33(1):68–78, 1988.

[Kim75] H. Kimura. Pole assignment by gain output feedback. *IEEE Transactions on Automatic Control*, 20(5):509–516, 1975.

[Kri97] M. M. Kristensen. An algorithm for reduced order controllers. Master's thesis, Technical University of Denmark, Department of Mathematics, August 1997.

[KYS96] M. Kawanishi, T. Yada, and T. Sugie. Controller design of 2 mass-spring system based on BMI optimization. In *Proceedings of the Conference on Decision and Control*, pages 2363–2364, Kobe, Japan, December 1996.

[Lue68] D. Luenberger. *Optimization by Vector Space Methods*. John Wiley and Sons, 1968.

[LW66] E.L. Lawler and D.E. Wood. Branch-and-bound methods: A survey. *Operations Research*, 14:699–719, 1966.

[Lya47] A.M. Lyapunov. *Problème général de la stabilité du mouvement*. Annals of Mathematics Studies. Princeton University Press, Princeton, 1947.

[MOS95] I. Masubuchi, A. Ohara, and N. Suda. LMI-based output feedback controller design. In *Proceedings of the American Control Conference*, pages 3473–3477, 1995.

[MP96] M. Mesbahi and G. P. Papavassilopoulos. A cone programming approach to the bilinear matrix inequality problem and its geometry. Technical report, University of Southern California, 1996. Accepted for publication in Mathematical Programming.

[MP97a] M. Mesbahi and G. P. Papavassilopoulos. On the rank minimization problem over a positive semidefinite linear matrix inequality. *IEEE Transactions on Automatic Control*, 42(2), February 1997.

[MP97b] M. Mesbahi and G. P. Papavassilopoulos. Solving a class of rank minimization problems via semi-definite programs, with application to the fixed order output feedback synthesis. In *Proceedings of the American Control Conference*, 1997.

[NG94] A. Nemirovskii and P. Gahinet. The projective method for solving linear matrix inequalities. *Proceedings of the American Control Conference*, pages 840–844, 1994.

[NN94] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, 1994.

[Pag96] F. Paganini. Robust $\mathcal{H}_2$ analysis for continuous time systems. Technical Report LIDS-P-2342, Massachusetts Institute of Technology, January 1996.

[PF95] P. Psarris and C.A. Floudas. Robust stability analysis of systems with real parametric uncertainty: A global optimization approach. *International Journal of Robust and Nonlinear Control*, 5:699–717, 1995.

[PLD94] R.V. Patel, A.J. Laub, and P.M. Van Dooren. *Numerical Linear Algebra Techniques for Systems and Control*. IEEE Pres, 1994.

[PZPB91] A. Packard, K. Zhou, P. Pandey, and G. Becker. A collection of robust control problems leading to LMIs. In *Proceedings of the Conference on Decision and Control*, pages 1245–1250, Brighton, England, December 1991.

[Sch90] C. Scherer. *The Riccati Inequality and State-Space $\mathcal{H}_\infty$-Optimal Control*. PhD thesis, University of Würzburg, 1990.

[Sch95] C. Scherer. Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control. In Alberto Isidori, editor, *Trends in Control - A European Perspective*. Springer Verlag, 1995.

[SG91] N.V. Sahinidis and I.E. Grossmann. Convergence properties of generalized Benders decomposition. *Computers and Chemical Engineering*, 15(7):481–491, 1991.

[SGC97] C. Scherer, P. Gahinet, and M. Chilali. Multi-objective output-feedback control via LMI optimization. *IEEE Transactions on Automatic Control*, January 1997. Revised version submitted.

[SGL94] M.G. Safonov, K.C. Goh, and J.H. Ly. Control system synthesis via bilinear matrix inequalities. *Proceedings of the American Control Conference*, pages 45–49, 1994.

[SIG97] R.E. Skelton, T. Iwasaki, and K.M. Grigoriadis. *A Unified Algebraic Approach to Control Design*. Taylor and Francis, 1997. Completed manuscript.

[Sto92] A. A. Stoorvogel. *The $\mathcal{H}_\infty$ Control Problem: A State Space Approach*. Prentice Hall, 1992. Out of print, available via internet.

[SUNS96] B. Shafai, V. Uddin, H.H. Niemann, and J. Stoustrup. An LMI approach to fixed order LTR controller. In *Proceedings of the Conference on Decision and Control*, pages 4769–4771, Kobe, Japan, 1996.

[THT96] H.D. Tuan, S. Hosoe, and H. Tuy. D.C. optimization approach to robust control: Feasibility problems. Technical Report 9601, Nagoya University, October 1996.

[TO95] O. Toker and H. Özbay. On the NP-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback. In *Proceedings of the American Control Conference*, 1995.

[Tuy95] H. Tuy. D.C. optimization: Theory, methods, and algorithms. In R. Horst and P. Pardolos, editors, *Handbook of Global Optimization*, pages 149–216. Kluwer Academic Publishers, 1995.

[Van97] J. G. VanAntwerp. Globally optimal robust control for systems with time-varying nonlinear perturbations. Master's thesis, University of Illinois at Urbana-Champaign, 1997.

[Vav95] S.A. Vavasis. Complexity issues in global optimization: A survey. In Reiner Horst and Panos M. Pardalos, editors, *Handbook of Global Optimization*, pages 27–42. Kluwer Academic Publishers, 1995.

[VB94] L. Vandenberghe and S. Boyd. SP, software for semidefinite programming, users quide. Available via `http://www-ISL.stanford.edu/people/boyd/SP.html`, November 1994.

[VB96] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM review*, March 1996.

[VBW96] L. Vandenberghe, S. Boyd, and S.-P. Wu. Determinant maximization with linear matrix inequality constraints. *submitted to SIMAX*, February 1996.

[VF96a] Visweswaran V. and C.A. Floudas. New formulations and branching strategies for the GOP algorithm. In I.E. Grossmann, editor, *Global Optimization in Chemical Engineering,*, chapter 3, pages 75–100. Kluwer Academic Publishers, 1996.

[VF96b] V. Visweswaran and C.A. Floudas. cGOP: A deterministic global optimization package - user's guide, 1996.

[WA86] C. Weber and J. Allebach. Reconstruction of frequency offset Fourier data by alternating projections. In *Proc. 24th Allerton Conference*, pages 194–201, 1986.

[WB96] S.-P. Wu and S. Boyd. Design and implementation of a parser/solver for SDPs with matrix structure. In *Proceedings of the 1996 IEEE International Symposium on Computer-Aided Control System Design*, pages 240–245, Dearborn, Michigan, 1996.

[Wil71] J. C. Willems. Least squares stationary optimal control and the algebraic Riccati equation. *IEEE Transactions on Automatic Control*, AC-16(6), December 1971.

[Wil89] D.A. Wilson. Convolution and Hankel operator norms for linear systems. *IEEE Transactions on Automatic Control*, 34(1):94–97, January 1989.

[YW82] D. C. Youla and H. Webb. Image restoration by the method of convex projections. *IEEE Trans. Medical Imaging*, 1982.

[ZDG95] K. Zhou, J. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, New Jersey, 1995.