Semidefinite Programming Relaxations and Algebraic Optimization in Control

Pablo A. Parrilo[1] Sanjay Lall[2]

European Journal of Control, V. 9, No. 2-3, pp. 307–321, 2003

Abstract

We present an overview of the essential elements of semidefinite programming as a computational tool for the analysis of systems and control problems. We make particular emphasis on general duality properties as providing suboptimality or infeasibility certificates. Our focus is on the exciting developments occurred in the last few years, including robust optimization, combinatorial optimization, and algebraic methods such as sum-of-squares. These developments are illustrated with examples of applications to control systems.

Keywords: Semidefinite programming, linear matrix inequalities, control theory, duality, sum of squares.

[1]Automatic Control Laboratory, Swiss Federal Institute of Technology, CH-8092 Zürich, Switzerland. Email: `parrilo@control.ee.ethz.ch` (corresponding author).

[2]Department of Aeronautics and Astronautics, Stanford University, Stanford CA 94305-4035, U.S.A. Email: `lall@stanford.edu`

Contents

1 Introduction

This paper, prepared as part of a mini-course to be given at the 2003 European Control Conference, presents a self-contained treatment of the basics of semidefinite programming, as well as an overview of some recent developments in convex optimization and their application to control systems. There have been tremendous advances in theory, numerical approaches, and applications in the past few years, with developments including robust optimization, combinatorial optimization and integer programming, and algebraic methods. Our objective in the course, and in this paper, is to highlight some of these tremendously exciting developments, with an emphasis on control applications and examples.

The roots of semidefinite programming can be traced back to both control theory and combinatorial optimization, as well as the more classical research on optimization of matrix eigenvalues. We are fortunate that many excellent works dealing with the development and applications of SDP are available. In particular, we mention the well-known work of Vandenberghe and Boyd [60] as a wonderful survey of the basic theory and initial applications, and the handbook [64] for a comprehensive treatment of the many aspects of the subject. Other survey works, covering different complementary aspects are the early work by Alizadeh [1], Goemans [22], as well as the more recent ones due to Todd [58], and Laurent and Rendl [33]. The upcoming book [11] presents a beautiful and clear exposition of the theory, numerical approaches and broad applications of convex programming. Other works dealing more specifically with semidefinite programming for a control audience are the classical research monograph [10] and the subsequent [21]. Excellent available references that collect results at the different stages of the development of SDPs from a control perspective are the review paper of Doyle, Packard, and Zhou [16] and the lecture notes by Scherer and Weiland [53].

1.1 Notation

We make use of the following notation. The ***nonnegative orthant*** in $\mathbb{R}^n$ is denoted by $\mathbb{R}^n_+$, defined by

$$\mathbb{R}^n_+ = \left\{ x \in \mathbb{R}^n \mid x_i \geq 0 \text{ for all } i = 1, \dots, n \right\},$$

and the ***positive orthant*** $\mathbb{R}^n_{++}$ is

$$\mathbb{R}^n_{++} = \left\{ x \in \mathbb{R}^n \mid x_i > 0 \text{ for all } i = 1, \dots, n \right\}.$$

The nonnegative orthant is a closed convex cone. It defines a partial ordering on $\mathbb{R}^n$ as follows. For $x, y \in \mathbb{R}^n$ we write $x \geq y$ to mean $x - y \in \mathbb{R}^n_+$. Similarly $x > y$ means $x - y \in \mathbb{R}^n_{++}$. The set of real symmetric $n \times n$ matrices is denoted $\mathbb{S}^n$. A matrix $A \in \mathbb{S}^n$ is called ***positive semidefinite*** if $x^T A x \geq 0$ for all $x \in \mathbb{R}^n$, and is called ***positive definite*** if $x^T A x > 0$ for all nonzero $x \in \mathbb{R}^n$. The set of positive semidefinite matrices is denoted $\mathbb{S}^n_+$ and the set of positive definite matrices is denoted by $\mathbb{S}^n_{++}$. Both $\mathbb{S}^n_+$ and $\mathbb{S}^n_{++}$ are convex cones. The closed convex cone $\mathbb{S}^n_+$ induces a partial order on $\mathbb{S}^n$, and we write $A \succeq B$ to mean $A - B \in \mathbb{S}^n_+$. Similarly $A \succ B$ means $A - B \in \mathbb{S}^n_{++}$. Both the inequality $\geq$ defined on $\mathbb{R}^n$ and the inequality $\succeq$ defined on $\mathbb{S}^n$ are called ***generalized inequalities***. We also use the notation $A \leq 0$ to mean $-A \geq 0$, and similarly for $<$, $\prec$, and $\preceq$. A set $S \subset \mathbb{R}^n$ is called ***affine*** if $\lambda x + (1 - \lambda) y \in S$ for all $x, y \in S$ and all $\lambda \in \mathbb{R}$. It is called ***convex*** if $\lambda x + (1 - \lambda) y \in S$ for all $x, y \in S$ and all $\lambda \in [0, 1]$. The ring of polynomials in n variables with real coefficients is denoted $\mathbb{R}[x_1, \dots, x_n]$.

2 Formulations for Optimization Problems

2.1 Optimization and Feasibility Problems

In this paper we consider two main types of problems; *feasibility problems* and *optimization problems*. In a feasibility problem, we typically have a system of inequalities defined by functions $f_i : \mathbb{R}^n \to \mathbb{R}$ for $i = 1, \dots, m$. We are then interested in the set

$$P = \left\{ x \in \mathbb{R}^n \mid f_i(x) \leq 0 \text{ for all } i = 1, \dots, m \right\}. \tag{1}$$

Typically we would like to know if the set P is non-empty; that is, if there exists $x \in \mathbb{R}^n$ which simultaneously satisfies all of the inequalities. Such a point is called a ***feasible point*** and the set P is then called ***feasible***. If P is not feasible, it is called ***infeasible***. If P is known to be feasible, it is a further and typically more computationally difficult question to find a feasible x. Feasibility problems may also be defined using generalized inequalities, where the functions f_i map into $\mathbb{R}^n$ or $\mathbb{S}^n$, and the inequalities are induced by arbitrary proper cones.

In an optimization problem, we have an additional function $f_0 : \mathbb{R}^n \to \mathbb{R}$, and would like to solve the following problem.

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & x \in P \end{array} \tag{2}$$

There may also be additional linear equality constraints of the form $Ax = b$. These can be incorporated in the following development using minor additions, and we omit them to simplify the presentation.

2.2 Semidefinite Programming

A semidefinite program is a generalization of a *linear program* (LP), where the inequality constraints in the latter are replaced by generalized inequalities corresponding to the cone of positive semidefinite matrices.

Concretely, a semidefinite program (SDP) in the *pure primal form* is defined as the optimization problem

$$\begin{array}{lll} \text{minimize} & \operatorname{trace}(CX) & \\ \text{subject to} & \operatorname{trace}(A_i X) = b_i & \text{for all } i = 1, \ldots, m \\ & X \succeq 0, & \end{array} \tag{3}$$

where $X \in \mathbb{S}^n$ is the decision variable, $b \in \mathbb{R}^m$ and $C, A_1, \ldots, A_m \in \mathbb{S}^n$ are given symmetric matrices. There are m affine constraints among the entries of the positive semidefinite matrix X given by the equality relations. The crucial feature of semidefinite programs is that the feasible set defined by the constraints above is always convex. This is therefore a convex optimization problem, since the objective function is linear.

There are several alternative ways to write an SDP. We choose the above form since it exactly parallels the standard form used for linear programming, and is currently the most commonly used in the optimization community. Much of the literature on the use of SDPs for control has used a slightly different form with free variables, corresponding either a parametrization of the solution set (see below) or to the standard dual SDP discussed in Section 3.3. We join here the strong trend towards this unification of terminology.

We can understand the geometry of semidefinite programs in the following way. For a square matrix X, let X_k be the $k \times k$ submatrix consisting of the first k rows and columns of X. Then

$$X \succ 0 \qquad \text{if and only if } \det(X_k) > 0 \text{ for all } k = 1, \ldots, n.$$

That is, in principle we can replace the positivity condition in (3) by a set of polynomial inequalities in the entries of X. Note that, in the case of non-strict inequalities, the condition must be modified to include all minors, not just the principal ones. Sets defined by polynomial inequalities and equations are called *semialgebraic sets*. We therefore see that the feasible set defined by an SDP is a semialgebraic set. This construction is not too useful practically; however we shall see later that the reverse construction (converting a semialgebraic set to an SDP) is very useful. Unlike the LP case where the feasible sets are polyhedra, the feasible sets of SDPs (called *spectrahedra* in [48]) can have *curved* faces, together with sharp corners where faces intersect. For this reason, or equivalently, because of nondifferentiability of the eigenvalues, the problem (3), while convex, is not *smooth*.

An example is as follows. Consider the feasible set defined by $X \succ 0$ and the equality constraints

$$X_{11} - X_{12} + X_{22} = 7, \quad X_{33} - X_{12} + X_{22} = 4, \quad X_{13} = 1, \quad X_{23} = 0.$$

These four affine constraints define a two-dimensional matrix subspace, since $\dim(\mathbb{S}^3) = 6$, and $6 - 4 = 2$. For easier visualization, we find a parametrization of this two-dimensional

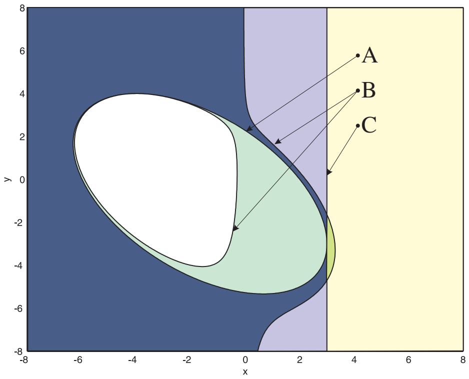

Figure 1: Example feasible set for an SDP

subspace in terms of the free variables (x, y), obtaining that the feasible set is

$$\left\{ X \in \mathbb{S}^n \;\middle|\; X = \begin{bmatrix} 3-x & -(x+y) & 1 \\ -(x+y) & 4-y & 0 \\ 1 & 0 & -x \end{bmatrix},\ X \succ 0,\ x, y \in \mathbb{R} \right\} \tag{4}$$

This set may readily be expressed as a semidefinite program of the form (3). By looking at the determinants of its principal submatrices, we have X satisfies (4) if and only if the parameters x and y simultaneously satisfy the polynomial inequalities

$$\begin{aligned} 3-x &> 0 \quad (C) \\ (3-x)(4-y)-(x+y)^2 &> 0 \quad (A) \\ -x((3-x)(4-y)-(x+y)^2)-(4-y) &> 0 \quad (B) \end{aligned}$$

The feasible set of x and y is shown in Figure 1.

We have seen that the feasible sets of SDPs are convex and semialgebraic. A natural question is therefore whether every convex semialgebraic set is representable by an SDP. Without using additional variables this is not always possible, as there are nontrivial obstructions that must be avoided. We refer the reader to the recent result by Helton and Vinnikov [27] for a complete answer in the two-dimensional case. The results in Section 7 can be used to *constructively* build arbitrarily close approximations.

2.3 Other Classes of Optimization Problems

Semidefinite programs define a fairly wide class of optimization problems. However, it is sometimes convenient to further refine the classification. This leads to significant computational benefits, since some subclasses of SDP possess highly efficient specialized algorithms. This also provides analytical benefits, since in this case further algebraic or geometric properties may provide additional insight (for instance, the polyhedrality of LP).

The language of *conic programming* [39] provides a convenient framework to analyze these issues. By posing an abstract optimization problem of the form

$$\begin{aligned} \text{minimize} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \mathcal{A}X = b, \\ & X \in \mathcal{K}, \end{aligned}$$

where X is an element of a vector space V, and $\mathcal{A} : V \to \mathbb{R}^m$ is a linear operator and $\mathcal{K}$ is a given proper cone, a whole collection of different problems can be analyzed simultaneously. Among the classes of problems that can be interpreted as particular cases of the general conic formulation we have *linear programs*, *second-order cone programs* (SOCP), and SDP, when we take the cone $\mathcal{K}$ to be the nonnegative orthant $\mathbb{R}^n_+$, the second order cone in n variables, or the PSD cone $\mathbb{S}^n_+$. We have then the following natural inclusion relationship among the different optimization classes.

$$\text{LP} \quad \subseteq \quad \text{SOCP} \quad \subseteq \quad \text{SDP}.$$

For numerical efficiency reasons, we want to formulate and solve a given problem in as simplified a setting as possible.

Second-order cone programs partially enjoy the expressive modeling power that non-polyhedral cones such as the PSD cone have, but at the same time share with LP the scalability properties necessary for solving large-scale instances (of the order of tens of thousands of variables), that are currently out of reach for SDP. More details on the theory and applications of second-order cone programming can be found in [39, 34, 2].

We are confident that the application areas where SOCP techniques are used will increase dramatically in the future, either as the natural modeling framework, or as a computational efficient device to approximate an underlying more complicated set, such as the SDP cone.

The case of optimization problems on the cone of sum of squares (SOS) polynomials, to be discussed in Section 7, has attracted lately the attention of several researchers. From a strictly mathematical viewpoint it is equivalent to SDP; SDP corresponds to nonnegativity conditions on quadratic forms, which coincide with the SOS forms, and conversely, we can reformulate SOS programs as SDPs. Even though the possibilities of numerically exploiting the structure of SOS programs is not yet fully understood, there are at least a couple of reasons why it is convenient to think of them as a distinguished category of problems; the richness of the underlying algebraic structure, as well their convenient modeling flexibility.

Yet another interesting class, that only in the last few years has begun to be explored, is that of optimization problems involving *hyperbolic polynomials* and their associated cones [24, 5], which are of great theoretical interest. Research in this area is still at its early stage, with some open fundamental mathematical questions, and practical applications still under investigation.

3 Certificates and duality

3.1 Duality for Optimization Problems

If we have an optimization in the form (2), there is a useful and important duality theory. Define the ***Lagrangian function*** $L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ by

$$L(x, \lambda) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) \qquad \text{for all } x \in \mathbb{R}^n, \lambda \in \mathbb{R}^m.$$

and define the set D by

$$D = \left\{ \lambda \in \mathbb{R}^m \;\middle|\; \lambda \geq 0, \inf_{x \in \mathbb{R}^n} L(x, \lambda) \text{ is finite} \right\}.$$

Then define the ***Lagrange dual function*** $g : D \to \mathbb{R}$ by

$$g(\lambda) = \inf_{x \in \mathbb{R}^n} L(x, \lambda) \qquad \text{for all } \lambda \in D,$$

and the ***dual problem*** as

$$\begin{array}{ll} \text{maximize} & g(\lambda) \\ \text{subject to} & \lambda \in D. \end{array} \tag{5}$$

A point $\lambda \in D$ is called ***dual feasible***. The importance of the dual problem is that, given any dual feasible λ, we have a lower bound on the set of achievable values of the primal problem. That is, if x is primal feasible, and λ is dual feasible, we have

$$g(\lambda) \leq f_0(x).$$

This is known as *weak duality*, and follows because, for any $\lambda \in D$ and $x \in P$, we have

$$\begin{aligned} g(\lambda) &\leq L(x, \lambda) \\ &= f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) \\ &\leq f_0(x). \end{aligned}$$

There are also important cases under which the optimal value of the dual problem (5) is equal to the optimal value of the primal problem (2), called *strong duality*. This holds if the functions $f_0, \ldots, f_m$ are convex, and further standard conditions, called *constraint qualifications* hold. An example of a typical constraint qualification is that if there exists $x \in \mathbb{R}^n$ which is strictly feasible, that is, $f_i(x) < 0$ for all $i = 1, \ldots, m$, then strong duality holds. Note that we assume that the functions f_i are defined on all $\mathbb{R}^n$.

3.2 Duality for Feasibility Problems

For feasibility problems, the Lagrange dual function is

$$g(\lambda) = \inf_{x \in \mathbb{R}^n} \sum_{i=1}^{m} \lambda_i f_i(x) \qquad \text{for all } \lambda \in \mathbb{R}^m,$$

and the function g is assigned the value $-\infty$ if the minimization above is unbounded below. We define the set

$$D = \Big\{ \lambda \in \mathbb{R}^m \;\Big|\; \lambda \geq 0,\ g(\lambda) > 0 \Big\}. \tag{6}$$

This gives the *theorem of alternatives*, which states that at most one of P and D is feasible. To show this, suppose P is non-empty, and $x \in P$. Then we have, for all $\lambda \in \mathbb{R}^m$,

$$\begin{aligned} g(\lambda) &\leq \sum_{i=1}^{m} \lambda_i f_i(x) \\ &\leq 0, \end{aligned}$$

and so D is empty. The primal and dual feasibility problems defined by P and D are called *weak alternatives*. Note that it may happen that both P and D are infeasible. Again, if the functions $f_1, \ldots, f_m$ are convex, then under suitable constraint qualifications the sets P and D are *strong alternatives*; exactly one of P or D is non-empty.

Certificates of infeasibility. The theorem of alternatives above, and its generalizations discussed below, are of great importance. For many practical problems, one would like to develop an algorithm to find a feasible point x. Once such an x is found, verifying that it satisfies the desired conditions is straightforward, since one may evaluate the functions $f_1, \ldots, f_m$ at x and check that they are all negative. However, if a feasible point x cannot be found, one would like to demonstrate that *no solution exists*, and that the problem specification must be altered. The dual feasibility problem gives precisely this; if instead one can find a dual feasible point, then this provides a *certificate* that no solution to the primal system of inequalities exists. We emphasize that duality is a property by which infeasibility (or suboptimality) can be *concisely certified.*

3.3 Duality in Semidefinite Programming

While we will not delve into the details here, it is possible to extend the duality theory described in the previous subsection to the case of generalized inequalities. Since SDP naturally falls within this framework, it is not surprising then that semidefinite programs, being convex optimization problems, enjoy a rich duality structure.

The *standard dual* associated with the primal program (3) is also a semidefinite program, given by

$$\begin{array}{ll} \text{maximize} & b^T y \\ \text{subject to} & \displaystyle\sum_{i=1}^{m} y_i A_i \preceq C, \end{array} \tag{7}$$

where the dual variable $y \in \mathbb{R}^m$. As above we have weak duality; any feasible solution of the dual provides a lower bound on the achievable values of the primal; conversely, feasible primal solutions give upper bounds on dual solutions. For SDPs, this may be shown directly, since for any primal feasible X and dual feasible y we have that the difference between the two objective functions satisfies

$$\begin{aligned} \operatorname{trace}(CX) - b^T y &= \operatorname{trace}(CX) - \sum_{i=1}^{m} y_i \operatorname{trace}(A_i X) \\ &= \operatorname{trace}\Big(\Big(C - \sum_{i=1}^{m} y_i A_i\Big) X\Big) \\ &\geq 0, \end{aligned}$$

with the last inequality being true because of self-duality of the PSD cone. The following result expresses a strong duality property possessed by semidefinite programs.

Theorem 1. *Consider the primal-dual SDP pair* (3) *and* (7). *If either feasible set has has a nonempty interior, then for every* $\epsilon > 0$, *there exist feasible* X, y *such that*

$$\operatorname{trace}(C^T X) - b^T y < \epsilon.$$

Furthermore, if both feasible sets have nonempty interiors, then the optimal solutions are achieved by some $X_\star$, $y_\star$.

4 Semidefinite Programming and Control

There has been extensive application of SDP techniques for analysis and synthesis for control systems, and the field is still experiencing a continuous growth. Significant changes have resulted from this, one of the most important being a radical change in what it means to have *solved* a control problem. Historically, control problems have been considered solved when they have been reduced to methods computable with the technology of the time; at various points in history, this has meant a reduction to equations soluble via graphical plots (such as root-locus), and a reduction to linear algebraic systems, such as Riccati equations. Nowadays it is almost universally accepted that the reduction of a question to an easily tractable convex optimization problem (in particular, a semidefinite program) constitutes a solution.

It is also widely accepted that matrix inequalities and duality are inextricably linked with the study of linear control systems; this is witnessed by the fundamental role played by the Lyapunov and Riccati inequalities [63, 4]. Very significant work by many researchers has continuously expanded these boundaries; robustness analysis techniques based on quadratic stability, multipliers, structured singular values, or integral quadratic constraints (IQCs), as well as the corresponding synthesis methodologies, have dealt with uncertainties and nonlinearities, and can all be cast under the convenient computational machinery of matrix inequalities. Given our purposes, we will not go here into a detailed survey of the many available results, but instead refer the reader to the extensive available literature, a good start being the general reference works mentioned in the introduction. Also, as a representative example that illustrates several of the now standard techniques available, we present next a simple case of multiobjective design.

Consider the linear system

$$\begin{aligned}\dot{x}(t) &= Ax(t) + B_z z(t) + B_u u(t)\\ w(t) &= Cx(t).\end{aligned}$$

We are interested in designing a state-feedback controller $u = Kx$ that minimizes the $\mathcal{H}_2$ norm of the transfer function T_{zw} from the input z to the output w. Additionally, we would like the poles of the closed-loop system to be located in a specific region of the complex plane, described by the feasible set of the linear matrix inequality

$$\mathcal{P} = \left\{ z \mid P + Rz + R^T\bar{z} \succ 0 \right\}, \quad P \in \mathbb{S}^k, R \in \mathbb{R}^{k\times k},$$

where the inequality should be interpreted as forcing the Hermitian matrix on the left-hand side to be positive semidefinite. It is well known [10, 15, 53] that the two desired design specifications $\|T_{zw}\|_2 \leq \gamma$ and $\sigma(A + B_uK) \subset \mathcal{P}$ can be expressed as the SDP conditions

$$\operatorname{trace}(Z) \leq \gamma^2, \qquad \begin{bmatrix} A_{cl}X_1 + X_1A_{cl}^T & X_1C^T \\ CX_1 & -I\end{bmatrix} \preceq 0, \qquad \begin{bmatrix} X_1 & B_z \\ B_z^T & Z\end{bmatrix} \succeq 0,$$

and

$$P \otimes X_2 + R \otimes A_{cl}X_2 + R^T \otimes X_2A_{cl}^T \succ 0, \qquad X_2 \succ 0,$$

respectively, where $A_{cl} = A + B_uK$ is the closed-loop state matrix, and $\otimes$ is the Kronecker product.

The conditions here are not yet affine in the decision variables, since there are bilinear terms such as KX_2. To convert this into a tractable synthesis problem, a (possibly conservative) approach is to assume $X_1 = X_2 = X$, and then introduce the variable $Y = KX$, so $A_{cl}X_1 = A_{cl}X_2 = AX + B_uY$. The system to be solved reduces now to

$$\begin{array}{ll} \text{minimize} & \operatorname{trace}(Z) \\ \text{subject to} & 0 \succeq \begin{bmatrix} (AX + B_uY) + (AX + B_uY)^T & XC^T \\ CX & -I \end{bmatrix} \\ & 0 \preceq \begin{bmatrix} X & B_z \\ B_z^T & Z \end{bmatrix} \\ & 0 \prec P \otimes X + R \otimes (AX + B_uY) + R^T \otimes (AX + B_uY)^T \\ & 0 \prec X \end{array}$$

which is a *bona-fide* semidefinite program. After solving it numerically, the controller can be recovered from the solution using $K = YX^{-1}$.

As a simple illustration, assume the numerical data

$$A = \begin{bmatrix} 2 & 1 \\ -1 & 4 \end{bmatrix}, \quad B_u = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad B_z = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 \end{bmatrix},$$

and the region in the complex plane defined by $|z+2| < 1$, i.e.,

$$P = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

Ignoring the constraint on the closed-loop pole locations, the optimal controller achieves a value of $(\operatorname{trace} Z)^{\frac{1}{2}} = \|T_{zw}\|_2$ arbitrarily close to $\sqrt{10} \approx 3.16$, but the gains K grow unboundedly, as does the magnitude of one of the poles. Imposing now the left-over constraint, the optimal solution achieves a value of $(\operatorname{trace} Z)^{\frac{1}{2}} \approx 21.26$. This is only an upper bound on the $\mathcal{H}_2$ norm; the actual achieved value is $\|T_{zw}\|_2 = 7.57$, with the final controller being $K = [-9.92, 35]$, and closed-loop poles at $-1.96 \pm 0.66\jmath$.

Several improved modifications of these methods are also available, including better ways of handling the requirement on a common Lyapunov function, and extensions to multiobjective output feedback; once again, we refer the reader to the cited bibliography for the details.

5 Combinatorial Optimization

Many well-known problems in linear control theory have the property that they can be *exactly* reformulated in terms of semidefinite programs, although in some cases the required transformations can be far from obvious. When attempting to extend convexity-based methods to larger classes of problems, a complete characterization of the solution set via a simple SDP may no longer be possible. The difficulties arise in two different (though related) fronts. On the one hand, it may be computationally difficult to find feasible points, that show that the problem indeed has a solution; on the other hand, there may not exist a *concise* way of demonstrating infeasibility. There is a clear interdependence of these two issues, since if the first question were always easy, then the second would follow, but they are not equivalent. For the case of feasibility problems defined by convex functions, the first property is satisfied since we can efficiently search for solutions. As we have seen, convex duality gives us a useful way of certifying the nonexistence of solutions. After introducing the formal machinery of a computational model and polynomial-time algorithms to make their meaning more precise, these questions play an essential role in some of the central issues in computational complexity; whether P=NP and NP=co-NP, respectively [20].

Note also that convexity properties by themselves do not automatically imply polynomial-time solvability. The specific *representation* of the feasible set plays a crucial role here, and the existing results require the availability of either subgradients [55], or a self-concordant barrier function [39]. There are many examples of optimization problems over (possibly implicitly given) convex sets, where all these operations, or even checking membership, are computationally hard.

Computational hardness arises in many cases as a direct consequence of the combinatorial nature of the problem. When this happens, a class of approximate methods, usually called *convex relaxations*, are typically used to either bound the achievable optimal values, or obtain reasonably good candidate solutions. While relaxations based on linear programming have been and still are extensively used in many application areas such as integer programming, in the last few years a vast collection of new and powerful SDP-based relaxations have attracted the attention of many researchers.

An important and well-known example where these SDP relaxations are extensively used is boolean programming. Consider the NP-hard problem of minimizing a quadratic form over the discrete set given by the vertices of a hypercube, i.e.,

$$\begin{array}{ll} \text{minimize} & x^T Q x \\ \text{subject to} & x_i \in \{+1, -1\} \quad \text{for all } i = 1, \ldots, n \end{array} \tag{8}$$

where $Q \in \mathbb{S}^n$. Notice that the feasible set is discrete and has 2^n points, so convexity-based techniques would seem not to be applicable. However, a lower bound on the optimal solution can be found by solving the primal-dual pair of SDPs given by

$$\begin{array}{ll} \text{minimize} & \operatorname{trace}(QX) \\ \text{subject to} & X \succeq 0 \\ & X_{ii} = 1 \quad \text{for all } i = 1, \ldots, n \end{array} \tag{9}$$

and its dual

$$\begin{array}{ll} \text{maximize} & \operatorname{trace}(\Lambda) \\ \text{subject to} & \Lambda \text{ diagonal} \\ & Q - \Lambda \succeq 0 \end{array} \tag{10}$$

This pair of SDPs have several possible interpretations, extensively discussed in the literature. First notice, that, on the primal side, letting $X = xx^T$, we can equivalently rewrite the original objective function as $x^T Q x = \operatorname{trace}(Qxx^T) = \operatorname{trace} QX$. The feasible set is given now by $X \succeq 0$, $X_{ii} = 1$, $\operatorname{rank}(X) = 1$. Dropping the rank constraint, we obtain directly the SDP (9).

Another interpretation is given from the dual SDP; an argument based on Lagrangian duality applied to the constraints $x_i^2 - 1 = 0$ is enough. We have

$$L(x, \lambda) = x^T Q x - \sum_i \lambda_i (x_i^2 - 1) = x^T (Q - D) x + \operatorname{trace} \Lambda.$$

For L to be bounded below, the condition $Q - D \succeq 0$ is required, and therefore the SDP (10) directly follows.

This relaxation can also be interpreted as a specific case of the so-called S-procedure; see [10] and the references therein. Several important results are available regarding the performance of these relaxations; the celebrated Goemans-Williamson approximation scheme for MAXCUT [23] relies exactly on this relaxation, followed by a randomized rounding step. For many other related problems, it has been possible to prove *a priori* approximation guarantees.

Combinatorial Optimization and Control. An example of combinatorial optimization techniques applied to control is as follows. Consider the problem of designing an optimal open-loop input for the single-input discrete-time system

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned}$$

for $t = 0, \ldots, N$, where the one would like to minimize the ℓ_2 criterion $q(u) = \|y(t) - y_r(t)\|^2$, with y_r being the desired reference output trajectory, and the input $u(t)$ constrained by $|u(t)| = 1$ for all $t = 0, \ldots, N$. In other words, we have an open-loop LQR-type problem, but where the input is *bang-bang*, taking only two possible values.

It is clear that a simple lower bound on the optimal value can be obtained from the solution of an associated unconstrained LQR problem. We are interested in computing better, less conservative lower bounds on the achievable value of $q(u)$. We can formulate the search for such a lower bound as an SDP. For notational simplicity, let $q(u) = u^T Q u + 2r^T u + s$, where the expressions of Q, r, s can be easily obtained from those of A, B, C and y_r. We are thus interested in bounding the value of the following optimization problem.

$$
\begin{array}{ll}
\text{minimize} & \begin{bmatrix} u \\ 1 \end{bmatrix}^T \begin{bmatrix} Q & r \\ r^T & s \end{bmatrix} \begin{bmatrix} u \\ 1 \end{bmatrix} \\
\text{subject to} & u_i \in \{+1, -1\} \text{ for all } i
\end{array}
$$

Let q^* denote the optimal value of this problem. Similar to the one described earlier, a simple SDP bound on the optimal value can be obtained as follows. Let q_* be the optimal value of

$$
\begin{array}{ll}
\text{maximize} & \operatorname{trace}(D) \\
\text{subject to} & \begin{bmatrix} Q - D & r \\ r^T & s \end{bmatrix} \succeq 0,
\end{array}
$$

where D is a diagonal matrix. Clearly $q_* \leq q^*$, as can easily be seen by multiplying the matrix inequality left and right by $\begin{bmatrix} u^T & 1 \end{bmatrix}$ and its transpose. Notice also that $q_* \geq 0$, since $D = 0$ is always a feasible point.

6 Robust Optimization

An important development in recent years in the field of robust optimization has been *robust semidefinite programming*. Here one has a parametrized family of linear matrix inequalities, and one would like to find a point which is simultaneously feasible for the whole family.

We give here an example from finite-horizon control, which may be used, for example, in receding horizon control. We have a linear dynamical system

$$
\begin{aligned}
x(t+1) &= Ax(t) + B_w w(t) + B_u u(t) \\
y(t) &= Cx(t)
\end{aligned}
\tag{11}
$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^{n_u}$ and $w(t) \in \mathbb{R}^{n_w}$ for all $t = 0, \ldots, N$, and the system evolves on the discrete-time interval $[0, N]$. We would like to design an input signal $u(0), \ldots, u(N-1)$ so that the system output y tracks a desired signal y_{des}. Define

$$
u = \begin{bmatrix} u(0) \\ \vdots \\ u(N-1) \end{bmatrix} \qquad w = \begin{bmatrix} w(0) \\ \vdots \\ w(N-1) \end{bmatrix} \qquad y = \begin{bmatrix} y(0) \\ \vdots \\ y(N) \end{bmatrix} \qquad y_{\mathrm{des}} = \begin{bmatrix} y_{\mathrm{des}}(0) \\ \vdots \\ y_{\mathrm{des}}(N) \end{bmatrix}.
$$

Suppose the set of disturbances w is given by

$$
W = \left\{ w \in \mathbb{R}^{N n_w} \mid \|w(t)\|_2 \leq w_{\max} \text{ for all } t = 0, \ldots, N-1 \right\}.
$$

In this way the disturbances are specified so that at each time t the vector $w(t)$ lies within a Euclidean ball. Such sets, and generalizations thereof, are called *ellipsoidal uncertainty* in [6].

We would like to find an input sequence u which solves

$$
\min_u \max_{w \in W} \|y - y_{\mathrm{des}}\|_2
$$

As standard, one may construct block-Toeplitz matrices T and S such that $y = Tu + Sw$ for all u, y satisfying the dynamics (11), and so we can write this problem as

$$\min_{u} \max_{w \in W} \|Tu + Sw - y_{\text{des}}\|_2 \tag{12}$$

This problem is therefore a ***robust least-squares*** problem; recent work [17] in this area has produced solutions using semidefinite programming for this and more general versions, and certain classes of this problem may be solved using second-order cone programming [34].

Let $s_1, \ldots, s_{Nn_w}$ be the columns of S, and define

$$F_0(u) = \begin{bmatrix} t & (Tu - y_{\text{des}})^T \\ Tu - y_{\text{des}} & I \end{bmatrix} \qquad F_i(u) = \begin{bmatrix} 0 & s_i^T \\ s_i & 0 \end{bmatrix} \text{ for all } i = 1, \ldots, Nn_w.$$

Then problem (12) is equivalent to the robust semidefinite program

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & F_0(u) + \displaystyle\sum_{i=1}^{Nn_w} w_i F_i(u) \succeq 0 \text{ for all } w \in W, \end{array} \tag{13}$$

where w_i is the i'th (scalar) element of w. An upper bound on the solution of this robust SDP, along with an input u that achieves it, may be found by solving the following SDP in the variables $t, u, P_1, \ldots, P_N, Q_1, \ldots, Q_N$.

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \begin{bmatrix} P_i & F_{(i-1)n_w+1} & \cdots & F_{in_w-1} \\ F_{(i-1)n_w+1} & Q_i & & \\ \vdots & & \ddots & \\ F_{in_w-1} & & & Q_i \end{bmatrix} \succeq 0 \text{ for all } i = 1, \ldots, N \\ & \begin{bmatrix} t & (Tu - y_{\text{des}})^T \\ Tu - y_{\text{des}} & I \end{bmatrix} - \dfrac{1}{2} \displaystyle\sum_{i=1}^{N} (P_i + Q_i) \succeq 0. \end{array} \tag{14}$$

Given a feasible solution, we have

$$\max_{w \in W} \|Tu + Sw - y_{\text{des}}\|_2 \le t^{\frac{1}{2}}.$$

The particular problem shown above is of course just an example of the application of the robust SDP approach to control. Many other problems for which one has synthesis conditions in terms of linear matrix inequalities may be similarly generalized to construct robust synthesis conditions.

7 Sum of Squares Methods

Many optimization problems have feasible sets specified by polynomial equations and inequalities. This includes the feasible set of a semidefinite program, as well as many more general classes. Finite integer constraints can be easily expressed as polynomial equations; for example, $x \in \{0, 1\}$ is equivalent to the constraint equation $x(x-1) = 0$.

Specifically, a (basic) ***semialgebraic set*** is a subset of $\mathbb{R}^n$ defined by finite number of polynomial equations and inequalities. For example, the subset of $\mathbb{R}^2$ defined by

$$\{ (x_1, x_2) \in \mathbb{R}^2 \mid x_1^2 + x_2^2 \le 1, \ x_1^3 - x_2 \le 0 \}$$

is a semialgebraic set. Given a set of polynomials specifying such a set, one would like to find either a feasible point or a certificate of infeasibility. Clearly, a semialgebraic set need not be convex, and in general, testing feasibility of such sets is intractable.

Recent developments have led to an approach using semidefinite programming to test feasibility of semialgebraic sets [43, 44, 45] (see also [32] for a dual approach, and [14]). By solving a semidefinite program, one may obtain a *certificate of infeasibility* for an infeasible semialgebraic set. The size of this certificate (i.e., the size of the SDP to be solved) may depend on the specific set. One of the appealing features of this approach is that there is a hierarchy of such certificates, allowing one to solve a sequence of semidefinite programs in an attempt to prove infeasibility of a semialgebraic set.

The simplest case is when we are concerned with a set defined by a single polynomial inequality, as follows. Given $f \in \mathbb{R}[x_1, \ldots, x_n]$, we would like to know if there exists $x \in \mathbb{R}^n$ such that $f(x) < 0$. If there does not exist such an x, we have $f(x) \geq 0$ for all x and f is called *positive semidefinite*, or PSD.

An example is as follows. Suppose f is given by

$$f(x) = 4x_1^4 + 4x_1^3x_2 - 7x_1^2x_2^2 - 2x_1x_2^3 + 10x_2^4,$$

and we would like to determine the feasibility of the set

$$P = \{\, x \in \mathbb{R}^2 \mid f(x) < 0 \,\}.$$

Assume now that there exists a matrix $Q \in \mathbb{S}^3$ such that

$$\begin{aligned} f(x) &= \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \end{bmatrix}^T \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{bmatrix} \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \end{bmatrix} \\ &= z^T(x)Qz(x), \end{aligned} \tag{15}$$

In general, there are many such Q, since this is just a system of linear equations in the entries of Q, as can be seen by matching coefficients on the right- and the left-hand sides. However, an important consequence is the following; *if* we can find a PSD Q that satisfies (15), then $f(x)$ would have to take only nonnegative values, so one immediately has a proof that there cannot exist a real $x \in P$ and the matrix Q is then a certificate of infeasibility of the semialgebraic set P.

In the example above, we can parametrize the set of possible Q by

$$f(x) = \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \end{bmatrix}^T \begin{bmatrix} 4 & -\lambda & 2 \\ -\lambda & 10 & -1 \\ 2 & -1 & -7+2\lambda \end{bmatrix} \begin{bmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \end{bmatrix}$$

for all $\lambda \in \mathbb{R}$, and therefore if there exists $\lambda \in \mathbb{R}$ such that

$$\begin{bmatrix} 4 & -\lambda & 2 \\ -\lambda & 10 & -1 \\ 2 & -1 & -7+2\lambda \end{bmatrix} \succeq 0,$$

then P is infeasible. This is a semidefinite program. In this case, picking $\lambda = 6$ gives a positive semidefinite matrix, and therefore a valid proof of the infeasibility of the set P.

This approach is quite general. For every polynomial of degree d in n variables $f \in \mathbb{R}[x_1, \ldots, x_n]$, we form the vector of monomials $z(x)$ which has as elements all monomials of degree $d/2$ or less. In fact, fewer monomials may be included, which gives a reduced-size SDP, depending on the particular polynomial f. The set of matrices Q for which $f(x) = z(x)^TQz(x)$ for all x is an affine set, and is simple to construct.

The above condition can be interpreted as follows. If there exists a positive semidefinite matrix Q such that $f(x) = z(x)^T Q z(x)$, then we may factorize Q. Every positive semidefinite matrix Q may be factorized as $Q = VV^T$, where $V \in \mathbb{R}^{n \times r}$ is a real matrix, and r is the rank of Q. With $v_1, \ldots, v_r$ the columns of V, we have

$$\begin{aligned} f(x) &= z(x)^T Q z(x) \\ &= \sum_{i=1}^{r} \left(v_i^T z(x)\right)^2 \end{aligned}$$

and in this case we may choose a factorization with

$$V = \begin{bmatrix} 0 & 2 \\ 1 & -3 \\ 2 & 1 \end{bmatrix},$$

giving

$$f(x) = (x_2^2 + 2x_1x_2)^2 + (2x_1^2 - 3x_2^2 + x_1x_2)^2$$

which is a *sum-of-squares* (SOS) decomposition of f ([46, 51]). From this decomposition, it is immediately clear that there cannot exist a feasible x, and the decomposition (characterized by Q) is clearly a certificate of infeasibility. Hence we may decide whether a polynomial is a sum-of-squares by solving a semidefinite program.

The natural question to ask, when P infeasible, is whether there always exists such a certificate of infeasibility. Expressed another way, is every PSD polynomial a sum of squares? It was shown by Hilbert in 1888 that in general this is not the case; there exist polynomials which are PSD but not SOS. This is to be expected, since it is known that testing feasibility of a semialgebraic set (even if it is defined by just one polynomial) is intractable; but if every PSD polynomial was SOS then feasibility would be testable using a simple semidefinite program. An example of a polynomial which is PSD, but not SOS, is the well-known Motzkin form

$$M(x, y, z) = x^4y^2 + x^2y^4 + z^6 - 3x^2y^2z^2.$$

Besides the more general methods in the next section, there is a further result that gives a method for bridging the gap between SOS and PSD polynomials. It was shown by Reznick [50] that if $f(x) > 0$ for all $x \in \mathbb{R}^n$, then there exists some r such that

$$f(x)\left(\sum_{i=1}^{n} x_i^2\right)^r$$

is a sum-of-squares. The coefficients of this polynomial are affine functions of the coefficients of f, and so for each r one may test whether this product is SOS by solving a semidefinite program. This gives a *sequence* of semidefinite programs, of growing dimension, which may be used to test whether a given polynomial is positive definite.

An important remaining question is how to find feasible points x when P is feasible. This is a distinct question from finding a certificate of infeasibility; however the dual problem to the above SDP can be used in certain cases to do this.

7.1 Generalized Duality and the Positivstellensatz

We would now like to consider how to certify infeasibility for semialgebraic sets defined by multiple polynomial inequalities, that is sets P of the form

$$P = \Big\{ x \in \mathbb{R}^n \mid f_i(x) \geq 0 \text{ for all } i = 1, \ldots, m \Big\}, \tag{16}$$

where the functions f_i are polynomials on $\mathbb{R}^n$. For convex feasibility problems which satisfy a constraint qualification, the theorem of alternatives provides a necessary and sufficient condition for feasibility of the primal. However, for semialgebraic sets, the feasibility problems are defined by non-convex functions $f_1, \ldots, f_m$. In this case, the following stronger result is very useful. Define the set of SOS polynomials on $\mathbb{R}^n$ by $\mathcal{P}^n_{\text{sos}}$ by

$$\mathcal{P}^n_{\text{sos}} = \left\{ s \in \mathbb{R}[x_1, \ldots, x_n] \;\middle|\; s \text{ is a sum-of-squares} \right\}.$$

Define the functional $G : \mathcal{P}^n_{\text{sos}} \to \mathbb{R}$ by

$$G(s_1, \ldots, s_m) = \sup_{x \in \mathbb{R}^n} \sum_{i=1}^{n} s_i(x) f_i(x) \qquad \text{for all } s_1, \ldots, s_m \in \mathcal{P}^n_{\text{sos}}$$

Then define the dual set of functions

$$D_1 = \left\{ (s_1, \ldots, s_m) \;\middle|\; s_i \in \mathcal{P}^n_{\text{sos}},\ G(s_1, \ldots, s_m) < 0 \right\}$$

Then P and D_1 are weak alternatives; at most one of P and D_1 is non-empty. To see this, suppose that P is feasible, and $x \in P$. Then, for all $s_1, \ldots, s_m \in \mathcal{P}^n_{\text{sos}}$, we have

$$\begin{aligned} G(s_1, \ldots, s_m) &\geq \sum_{i=1}^{m} s_i(x) f_i(x) \\ &\geq 0 \end{aligned}$$

and hence D_1 is empty. This refutation is a simple form of weak duality.

The Positivstellensatz. A slight modification of this result is as follows. If there exist sum-of-squares polynomials $s_0, s_1, \ldots, s_m$ such that

$$\sum_{i=1}^{n} s_i(x) f_i(x) + s_0(x) + 1 = 0 \tag{17}$$

then the set P is infeasible. This sufficient condition for infeasibility of P may be tested using semidefinite programming. One picks a fixed degree d over which to search for SOS polynomials $s_1, \ldots, s_m$, satisfying this condition. The decision variables are the coefficients of the polynomials and the constraints that the s_i be SOS are imposed as positive semidefiniteness constraints. Clearly, if we can find a set of functions $s_1, \ldots, s_m$ satisfying (17) then the set P is infeasible. A similar, but stronger condition is

$$\sum_{i=1}^{n} s_i(x) f_i(x) + \sum_{i=1}^{n} \sum_{j=1}^{n} t_{ij}(x) f_i(x) f_j(x) + s_0(x) + 1 = 0 \tag{18}$$

which again may be tested via SDP. Refutations of this form have very strong duality properties. It can be shown that, by allowing for unrestricted degrees of the polynomials s_i and t_{ij}, and arbitrary products of f_i, one may always construct a refutation for any given infeasible semialgebraic set P using SOS polynomials [8]. No assumptions whatsoever on the polynomials f_i are required. This result is called the *Positivstellensatz.* Software for testing feasibility of semialgebraic sets using the above methods is available in the form of a Matlab toolbox, called SOSTOOLS [47].

Applications. Testing feasibility of semialgebraic sets has important applications in control and combinatorial optimization. For example, the integer program of (8) can be formulated as

$$\begin{array}{rl} \text{minimize} & t \\ \text{subject to} & x^T Q x \leq t \\ & x_i^2 - 1 = 0 \text{ for all } i = 1, \dots, n \end{array} \tag{19}$$

hence we check feasibility for a fixed t using the above Positivstellensatz approach. In fact, the relaxation discussed in that section exactly coincides with that obtained using the first level of the general approach just presented.

SOS and Lyapunov functions. The possibility of reformulating conditions for a polynomial to be a sum-of-squares as an SDP is very useful, since we can use the SOS property in a control context as a convenient sufficient condition for polynomial nonnegativity. Recent work has applied the sum-of-squares approach to the problem of finding a Lyapunov function for nonlinear systems [43, 41, 57]. This approach allows one to search over affinely parametrized polynomial or rational Lyapunov functions for systems with dynamics of the form

$$\dot{x}_i(t) = f_i(x(t)) \qquad \text{for all } i = 1, \dots, n$$

where the functions f_i are polynomials or rational functions. Then the condition that the Lyapunov function be positive, and that its Lie derivative be negative, are both directly imposed as sum-of-squares constraints in terms of the coefficients of the Lyapunov function.

As an example, we consider the following system, suggested by M. Krstić.

$$\begin{aligned} \dot{x} &= -x + (1 + x)y \\ \dot{y} &= -(1 + x)x. \end{aligned}$$

Using SOSTOOLS [47] we easily find a quartic polynomial Lyapunov function, which after rounding (for purely cosmetic reasons) is given by

$$V(x, y) = 6x^2 - 2xy + 8y^2 - 2y^3 + 3x^4 + 6x^2y^2 + 3y^4.$$

It can be readily verified that both $V(x, y)$ and $(-\dot{V}(x, y))$ are SOS, since

$$V = \begin{bmatrix} x \\ y \\ x^2 \\ xy \\ y^2 \end{bmatrix}^T \begin{bmatrix} 6 & -1 & 0 & 0 & 0 \\ -1 & 8 & 0 & 0 & -1 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 \\ 0 & -1 & 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \\ x^2 \\ xy \\ y^2 \end{bmatrix}, \quad -\dot{V} = \begin{bmatrix} x \\ y \\ x^2 \\ xy \end{bmatrix}^T \begin{bmatrix} 10 & 1 & -1 & 1 \\ 1 & 2 & 1 & -2 \\ -1 & 1 & 12 & 0 \\ 1 & -2 & 0 & 6 \end{bmatrix} \begin{bmatrix} x \\ y \\ x^2 \\ xy \end{bmatrix},$$

and the matrices in the expression above are positive definite. Similar approaches may also be used for finding Lyapunov functionals for certain classes of hybrid systems.

8 Exploiting Control-Specific Structure

The SDP problems that arise from the analysis and synthesis of linear control systems, usually have a very specific structure, that should be exploited to achieve the best possible computational efficiency. In particular, the ubiquity of Lyapunov-style terms of the form $A^T P + PA$ (or $A^T PA - P$, in the discrete case) suggest that generic implementations using the SDP standard forms described in (3) and (7) will not be optimal, unless the extra structure is somehow taken into account.

An important subclass for which several customized algorithms are already available is that of optimization problems with an structure induced by the Kalman-Yakubovich-Popov lemma (see [63, 49] and the references therein). This fundamental result establishes the equivalence between a frequency domain inequality and the feasibility of a particular SDP. It is an important generalization of classical linear control results, such as the bounded real and positive real lemmas, and the cornerstone of several analysis and synthesis results.

The harder direction of the KYP lemma, that the frequency domain inequality implies the existence of a storage function, can be interpreted as a *lossless* property of an associated SDP relaxation, much in the style of the ones presented in Section 5. Yet another related reformulation of the KYP lemma is given by the equivalence of the structured singular value μ and its upper bound in the case where there is a full block and one scalar block [40].

In the case of systems with large state dimension n, the KYP approach is not very efficient, since the matrix variable representing the storage function that appears in the LMI has $(n^2 + n)/2$ components, and therefore the computational requirements are quite large, even for medium sized problems.

Several different methods have been proposed recently for the efficient solution of this and related kind of problems. The approaches by Kao *et al.* in [29, 30], Parrilo [42] and Varga and Parrilo [62] all rely on outer approximation methods based on a discretization of the frequency axis, with the former using linear programming and the latter using SDP cuts. The scheme by Hansson and Vandenberghe [25] is based on the interior point machinery, but cleverly exploits the Lyapunov structure at each iteration at the linear algebra level. Rotea and D'Amato exploit several properties of the μ SDP upper bound to obtain significant speed-ups in the computation of the worst-case frequency response [52]. Additionally, methods for taking advantage of autocorrelation structure have also been developed by Alkire and Vandenberghe in [3].

9 Available Implementations

Despite the impressive advances on the theoretical and modeling aspects of SDP, much of its impact in applications has undoubtedly been a direct consequence of the efforts of many researchers in producing and making available good quality software. In this section we give pointers to and discuss briefly some of the current computational tools for effectively formulating and solving SDPs.

9.1 Solvers

From a computational viewpoint, semidefinite programs can be efficiently solved, both in theory and in practice. In the last few years, research on SDP has experienced an explosive growth, particularly in the areas of algorithms and applications. Two of the main reasons for this practical impact are the versatility of the problem formulation, and the availability of high-quality software,

For applications of semidefinite programming to control, the pioneering codes were the MATLAB LMI toolbox [19] and SP [61]. Today there is a wide variety of excellent SDP solvers to choose from. For general-purpose small- and medium-scale problems, interior-point based solvers are probably the best choice, combining good performance and accuracy, primal and dual solutions, as well as reasonable speed-ups depending on the problem sparsity. We mention a few of the best-known ones: SeDuMi [56], SDPT3 [59], SDPA [18], CSDP [9], DSDP [7], among others. Other good pointers to the available SDP solvers are the SDP webpages of C. Helmberg and H. Wolkowicz.

Several approaches other than interior-point methods have also been investigated, a few of them being bundle methods [26, 37], or nonlinear approaches based on special factorizations [13]. This research has increased steadily in the last few years. The codes based on these new developments are the only ones achieving satisfactory performance for some of the very large and structured problems arising from combinatorial optimization, with MAXCUT and the Lovász theta function being prime examples.

A comprehensive benchmarking effort of the performance of several solvers in a representative collection of problems (including some arising from control) has been recently reported by Hans Mittelmann in [38]; an up-to-date version of the results is available at the web page `http://plato.la.asu.edu/bench.html`.

The consensus and experience among researchers, backed up by the hard data mentioned, seems to be indicate that there is no algorithm or code that uniformly outperforms all others. While most software packages will work satisfactorily on different problems, there may be (sometimes significant) differences in the computation times. It is therefore good advice to try different codes at the initial stages of solving a large-scale problem; not only we benefit of the possible speed differences, but can also validate the solutions against each other. The availability of SDP parsers such as the ones described in the following subsection can notably help during this migration process.

9.2 Parsers

The solvers described in the previous subsections usually take as inputs either text files containing a problem description, or directly the matrices (A_i, b, c) corresponding to the standard primal/dual formulation. Needless to say, this is often inconvenient at the initial modeling and solution stages. A natural solution is therefore to formulate the problem in a more neutral description, that can later be automatically translated to fit the requirements of each solver. For generic optimization problems, this has indeed been the approach of much of the operations research community, which has developed some well-known standard file formats, such as MPS, or optimization modeling languages like AMPL and GAMS. An important remark to keep in mind, much more critical in the SDP case than in LP, is the extent to which the problem structure can be signaled to the solver. There is a growing push within the optimization community towards the possibility of adding SDP-oriented extensions to the standard modeling languages mentioned.

In the meantime, however, enterprising researchers in control and related areas have written specific parsers that partially or fully automate the conversion tasks, when used within a problem-solving environment such as MATLAB. Among them we mention the early effort SDPSOL [12], as well as the more recent ones YALMIP [35], SeDuMi Interface [31], and LMILab translator [54], dealing with general SDPs, as well as the more domain-specific IQCbeta [36], Gloptipoly [28], and SOSTOOLS [47].

Any of these parsers can make the task of posing and solving a specific problem a much simpler and enjoyable procedure than manual, error-prone methods. We strongly encourage the reader to take them for a test drive.

References

[1] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optim.*, 5(1):13–51, 1995.

[2] F. Alizadeh and D. Goldfarb. Second order cone programming. Technical Report 51-2001, RUTCOR, Rutgers University, 2001.

[3] B. Alkire and L. Vandenberghe. Convex optimization problems involving finite autocorrelation sequences. *Mathematical Programming Series A*, 93(3):331–359, 2002.

[4] V. Balakrishnan and L. Vandenberghe. Semidefinite programming duality and linear time-invariant systems. *IEEE Transactions on Automatic Control*, AC-48(1):30–41, January 2003.

[5] H. H. Bauschke, O. Güler, A. S. Lewis, and H. S. Sendov. Hyperbolic polynomials and convex analysis. *Canad. J. Math.*, 53(3):470–488, 2001.

[6] A. Ben-Tal, L. El Ghaoui, and A. Nemirovskii. Robustness. In H. Wolkowitz, R. Saigal, and L. Vandenberghe, editors, *Handbook of Semidefinite Programming: Theory, Algorithms and Applications*, pages 138–162. Kluwer, 2000.

[7] S. Benson, Y. Ye, and X. Zhang. *DSDP, A Dual Scaling Algorithm for Positive Semidefinite Programming*. Available from `http://www-unix.mcs.anl.gov/~benson/dsdp/`.

[8] J. Bochnak, M. Coste, and M.-F. Roy. *Real Algebraic Geometry*. Springer, 1998.

[9] B. Borchers. CSDP, A C Library for semidefinite programming. *Optimization Methods and Software*, 11(1):613–623, 1999. Available from `http://www.nmt.edu/~borchers/csdp.html`.

[10] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, 1994.

[11] S. Boyd and L. Vandenberghe. *Convex Optimization*. Preprint, 2002.

[12] S. Boyd and S.-P. Wu. *SDPSOL: A Parser/Solver for Semidefinite Programs with Matrix Structure. User's Guide, Version Beta.* Stanford University, March 1996.

[13] S. Burer and R. D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

[14] G. Chesi, A. Garulli, A. Tesi, and A. Vicino. LMI-based techniques for solving quadratic distance problems. In *Proceedings of the 40th IEEE Conference on Decision and Control*, pages 3587–3592, December 2001.

[15] M. Chilali and P. Gahinet. H_∞ design with pole placement constraints: an LMI approach. *IEEE Transactions on Automatic Control*, 41(3):358–367, 1996.

[16] J. C. Doyle, A. Packard, and K. Zhou. Review of LFTs, LMIs, and μ. In *Proceedings of the 30^{th} IEEE Conference on Decision and Control*, 1991.

[17] L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM Journal On Matrix Analysis and Applications*, 18(4):1035–1064, 1997.

[18] K. Fujisawa, M. Kojima, K. Nakata, and M. Yamashita. *SDPA Semidefinite Programming Algorithm – Version 6.00*, 2002. Available from `http://www.is.titech.ac.jp/~yamashi9/sdpa/`.

[19] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali. *LMI Control Toolbox*. The MathWorks Inc., Natick, Mass., May 1995.

[20] M. R. Garey and D. S. Johnson. *Computers and Intractability: A guide to the theory of NP-completeness.* W. H. Freeman and Company, 1979.

[21] L. El Ghaoui and S.-I. Niculescu, editors. *Advances on Linear Matrix Inequality Methods in Control.* SIAM, 1999.

[22] M. X. Goemans. Semidefinite programming in combinatorial optimization. *Math. Programming*, 79(1-3):143–161, 1997.

[23] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

[24] O. Güler. Hyperbolic polynomials and interior point methods for convex programming. *Math. Oper. Res.*, 22(2):350–377, 1997.

[25] A. Hansson and L. Vandenberghe. Efficient solution of linear matrix inequalities for integral quadratic constraints. In *Proceedings of the 39th IEEE Conference on Decision and Control*, December 2000.

[26] C. Helmberg. *SBmethod - A C++ Implementation of the Spectral Bundle Method*. Available from `http://www-user.tu-chemnitz.de/~helmberg/SBmethod/`.

[27] J.W. Helton and V. Vinnikov. Linear matrix inequality representation of sets. Preprint, March 2002. Available from `http://math.ucsd.edu/~helton/`.

[28] D. Henrion and J.-B. Lasserre. *GloptiPoly - Global Optimization over Polynomials with Matlab and SeDuMi*. Available from `http://www.laas.fr/~henrion/software/gloptipoly/`.

[29] C. Y. Kao, U. Jönsson, and A. Megretski. A cutting plane algorithm for robustness analysis of periodically time-varying systems. In *Proceedings of the American Control Conference*, 1999.

[30] C. Y. Kao, A. Megretski, and U. Jönsson. An algorithm for solving optimization problems involving special frequency dependent LMIs. In *Proceedings of the American Control Conference*, 2000.

[31] Y. Labit, D. Peaucelle, and D. Henrion. SeDuMi Interface 1.02: a tool for solving LMI problems with SeDuMi. In *Proceedings of the CACSD Conference, Glasgow*, September 2002.

[32] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[33] M. Laurent and F. Rendl. Semidefinite programming and integer programming. Technical Report PNA-R0210, CWI, Amsterdam, April 2002.

[34] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284:193–228, November 1998.

[35] J. Löfberg. YALMIP: A Matlab interface to SP, MAXDET and SOCP. Technical Report LiTH-ISY-R-2328, Department of Electrical Engineering, Linköping University, SE-581 83 Linkping, Sweden, Jan 2001. Available from `http://www.control.isy.liu.se/~johanl/yalmip.html`.

[36] A. Megretski, C-Y. Kao, U. Jönsson, and A. Rantzer. *A Guide To IQCβ: Software for Robustness Analysis*, June 1998.

[37] S. A. Miller and R. S. Smith. Solving large structured semidefinite programs using an inexact spectral bundle method. In *Proceedings of the 39th IEEE Conference on Decision and Control*, pages 5027–5032, December 2000.

[38] H. D. Mittelmann. An independent benchmarking of SDP and SOCP solvers. *Math. Progr.*, 95:407–430, 2003.

[39] Y. E. Nesterov and A. Nemirovski. *Interior point polynomial methods in convex programming*, volume 13 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, 1994.

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

[41] A. Papachristodoulou and S. Prajna. On the construction of Lyapunov functions using the sum of squares decomposition. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 3482–3487, December 2002.

[42] P. A. Parrilo. On the numerical solution of LMIs derived from the KYP lemma. In *Proceedings of the 38^{th} IEEE Conference on Decision and Control*, 1999.

[43] P. A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, May 2000. Available at `http://www.cds.caltech.edu/~pablo/`.

[44] P. A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. To appear in *Math. Prog. Series B*. Preprint available at `http://control.ee.ethz.ch/~parrilo/`, 2001.

[45] P.A. Parrilo and B. Sturmfels. Minimizing polynomial functions. In S. Basu and L. González-Vega, editors, *Algorithmic and Quantitative Real Algebraic Geometry*, vol-

ume 60 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. AMS, 2003. To appear, available from `arXiv:math.OC/0103170`.

[46] V. Powers and T. Wörmann. An algorithm for sums of squares of real polynomials. *Journal of pure and applied algebra*, 127:99–104, 1998.

[47] S. Prajna, A. Papachristodoulou, and P. A. Parrilo. *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*, 2002. Available from `http://www.cds.caltech.edu/sostools` and `http://control.ee.ethz.ch/~parrilo/sostools`.

[48] M. V. Ramana. Polyhedra, spectrahedra, and semidefinite programming. In P. M. Pardalos and H. Wolkowicz, editors, *Topics in Semidefinite and Interior-Point Methods*, Fields Institute Communications, V. 18., pages 27–38. American Mathematical Society, 1998.

[49] A. Rantzer. On the Kalman-Yakubovich-Popov lemma. *Systems & Control Letters*, 28:7–10, 1996.

[50] B. Reznick. Uniform denominators in Hilbert 17th problem. *Mathematische Zeitschrift*, 220(1):75–97, 1995.

[51] B. Reznick. Some concrete aspects of Hilbert's 17th problem. In *Real Algebraic Geometry and Ordered Structures*, Contemporary Mathematics (American Mathematical Society), V. 253. American Mathematical Society, 2000.

[52] M. A. Rotea and F.J. D'Amato. LFTB: an optimized algorithm to bound worst-case frequency response functions. In *Proceedings of the American Control Conference*, pages 3041–3048, June 2001.

[53] C. Scherer and S. Weiland. Linear Matrix Inequalities in Control. DISC course lecture notes, 2000.

[54] P. Seiler. *LMILab translator*. Available from `http://vehicle.me.berkeley.edu/~guiness/lmitrans.html`.

[55] N. Z. Shor. *Minimization methods for nondifferentiable functions*, volume 3 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1985.

[56] J. Sturm. *SeDuMi version 1.05*, October 2002. Available from `http://fewcal.kub.nl/sturm/software/sedumi.html`.

[57] B. Tibken and K. F. Dilaver. Computation of subsets of the domain of attraction for polynomial systems. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 2651–2656, December 2002.

[58] M. Todd. Semidefinite optimization. *Acta Numerica*, 10:515–560, 2001.

[59] K. C. Toh, R. H. Tütüncü, and M. J. Todd. *SDPT3 - a MATLAB software package for semidefinite-quadratic-linear programming*. Available from `http://www.math.cmu.edu/~reha/sdpt3.html`.

[60] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, March 1996.

[61] L. Vandenberghe, S. Boyd, and B. Alkire. SP Version 1.1. Software for Semidefinite Programming. `http://www.ee.ucla.edu/~vandenbe/sp.html`.

[62] A. Varga and P.A. Parrilo. Fast algorithms for solving $\mathcal{H}_\infty$ norm minimization problems. In *Proceedings of the 40^{th} IEEE Conference on Decision and Control*, 2001.

[63] J. C. Willems. Least squares stationary optimal control and the algebraic Riccati equation. *IEEE Transactions on Automatic Control*, 16(6):621–634, 1971.

[64] H. Wolkowicz, R. Saigal, and L. Vandenberghe, editors. *Handbook of Semidefinite Programming*. Kluwer, 2000.